



*Эффективная
работа™*

Flash MX



Этан Уотролл, Норберт Гербер



СЕРИЯ

Эффективная
Работа™

 ПИТЕР®

Ethan Watrall, Norbert Herber

Flash MX Savvy



San Francisco • Paris • Düsseldorf • Soest • London

Этан Уотролл, Норберт Гербер



FlashMX



Москва • Санкт-Петербург • Нижний Новгород • Воронеж
Ростов-на-Дону • Екатеринбург • Самара
Киев • Харьков • Минск

2003

Этан Уотролл, Норберт Гербер
Эффективная работа: Flash MX

Перевод с английского В. Кочерги

Главный редактор
Заведующий редакцией
Редакторы
Художник
Корректор
Технический редактор

*Е. Строганова
И. Корнеев
Т. Мартынюк, Е. Насырова
Н. Биржаков
И. Карпышенко
О. Заплаткина*

ББК 32.973-044

УДК 681.327.1

Уотролл Э., Гербер Н.

У65 Эффективная работа: Flash MX (+CD). — СПб.: Питер; Киев: BHV, 2003. — 720 с.: ил.

ISBN 5-8046-0009-5

Книга посвящена лучшей программе создания векторной анимации — Macromedia Flash MX. Кроме описания возможностей этого приложения в книге приведены практические примеры создания профессиональной анимации. Большое внимание уделено программированию на языке ActionScript, разработке интерактивного интерфейса фильмов, созданию звукового сопровождения и интеграции Flash MX с другими приложениями. Рассмотрены особенности публикации Flash-фильмов в Интернете и на компакт-дисках.

К книге прилагается компакт-диск с набором всех необходимых приложений, файлов поддержки и примеров, необходимых для выполнения практических заданий. Руководство будет полезно как начинающим, так и опытным пользователям, работающим в операционных системах MacOS и Windows.

© SYBEX Incorporated, 2002

© Издательская группа BHV, Киев, 2003

© ЗАО Издательский дом «Питер», 2003

Права на издание получены по соглашению с Sybex Inc.

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 5-8046-0009-5

ISBN 0-7821-4108-0 (англ.)

ООО «Питер Принт». 196105, Санкт-Петербург, ул. Благодатная, д. 67в.

Лицензия ИД № 05784 от 07.09.01.

ООО «Издательская группа BHV»

Свидетельство о занесении в Государственный реестр

серия ДК № 175 от 13.09.2000.

Налоговая льгота – общероссийский классификатор продукции ОК 005-93, том 2; 953005 – литература учебная.

Подписано в печать 03.10.02. Формат 70×100/16. Усл. п. л. 58,05. Тираж 4500 экз. Заказ № 1559.

Отпечатано с диапозитивов в ФГУП «Печатный двор» им. А. М. Горького

Министерства РФ по делам печати, телерадиовещания и средств массовых коммуникаций.

197110, Санкт-Петербург, Чкаловский пр., 15.

Краткое содержание

Об авторах	15
Благодарности	18
Вступительное слово	20
Введение	21

ЧАСТЬ 1. Знакомимся с Flash

Глава 1. Введение в Flash MX	28
Глава 2. Новинки Flash MX	35
Глава 3. Интерфейс	39
Практикум 1. Пользовательские комбинации клавиш	58

ЧАСТЬ 2. Содержимое фильма: создание и управление

Глава 4. Рисование и черчение	62
Глава 5. Работа с текстом	87
Глава 6. Работа с объектами	100
Глава 7. Объекты многократного использования	121
Глава 8. Слои	148
Практикум 2. Создание изображения сада с использованием объектов и слоев	160

ЧАСТЬ 3. Создание анимации в среде Flash

Глава 9. Анимация с использованием временной шкалы	168
Глава 10. Работа с клипами	190
Глава 11. Использование сцен при работе с анимационными фильмами	201
Практикум 3. Создание сцены полета космической ракеты	208

ЧАСТЬ 4. Интерактивные фильмы

Глава 12. Базовые действия ActionScript	222
Глава 13. Создание интерактивных элементов управления	247
Практикум 4. Создание интерактивного резюме с использованием базовых действий и навигационной панели	268

ЧАСТЬ 5. Язык ActionScript

Глава 14. Использование ActionScript	280
Глава 15. Написание сценариев ActionScript	301
Глава 16. Повышение интерактивности фильмов с помощью ActionScript	319
Глава 17. Управление вводом-выводом данных	349
Глава 18. Отладка сценариев ActionScript	362
Практикум 5. Использование ActionScript для создания интерактивной, многослойной временной шкалы	376

ЧАСТЬ 6. Работа со звуком

Глава 19. Принципы цифровой записи звука в среде Flash	386
Глава 20. Основы работы с аудиофайлами в среде Flash	400
Глава 21. Синхронизация звука и анимации	415
Глава 22. Программное управление звуком	427
Практикум 6. Создание звукового сопровождения фильма	452

ЧАСТЬ 7. Использование Flash с другими программами

Глава 23. Совместная работа Flash и Director	464
Глава 24. Совместная работа Flash и FreeHand	481
Глава 25. Использование звуковых редакторов совместно с Flash	490
Глава 26. Создание трехмерных объектов в среде Flash	510
Практикум 7. Создание анимированного трехмерного логотипа	546

ЧАСТЬ 8. Публикация и распространение Flash-фильмов

Глава 27. Публикация Flash-фильмов	554
Глава 28. Работа Flash с другими видеоформатами	574
Глава 29. Публикация Flash-фильма на компакт-диск	600
Практикум 8. Создание универсального компакт-диска	610

Приложения

Приложение А. Создание дополнительных элементов	618
Приложение Б. Справочник по ActionScript	639
Алфавитный указатель	707

Содержание

Об авторах	15
Благодарности	18
Вступительное слово	20
Введение	21

ЧАСТЬ 1. Знакомимся с Flash

Глава 1. Введение в Flash MX	28
История создания Flash	28
Сравнительный анализ растрового и векторного форматов	30
Области применения Flash	32
Глава 2. Новинки Flash MX	35
Улучшенный процесс разработки и практичность	35
Новые инструменты для создания визуальных эффектов	36
Расширенные медиавозможности	37
Новые возможности ActionScript	37
Новые возможности публикации и распространения	38
Резюме	38
Глава 3. Интерфейс	39
Освоение рабочей среды Flash	39
Знакомство с панелью инструментов Tools	43
Область действия и рабочая область	44
Практикум 1. Пользовательские комбинации клавиш	58
Изменение клавиш быстрого вызова команд	58

ЧАСТЬ 2. Содержимое фильма: создание и управление

Глава 4. Рисование и черчение	62
Выбор объектов	62
Рисование графических элементов	67
Инструмент Ink Bottle: управление обводкой	73
Изменение заливки с помощью инструмента Paint Bucket	74
Инструмент Fill Transform	75
Копирование цветов с помощью инструмента Eyedropper	77
Инструмент Eraser	78
Изменение обводки с помощью панели Properties	78
Изменение заливки с помощью панели Properties	80

Основные положения теории цвета.....	81
Разработки.....	85
Резюме.....	86
Глава 5. Работа с текстом.....	87
Текст, шрифт и гарнитура.....	87
Ограничения при использовании текста в Flash-фильмах.....	87
Использование инструмента Text.....	88
Текстовые поля.....	89
Манипулирование текстом: панель Properties.....	90
Разбивка текста.....	97
Изменение формы текстовых СИМВОЛОВ.....	98
Создание шрифтовых символов.....	98
Разработки.....	99
Резюме.....	99
Глава 6. Работа с объектами.....	100
Понятие объекта.....	100
Работа с растровыми изображениями.....	100
Группирование объектов.....	107
Позиционирование объектов.....	109
Панель Align.....	110
Порядок размещения объектов.....	113
Трансформация объектов.....	113
Работа с точками трансформации.....	119
Разработки.....	119
Резюме.....	120
Глава 7. Объекты многократного использования.....	121
Символы.....	121
Создание символов.....	123
Инструментальные средства библиотек.....	126
Работа с библиотеками символов.....	130
Работа с экземплярами символов.....	139
Панель Movie Explorer.....	143
Разработки.....	146
Резюме.....	147
Глава 8. Слои.....	148
Принципы использования слоев.....	148
Создание слоя.....	149
Добавление объектов на слой.....	150
Настройка слоев.....	150

Каталоги слоев.....	152
Редактирование слоев.....	154
Распределение объектов по слоям.....	156
Создание направляющих слоев.....	156
Создание слоев-масок.....	157
Разработки.....	158
Резюме.....	159

Практикум 2. Создание изображения сада с использованием объектов и слоев..... 160

Фон: небо.....	160
Почва.....	160
Подготовка к созданию клумб.....	161
Создание клумб.....	161
Перспектива.....	162
Завершение работы над клумбами.....	163
Элементы цветов.....	164
Посадка цветов.....	166

ЧАСТЬ 3. Создание анимации в среде Flash.....

Глава 9. Анимация с использованием временной шкалы..... 168

Что такое анимация.....	168
Создание покадровой анимации.....	174
Автоматическое построение промежуточных кадров.....	177
Анимирование маски.....	184
Остановка движения объекта во время анимации.....	186
Создание эффектов анимации с помощью панели Properties.....	186
Разработки.....	189
Резюме.....	189

Глава 10. Работа с клипами..... 190

Что такое клип.....	190
Создание нового клипа.....	191
Преобразование существующей анимации в клипы.....	192
Вставка клипа в кадр на главной временной шкале.....	192
Манипулирование клипами.....	193
Пример практического создания клипа.....	195
Разработки.....	199
Резюме.....	200

Глава 11. Использование сцен при работе с анимационными фильмами	201
Знакомство со сценами	201
Создание сцен и манипулирование ими	203
Переключение между сценами	204
Тестирование сцен	205
Возможности управления сценами с помощью сценариев ActionScript ..	206
Разработки	206
Резюме	207
Практикум 3. Создание сцены полета космической ракеты	208
Изображение космического корабля	208
Отправление и удаление от Земли	211
Путешествие в космосе	214
Приближение к планете X	216
Конечная сцена	218

ЧАСТЬ 4. Интерактивные фильмы

Глава 12. Базовые действия ActionScript	222
Действия	222
Использование действий	228
Разработки	245
Резюме	246
Глава 13. Создание интерактивных элементов управления	247
Управление интерактивными фильмами с помощью простых кнопок	247
Предварительный просмотр, тестирование и редактирование кнопок ..	253
Создание сложных кнопок	256
Разработки	266
Резюме	267
Практикум 4. Создание интерактивного резюме с использованием базовых действий и навигационной панели	268
Подготовка области действия	268
Создание интерактивных элементов	270
Создание элементов навигации	273
Написание сценария	275
Завершающий этап создания резюме	275

ЧАСТЬ 5. Язык ActionScript

Глава 14. Использование ActionScript	280
Знакомство с ActionScript	280
Терминология ActionScript	286
Использование панели Actions для создания сценариев	288
Присоединение сложных сценариев ActionScript	297
Резюме	300
Глава 15. Написание сценариев ActionScript	301
Синтаксис языка ActionScript	301
Элементы языка ActionScript	306
Разработки	318
Резюме	318
Глава 16. Повышение интерактивности фильмов с помощью ActionScript	319
Программная обработка действий пользователя	319
Сценарии ActionScript для фильмов с несколькими временными шкалами	331
Программное управление графическими элементами	340
Разработки	347
Резюме	348
Глава 17. Управление вводом-выводом данных	349
Приложения, использующие базы данных	349
Создание динамического внешнего интерфейса	350
Представление динамического содержимого	355
Генерирование содержимого из текстовых файлов	358
Разработки	360
Резюме	361
Глава 18. Отладка сценариев ActionScript	362
Правила разработки сценариев	362
Методы устранения ошибок	363
Использование отладчика	366
Контроль сценариев с помощью окна Output	371
Резюме	375
Практикум 5. Использование ActionScript для создания интерактивной, многослойной временной шкалы	376
Подготовка фильма	377
Добавление сценария ActionScript	378
Добавление кнопок управления воспроизведением	381
Резюме...	...383

ЧАСТЬ 6. Работа со звуком

Глава 19. Принципы цифровой записи звука в среде Flash	386
Основные принципы цифровой записи звука.....	386
Подготовка аудиофайлов для работы в среде Flash.....	391
Разработки.....	398
Резюме.....	399
Глава 20. Основы работы с аудиофайлами в среде Flash	400
Импортирование звуковых файлов.....	400
Включение звука в фильм.....	401
Остановка воспроизведения звука.....	403
Привязка звука к кнопкам.....	403
Создание аудиоэффектов.....	405
Экспортирование и сжатие звуковых файлов.....	408
Разработки.....	414
Резюме.....	414
Глава 21. Синхронизация звука и анимации	415
Типы синхронизации в среде Flash.....	415
Приемы синхронизации.....	419
Разработки.....	426
Резюме.....	426
Глава 22. Программное управление звуком	427
Работа с объектами Sound.....	427
Воспроизведение звуков с помощью объекта Sound.....	432
Программирование уровня громкости и панорамирования.....	435
Разработки.....	450
Резюме.....	451
Практикум 6. Создание звукового сопровождения фильма	452
Планирование использования звуков в фильме.....	452
Создание звукового сопровождения.....	453
Разработка интерфейса.....	460
Заключительные размышления о звуковом сопровождении.....	462

ЧАСТЬ 7. Использование Flash с другими программами

Глава 23. Совместная работа Flash и Director	464
Работа с Flash в среде Director.....	464
Комбинирование языков ActionScript и Lingo.....	472
Разработки.....	480
Резюме.....	480

Глава 24. Совместная работа Flash и FreeHand	481
Общие сведения о программе FreeHand.....	481
Создание многокадровой анимации.....	482
Моделирование анимации перехода.....	483
Предварительный просмотр Flash-анимации во FreeHand.....	484
Использование режима Anti-Alias.....	485
Привязка команд к графическому объекту.....	485
Задание параметров Flash-фильма во FreeHand.....	486
Экспорт файлов FreeHand во Flash.....	488
Разработки.....	489
Резюме.....	489
Глава 25. Использование звуковых редакторов совместно с Flash	490
Звуковые редакторы.....	490
Приемы обработки цифрового звука.....	499
Разработки.....	509
Резюме.....	509
Глава 26. Создание трехмерных объектов в среде Flash	510
Имитация трехмерного пространства в программе Flash.....	510
Экспорт трехмерных изображений в формат SWF.....	517
Работа с программами, не поддерживающими экспорт в формат SWF.....	541
Разработки.....	545
Резюме.....	545
Практикум 7. Создание анимированного трехмерного логотипа....	546
Экспорт трехмерной анимации.....	546
Работа с анимацией во Flash.....	548
Вставка анимации в документ Dreamweaver.....	550
ЧАСТЬ 8. Публикация и распространение Flash-фильмов	
Глава 27. Публикация Flash-фильмов	554
Установка параметров обработки Flash-фильма при его публикации.....	554
Предварительный просмотр Flash-фильма.....	569
Публикация Flash-фильма.....	569
Публикация фильма, доступного для людей с физическими недостатками.....	569
Создание фильмов, содержимое которых может быть выведено на печать.....	571
Разработки.....	573
Резюме....	573

Глава 28. Работа Flash с другими видеоформатами	574
Видео в среде Flash.....	575
Публикация Flash-фильмов на видеокассетах.....	584
Подготовка видео для переноса на видеоленту.....	594
Разработки.....	599
Резюме.....	599
Глава 29. Публикация Flash-фильма на компакт-диск	600
Автономные Flash-файлы.....	600
Создание файлов в формате projector.....	602
Интерфейс управления фильмом.....	603
Разработки.....	608
Резюме.....	609
Практикум 8. Создание универсального компакт-диска	610
Подготовка файлов.....	610
Разбивка диска.....	611
Запись данных и проверка компакт-диска.....	614
Дополнительные сведения.....	614

Приложения

Приложение А. Создание дополнительных элементов	618
Создание навигационных элементов управления.....	618
Анимации, используемые для отображения процесса загрузки.....	630
Использование массива для создания динамического приветствия.....	635
Приложение Б. Справочник по ActionScript	639
Действия.....	640
Свойства.....	653
Функции.....	658
Операторы.....	659
Объектные ссылки и ссылки на временную шкалу.....	665
Предопределенные объекты.....	667
Значения кодов клавиш для объекта Key.....	703
Устаревшие термины.....	704
Устаревшие термины.....	709
Алфавитный указатель	707

Об авторах

Этан Уотролл родился в городе Реджайна, провинция Саскачеван, Канада. Закончил с отличием университет Реджайны, получив степень бакалавра по антропологии и истории.

Будучи студентом, он принимал участие во многих археологических раскопках в Саскачеване, Манитобе и Индиане. Окончив университет, Уотролл поступил в аспирантуру университета Индианы по специальности антропология. Здесь он сконцентрировал свои усилия на двух основных предметах. Во-первых, продолжил свои археологические изыскания, занявшись изучением домашней утвари додинастического Египта. Он работал в Набта Плайя (огромной неолитической стоянке в Западной пустыне Египта) и Иераконполисе (возможно, одной из самых важных стоянок додинастического периода во всем Египте). В результате раскопок им были обнаружены такие ценные находки, как доисторические колодцы, глиняные копии, домашние постройки, загоны для животных, печи для обжига и сушки гончарных изделий и места захоронений древних людей. Его нынешние исследования местности НК11 (поселение позднего додинастического периода в Иераконполисе) позволили получить большой объем информации, значительно расширившей и во многом изменившей существующий взгляд на домашнее хозяйство додинастического Египта.

Вторым предметом интересов Этана Уотролла стало использование интерактивных средств в археологии. Он принял участие в дискуссии, посвященной этой теме, то ли просто считая данную сферу заслуживающей внимания, то ли заинтересовавшись ее образовательными возможностями в качестве средства для научных публикаций. Особое внимание он уделяет роли интерактивных развлекательных средств, знакомящих широкую публику с археологией. Он опубликовал ряд работ, посвященных этой теме и адресованных профессионалам как в области археологии, так и в области разработки компьютерных игр. Причем в них Этан не просто комментирует сложившуюся ситуацию, а выступает за активное участие ученых в индустрии интерактивных развлечений.

Кроме научной работы Уотролл принимает непосредственное участие в работе с интерактивными средствами и мультимедиа, ведет занятия по их разработке в Государственном техническом университете (Ivy Tech State University) города Колумбус штат Индиана и Государственном техническом колледже (Ivy Tech State College) города Блумингтон штат Индиана. Первая книга Этана Уотролла *Dreamweaver 4/Fireworks 4 Visual JumpStart* была опубликована издательством Sybex в феврале 2001 года и переведена на китайский и греческий языки. Сейчас он работает над третьей книгой *Dreamweaver MX: Design and Technique*, которая будет опубликована издательством Sybex летом 2002 года.

Цифровое alter ego Этана Уотролла можно найти по адресу: www.captainpri-mate.com.

Норберт Гербер вырос в Миннеаполисе, штат Миннесота. В шестнадцатилетнем возрасте он начал карьеру как джазовый саксофонист, создавая музыкальные композиции с известным тенором свинга Ирвом Уильямсом. Набравшись опыта,

Гербер поступил на музыкальное отделение университета Индианы (Indiana University School of Music) в Блумингтоне. Здесь он продолжил изучение джаза с Дэвидом Бейкером, занимаясь импровизацией, аранжировкой, теорией музыки, деревянными духовыми инструментами и сочинением музыки.

По окончании университета Норберт работал свободным музыкантом и аранжировщиком, создал свою группу и написал музыку к короткометражному фильму *Jambalaya*. Вскоре молодой композитор заинтересовался цифровым представлением художественных произведений. Стремясь изучить взаимодействие между звуком, текстом и изображением, он вернулся в университет Индианы для работы на факультете телекоммуникаций в программе MIME. Здесь он интенсивно и целенаправленно изучает электронную музыку, озвучивание фильмов, методы цифровой обработки звука и создание интерактивных музыкальных композиций.

Сейчас Норберт работает над несколькими интерактивными проектами цифровой звукозаписи, в том числе над акустическим исследованием произведений Уильяма С. Барроуза (проект *Word Is Virus*). Его интересы сконцентрировались на использовании аудио- и интерактивной среды, нелинейных музыкальных композиций и эффектов взаимодействия звук-образ как в линейных, так и в нелинейных средствах. Норберт является преподавателем Международной академии дизайна и технологий (International Academy of Design and Technology) в Чикаго, где ведет занятия по разработке и созданию сценариев для мультимедиа, интерактивному дизайну и цифровой звукозаписи.

О нынешних и более ранних проектах Норберта Гербера вы можете узнать по адресу www.x-tet.com.

Тейлор с надеждой, что однажды
она напишет книги лучше этой.

Этан Уотролл

Эта книга посвящается Дженни. Твое терпение,
любовь и понимание сделали возможным ее выход.

Норберт Гербер

Благодарности

Как это обычно бывает, особенно если книга имеет такой объем, многие люди заслужили благодарность за помощь в ее создании. В первую очередь я должен высказать благодарность своему другу, соавтору и партнеру во многих предприятиях, связанных с цифровыми технологиями, — Норберту Герберу. Когда обязанности по написанию объемной книги разделены между двумя авторами, это сложнейшая и не всегда разрешимая задача. Но мы преодолели трудности и смогли создать эту книгу. Что ж, за дальнейшее плодотворное сотрудничество! Огромное спасибо всем тем, кто любезно дал согласие на использование их работ в разделе «Разработки» каждой главы этой книги (к сожалению, всех перечислить невозможно). Моя особая благодарность Джусту ван Шейку и Брайану Тейлору, чей вклад в эту книгу и помощь в ее написании неоценимы. Как всегда, хочу выразить свою глубокую любовь и признательность Дженн. Она великодушно мирилась с моей бессонницей и долгим сидением за компьютером, пока в течение года писалась, переписывалась, проверялась и перепроверялась эта книга. Спасибо моим родителям и моему другу Нилу Берчу за поддержку в работе над этим проектом. Наконец, особая благодарность моему псу Оскару — ничто так не помогает спуститься на землю, как щенок, которому только бы играть и больше ничего не нужно. В заключение хочу принести свои извинения всем тем, о ком я умудрился забыть. Спасибо всем!

Этан Уотролл

Эта книга — плод не только обоюдных усилий двух авторов, поскольку каждый из нас имел свою собственную «группу поддержки», участники которой неустанно помогали в ее написании. Выражаю благодарность Биллу Фризелю, Майлзу Дейвису, Джеймсу Брауну и многим другим, чьи музыкальные работы вдохновляли меня при написании большей части этого текста. Спасибо Питу за чай и кофе, которые давали мне заряд в долгие часы работы над ActionScript, а также кафе Laughing Planet Cafe and Soma в городе Блумингтон штата Индиана за предоставление возможности проведения нескольких долгих собраний, посвященных книге. Спасибо всем тем, кто поделился своей работой для разделов «Разработки», особенно Нико Стампо, Жан Люку Ламарку, NPFC, Джо Спарксу и Донне Дюрант из LEGO Systems, Inc. Хочу также поблагодарить Шеннон Застроу и Райана Дугласа из DoReMedia за идеи и материал, вошедшие в главу 6. Спасибо моим чикагским коллегам Тиму Арройо и Берни Маку из IADT за их вклад в эту книгу, а также Брайану Грастару. Огромная благодарность моему «сообщнику» Этану Уотроллу, в первую очередь за то, что убедил меня взяться за эту работу. Я знал, что это сработает! Спасибо за усердие и изобретательность. Я также должен поблагодарить мою замечательную семью, оказавшую мне огромную поддержку: спасибо маме, папе, Женевьеве, Лиле, Сюзане, Хасану и особенно моей дорогой Дженни. Спасибо, что выслушивали рассказ за рассказом об этой «книге». Наконец, спасибо *Вам* за то, что читаете эту книгу!

Норберт Гербер

Мы оба хотим выразить благодарность многим людям, помогавшим нам воплотить эту книгу в действительность. Спасибо нашему агенту Дэвиду Фугейту из Waterside Productions, Inc. Мы, как всегда, в долгу перед ним за его труд и советы, которыми он снабжал нас во время работы над проектом. Заслуживает похвалы неоценимая помощь и труд сотрудников Sybex Дениса Фицджеральда, Джима Габберта, Керол Генри, Ким Уимпсетт, Дена Шиффа, Дена Маммерта и Морин Форис. Спасибо Дениз Тайлер, которая очень помогала нам во время написания книги, и Эрику Беллу за техническую проницательность. Без помощи Мериенн Барсоло, нашего редактора из Sybex, этот проект не мог бы быть завершен. Она всегда была рядом со своими советами, помощью и пониманием, особенно в наиболее трудные периоды. Огромная благодарность Тому Джиллспи за работу над введением, а также Брайану Тейлору (Rust Boy & XL5 Design), Виржинии Милке (Nelvana), Дереку Мошеру (Sony Screenblast) и Джусту ван Шейку (Djojo Studios) за их любезную поддержку. Мы также хотим поблагодарить всех, кто помогал нам с программами, помещенными на компакт-диск, и оборудованием, применявшемся при написании этой книги: Уильяма Роба (Wasom), Филиппа Штайгера (Eovia), Пола Бабба (Махон), Карен Карпентер (Curious Labs), Стефана Мосса (Ideaworks3D), Бена Йодера (Electric Image), Дейва Клейна (Electric Rain), Стива Фолдвари (Sonic Foundary), Дениеля Брауна (Adobe Systems Incorporated) и Зака Уиткрофта (BIAS, Inc.).

Вступительное слово

Когда появилось приложение FutureSplash, я и большая часть сообщества новых средств массовой информации считали, что главной программой для нас является Director. И вдруг в сети появился этот сумасшедший инструмент анимации. Я без конца загружал его и играл. Он напоминал Director, поскольку содержал временную шкалу, но, казалось, что работает он совершенно иначе, поэтому его трудно использовать. С другой стороны меня удивила легкость осуществления анимации, заставлявшей глаза мигать, а шары вращаться. Приложение FutureSplash было забавой и не казалось особо полезным, поэтому я возвратился к борьбе с растровыми изображениями, звуками и Lingo. Я твердо знал, за чем будущее.

Спустя несколько лет в университете Индианы я создал программу MIME для исследований в сфере интерактивных средств массовой информации. Двумя моими первыми студентами стали Этан и Норберт. Этан работал над диссертацией по археологии, а Норберт был джазовым саксофонистом. Этан занимался попеременно новыми средствами массовой информации и раскопками в долине Нила, Норберт создавал забавные звуки. Такие творческие натуры и бесспорно светлые головы, они прекрасно работали, но не были настоящими «технарями». Мне казалось, что из них может выйти толк, но я должен был работать с профессионалами. Я твердо знал, за чем будущее.

Оказалось, я не имел ни малейшего представления о том, за чем будущее. Flash сейчас имеет такую значимость, какой Director никогда не имела, а Этан и Норберт продолжают работу и добились многого с различными технологиями, именно потому, что они *не* «технари». И вот они написали эту книгу для умных творческих людей, интересующихся не технологиями ради технологий, а средствами для достижения конкретного результата.

Эта книга не похожа на традиционные технические книги с их циклами и переменными. Она также не относится к книгам типа «смотрите, что мы можем сделать» — этакая конфетка в блестящей обертке. В отличие от других книг, посвященных Flash, в ней содержится значительный объем информации об интеграции Flash с воспроизведением и записью звуков. Программа Flash MX предоставляет богатейшие возможности работы со звуком, а эта книга объясняет, как применять новые инструменты с пользой для аудитории.

Этан разбирается в культуре человечества: он знает, как люди жили, работали, обучались. Норберт — музыкант, который в дополнение к музыке занимается интерактивными разработками, графикой и преподаванием. Написав книгу о программе Flash, они показали ее возможности гораздо лучше, чем это могли бы сделать любые профессиональные «технари». Это прекрасная книга о том, как применять Flash, чтобы достичь реальных результатов в реальном мире.

Д-р Том Джиллсти,
директор программы MIME (www.mime.indiana.edu),
доцент факультета телекоммуникаций университета Индианы.

Введение

Летом 1996 года мало кому известная компания FutureWave выпустила небольшую программу FutureSplash Animator, предназначенную для создания линейных движущихся изображений на базе векторной графики. Вскоре после выхода приложение привлекло к себе внимание и было использовано при разработке Web-версии MSN компанией Microsoft и электронной версии Disney Daily Blast компании Disney. В ноябре 1996 года FutureWave получила предложение о совместной работе от компании Macromedia. Для небольшой компании, в которой к тому времени работало всего шесть человек, это был прекрасный шанс. В декабре 1996 года компания Macromedia купила технологию у FutureWave и в начале 1997 года выпустила первую версию Flash.

К настоящему времени вышло шесть версий приложения, и можно смело утверждать, что последняя дублированная версия Flash MX является одной из наиболее популярных интерактивных программ. С ее помощью можно создавать не только замечательную анимацию, но и достаточно сложные динамические интерактивные программы, которые управляются базами данных.

Хотя изначально Flash была предназначена для работы в сети, огромные возможности и феноменальная мощь сделали ее популярной далеко за пределами Web. В наши дни интеграцию Flash вы можете увидеть в инструментальных переносных средствах, широкоэмитательных устройствах и игровых приставках.

Чтобы понять, насколько широко используется Flash в современных цифровых технологиях, достаточно посмотреть на число пользователей, применявших проигрыватель Flash за эти годы, — более 430 миллионов. Цифра впечатляет, почти полмиллиарда! Нужно ли еще что-либо говорить?

О данной книге

Мы, авторы, вложили в эту книгу много сил и энергии. Нашей целью было создание максимально наглядного учебного пособия по Flash.

В отличие от многих других пособий по данной теме наша книга пытается объяснить не только, *как* работать в среде Flash MX, но еще и *зачем*. Это ключевой шаг в процессе обучения, поскольку ответ на такой вопрос дает пользователям возможность творчески применять полученные знания и навыки для реализации собственных идей. Методики, представленные в данной книге, дадут вам базовые знания для создания огромного числа различных элементов в среде Flash. Кроме того, учебный материал составлен таким образом, чтобы стимулировать интерес к дальнейшему обучению. Надеемся, что наши уроки вы используете как отправную точку в процессе развития ваших собственных творческих идей.

Данная книга будет в равной степени полезна для пользователей и Macintosh и Windows-платформы. Как творческая группа мы разделились: Норберт — сторонник Macintosh, а Этан — сторонник Windows. Мы старались, насколько это было возможно, представить сбалансированный подход. Таким образом, независимо от используемой платформы, вы найдете в данной книге надежного помощника при изучении Flash.

Кому нужна эта книга

Все мужчины, женщины и дети на нашей планете нуждаются в этой книге!

Но если серьезно, наверное, невозможно написать книгу, которая нужна всем. Однако мы приложили максимум усилий, чтобы наш труд нашел как можно большее число сторонников. Темы были тщательно отобраны и освещены таким образом, чтобы материал могли использовать различные люди с различным уровнем знаний и опыта. В крайнем случае, если содержимое книги вам не понравится, ее можно подложить под шаткую ножку кровати или успешно использовать в качестве добротного дверного упора.

Хотя данная книга не адресована разработчикам, имеющим большой опыт работы с Flash, мы бы не хотели отговаривать их от знакомства с ней — совсем наоборот! Ведь здесь собран большой материал, с которым могут быть незнакомы даже опытные пользователи.

Думаем, что каждый, кто желает быть участником «Flash-революции» должен прочесть эту книгу. В данный круг людей входят студенты, которые хотят разместить проекты в Web, аниматоры или художники, желающие представить свои работы в цифровом формате, все, кто разочарован ограниченными возможностями HTML, и все, кому интересна векторная анимация и интерактивность. Одним словом те, кого интересуют безграничные возможности и феноменальная мощь Flash, но кто не знает с чего начать. Если вы один из таких людей, эта книга для вас!

Как составлена эта книга

Работая с программой Flash MX, вы заметите, как ваши навыки естественным образом будут улучшаться. Мы структурировали книгу с учетом принципа предметности и перспективности, но при этом пытались сделать каждую главу (или часть) насколько можно самостоятельной. То есть фактически книга может использоваться как справочное руководство теми пользователями, которые имеют базовые навыки и интересуются только специфическими вопросами.

MX С помощью специального значка «Новое в MX» мы указываем абзацы, в которых речь идет о новых свойствах, появившихся в версии Flash MX. Такие указатели создадут удобства пользователям, которые работали с версией Flash 5 и впервые знакомятся с возможностями программы Flash MX.

Давайте вкратце рассмотрим, какие темы будут освещены в каждой главе.

Часть 1. Знакомство с Flash

Работа над частью 1 — закладка фундамента всей вашей будущей работы с Flash. Глава 1 представляет собой введение в мир Flash. Вы начнете с изучения истории развития Flash, включая рассмотрение свойств и возможностей каждой версии, узнаете о различии между растровой и векторной графикой и познакомитесь с теми продуктами, которые создаются с использованием технологии Flash. В главе 2 мы расскажем о новых свойствах, появившихся в последней версии. Наконец, в главе 3 будет детально представлен интерфейс Flash MX, и вы узнаете

обо всех основных его элементах, а также практически научитесь настраивать рабочую среду.

Часть 2. Создание и управление содержимым фильма

В части 2 ваше внимание будет сфокусировано на вопросах создания и манипулирования всем визуальным содержимым фильма Flash. В главе 4 вы познакомитесь с инструментами Flash MX для рисования и черчения изображений. Далее в главе 5 мы рассмотрим процесс создания и манипулирования текстом. Из главы 6 вы узнаете, как эффективно использовать объекты (неопределенный термин, применимый к любому графическому элементу в Flash MX). В главе 7 детально рассматриваются методы работы с объектами многократного использования, в частности, речь пойдет о библиотеках. Глава 8 освещает вопросы работы со слоями, эта тема очень важна, если вы хотите создавать анимацию произвольной степени сложности.

Часть 3. Анимирование изображений с помощью Flash

Часть 3 данной книги призвана научить вас всему, что необходимо знать о создании анимации с помощью Flash. В главе 9 детально рассматриваются вопросы использования временной шкалы для создания движущихся изображений. Далее из главы 10 вы узнаете, как создаются клипы — один из наиболее важных элементов фильма Flash. Изучив главу 11, вы научитесь работать со сценами, являющимися важнейшим инструментом для разделения и организации анимированного содержимого.

Часть 4. Написание сценариев для интерактивных фильмов с помощью языка ActionScript

Эта часть книги посвящена одному из наиболее сложных и важных аспектов разработки фильмов Flash: языку ActionScript. В главе 12 мы расскажем о базовых понятиях языка ActionScript и о технике добавления сценариев для управления интерактивностью фильмов. Глава 13 посвящена созданию и разработке дизайна интерактивных элементов контроля — важнейшей части любого интерактивного фильма.

Часть 5. Создание сложных интерактивных фильмов с помощью языка ActionScript

Если вы любите Flash, но не решаетесь погрузиться в мир написания сценариев, часть 5 этой книги адресована вам. В главе 14 описаны некоторые базовые концепции программирования и объяснен механизм функционирования сценария ActionScript в фильме. В главе 15 подробно рассмотрен синтаксис, структура и элементы ActionScript. Глава 16 представляет различные интересные примеры фильмов, в которых для расширения функциональных возможностей используется язык ActionScript. Из главы 17 вы узнаете о том, как посредством передачи или приема данных сделать Flash-фильм динамичным. В главе 18 рассматриваются средства и инструменты, используемые для нахождения ошибок в сценариях ActionScript.

Часть 6. Работа со звуком

Часть 6 посвящена одному из наиболее важных аспектов в разработке мультимедийных продуктов: аудио. В главе 19 мы представим полезную информацию об основах цифровых звуковых форматов. Данная глава особенно важна для тех пользователей, у которых нет опыта работы со звуковыми файлами. Изучив основы цифровых аудиоформатов, вы перейдете к главе 20, где рассказывается о том, как работать со звуком во Flash, и как публиковать озвученные Flash-фильмы. Поскольку вопросы синхронизации звуков с различными событиями чрезвычайно важны, глава 21 посвящена изучению различных средств для синхронизации звука и изображения. Наконец, в главе 22 рассказывается о методах интерактивного контроля аудиоэлементов средствами ActionScript.

Часть 7. Интеграция Flash с другими программами

В части 7 данной книги освещены вопросы интеграции Flash со многими другими приложениями. В главе 23 рассматривается использование Flash в контексте с другим мультимедийным продуктом от компании Macromedia: Director. Из главы 24 вы узнаете, как интегрировать Flash MX с приложением FreeHand — еще одной популярной программой для работы с векторной графикой от компании Macromedia. В главе 25 описаны различные способы использования аудио-приложений других компаний-разработчиков для того, чтобы отредактировать звуки, составить мелодии и обработать цифровые аудиофайлы, прежде чем интегрировать их с Flash-фильмами. Наконец, из главы 26 вы узнаете, как интегрировать 3D-приложение с вашим Flash-фильмом. В данной главе будет рассмотрено 3D-приложение от другого разработчика, а также рассказано о методах, используемых для имитации 3D с помощью Flash-инструментов рисования и черчения.



CD-ROM, выпущенный в комплекте с книгой, содержит целую главу (в формате Adobe Acrobat PDF) об интеграции Flash с приложением Dreamweaver компании Macromedia.

Часть 8. Публикация и распространение Flash-фильмов

В части 8 данной книги внимание сфокусировано на том, как донести до зрителя ваши разработки. Из главы 27 вы узнаете, как создавать форматы Flash-фильмов, подходящие для размещения на Web-узле. Flash MX — первая версия, в которой действительно поддерживается цифровое видео. В главе 28 описаны техники, используемые для публикации Flash-фильмов как цифрового видео в форматах QuickTime и AVI. Кроме того, вы научитесь переносить ваши Flash-фильмы на кассеты формата VHS, используя приложение Adobe Premiere. Наконец, в главе 29 рассмотрены вопросы, связанные с разработкой Flash-фильмов для распространения на дисках CD-ROM.

Примеры разработки

Успех каждого творческого проекта, не исключая и фильм Flash MX, определенно зависит от вдохновения. Даже самые талантливые художники и дизайнеры иногда переживают творческий кризис.

Мы старались сделать книгу наглядной и познакомить вас с наиболее удачными примерами использования возможностей Flash, для чего в конце практически

каждой главы включили раздел «Разработки». В каждом таком разделе представлен пример Flash-фильма, технически связанного с темой главы. Например, в конце главы «Работа с текстом» представлен Web-узел www.photographic.com, где прекрасно сочетаются текст и анимация Flash. Автором этого Web-узла является Джимми Чен. Надеемся, что примеры разработок не только помогут вам в творческом процессе, но и дадут возможность почувствовать силу и мощь Flash.

Практикумы

Эта книга содержит восемь практикумов, включенных в книгу для того, чтобы дать читателю возможность научиться применять полученные знания и приобрести необходимые навыки работы с Flash. В Практикуме 1 показано, как изменять комбинации клавиш быстрого вызова команды с помощью редактора Keyboard Shortcuts. Практикум 2 представляет пошаговую инструкцию для использования инструментов рисования и черчения при создании статической сцены сада. Практикум 3 вы используете для создания небольшой анимации на тему научной фантастики. В Практикуме 4 будет рассказано о создании интерактивного резюме с применением возможностей языка ActionScript. Более глубоко эти возможности рассматриваются в практикуме 5 на примере работы с фильмом, где используется множество SWF-файлов. В Практикуме 6 демонстрируется использование приложения DoReMedia Sound Families для создания динамических звуковых дорожек к Flash-фильмам. Практикум 7 показывает, как разработать анимированный 3D-логотип и поместить его в HTML-документ с использованием приложений Amorphium Pro и Macromedia Dreamweaver. И наконец, из практикума 8 вы узнаете, как создавать совместимый с несколькими операционными системами CD-ROM, используя приложение Toast 5 Titanium.

Приложения

Приложение А содержит пошаговую инструкцию для создания всевозможных забавных элементов и декоративных приспособлений Flash-фильма. Сюда входят декоративные элементы для навигации, предварительной загрузки и элементы, функционирующие на основе массивов.

В приложении Б вы познакомитесь с синтаксисом языка ActionScript, текстовыми примерами сценариев и терминами, которые используются в языке написания сценариев. Это не полное описание языка, а лишь те его элементы, которые на наш взгляд являются *самыми важными*. Данное приложение будет очень полезно тем, кто собирается создавать сценарии и хочет освоить основы языка ActionScript.

Как пользоваться этой книгой

Книгой можно пользоваться двумя способами. Во-первых, как уже говорилось ранее, ее можно читать от начала до конца, поскольку представленный материал изложен в логической последовательности. Такой способ позволит вам твердо изучить основы, а также разобраться с некоторыми наиболее сложными средствами Flash MX.

Во-вторых, книгу можно использовать как справочник. Если вы желаете изучить какой-либо узкий вопрос, то легко сможете найти необходимый раздел, используя индекс или оглавление.

Какой бы способ вы не выбрали, надеемся, что наш труд поможет вам получить знания и навыки, необходимые для продолжения путешествия в замечательный мир Flash MX.

О прилагаемом компакт-диске

Как многие другие компьютерные издания, книга Flash MX Savvy выпускается в комплекте с компакт-диском, который можно использовать как на Macintosh, так и в Windows. Для данного компакт-диска мы постарались подобрать наиболее полезную информацию.

На диске вы найдете набор демонстрационного и пробного программного обеспечения. Большинство этих приложений (если не все) описаны в книге; некоторые из них вам понадобятся для выполнения упражнений из разделов «Практикум». Кроме того, компакт-диск содержит все необходимые файлы поддержки и примеров, используемые в главах и разделах этих «Практикумов».



Каждый абзац, при изучении которого необходимо обратиться к файлам на компакт-диске, обозначен небольшим специальным значком. Данные файлы вы можете свободно использовать в качестве основы для своих проектов.

Связь с авторами и продолжение сотрудничества

Для читателей (и потенциальных читателей) этой книги нами был разработан коллективный ресурс. Приглашаем вас посетить Web-узел www.vonflashenstein.com, чтобы узнать о лаборатории доктора Гельмута Ван Флашенштейна и его верного помощника Муви Клипа. В стенах своей лаборатории доктор Флашенштейн представит вам свои причудливые разработки, примеры фильмов и другие материалы поддержки книги.

На этом Web-узле размещены следующие материалы:

- ▶ расширенный раздел «Разработки»;
- ▶ еженедельно обновляемые примеры сценариев ActionScript и исходные файлы формата FLA;
- ▶ подписка на новости и события в мире Flash, рассылаемые по электронной почте;
- ▶ большое число учебного материала и статей, посвященных Flash.

Для связи с авторами направляйте ваши электронные сообщения по следующим адресам:

norbert@vonflashenstein.com

ethan@vonflashenstein.com

ЧАСТЬ 1

Знакомимся с Flash

Изначально программа Flash предназначалась для создания быстро загружаемой Web-анимации. Затем это средство подверглось стремительному развитию, становясь все более эффективным и мощным. В настоящее время Flash, как и прежде, широко используется для анимирования изображений, но наряду с этим его можно применять для создания сложных интерактивных схем навигации, динамических Web-узлов, мультсериалов, игр, музыкального видео, проигрывателей и т. д. Итак, программа Flash прошла путь от простого аниматора до мощного средства разработки мультимедийных продуктов. Сегодня трудно представить Интернет без Flash и всего того, что было создано благодаря этой технологии.

В первой части книги мы совершим экскурс в историю Flash, поговорим о роли программы в современном мире цифровых медиатехнологий и о перспективах ее развития. Конечно же, вам будет представлена последняя версия из семейства Flash — Flash MX. Среди ее особенностей — новый интерфейс, простой и наглядный, который в значительной степени способствует повышению эффективности работы. Вот почему данный раздел будет полезен как новичкам, так и пользователям, имеющим опыт работы с Flash.

Глава 1. Введение в Flash.....	28
Глава 2. Новинки Flash MX.....	35
Глава 3. Путешествие по интерфейсу.....	39

Глава 1

Введение в Flash MX

Приложение Flash является универсальным средством создания цифровых мультимедийных продуктов. Технология Flash, базирующаяся на векторной графике, всегда занимала лидирующую позицию среди цифровых технологий передачи быстро загружаемой анимации. Данные свойства Flash хорошо известны широкому кругу пользователей. Но этим возможности программы не ограничиваются. По мере развития приложения расширялась и область его применения. Сегодня Flash предлагает средства создания продуктов как для Интернета и современных персональных компьютеров, так и для игровых приставок, а также средств мобильной связи. Это на самом деле *мультимедиа*.

Чтобы понять, что на самом деле представляет собой технология Flash, необходимо изучить историю ее создания, составляющие компоненты и спектр возможностей. В этой главе будут освещены такие темы:

- ▶ история создания Flash;
- ▶ сравнительный анализ растрового и векторного форматов;
- ▶ области применения Flash.

История создания Flash

Перенесемся в конец 1980-х годов, когда в результате слияния в Калифорнии образовалась мощная компания Micromedia. Кто же участвовал в слиянии? Во-первых, Macromind — небольшая чикагская компания, специализировавшаяся на разработке программного обеспечения, одним из продуктов которой было приложение VideoWorks. Во-вторых, Paracomp — компания из Сан-Франциско, известная благодаря своему 3D-приложению Swivel3D, рассчитанному на Macintosh. В-третьих, компания Authorware, штат Миннесота, получившая известность после выпуска мультимедийного приложения Authorware. В 1991 году компании Macromind и Paracomp были объединены в компанию Macromind-Paracomp, а затем к ним примкнула компания Authorware.

Как вы заметили, до сих пор речь шла о трех компаниях. Основателем четвертой компании был Джонатан Гай, автор таких известных игр для Macintosh, как Dark Castle и Beyond Dark Castle. В январе 1993 года он убедил своего партнера и программиста Чарли Джексона (основателя компании Silicon Beach Software) инвестировать средства и оказать помощь в создании компании FutureWave. Планировалось, что новая компания будет разрабатывать программное обеспечение, которое доминировало бы на рынке планшетных компьютеров. Первый продукт вышел под названием Go. К сожалению, ранние планшетные компьютеры так и не стали популярными. Кроме того, нельзя не упомянуть о корпоративном противостоянии AT&T. В силу этих причин приложение Go так и не нашло своего места на рынке.

Таким образом, компания FutureWave столкнулась с серьезными трудностями: целый год ушел на разработку приложения, которое не нашло своего потребителя. Ни о расширении, ни об увеличении доходов говорить не приходилось.

Положение удалось исправить после выпуска небольшой графической программы SmallSketch, которая создавалась как дополнение к Go. Новый продукт продвигался на рынке как программа рисования, рассчитанная и на Macintosh, и на Windows. Возможно, одним из ярчайших примеров дальновидных действий разработчиков можно считать принятие решения о переделке SmallSketch из приложения, предназначенного для создания статических изображений, в программу подготовки анимации. Толчком послужило развитие Интернета. Глобальная сеть должна была стать прекрасной средой для передачи двухмерной анимации.

После того как компании Adobe и Fractal Design отказались покупать данную технологию (как, вероятно, они сожалеют об этом сейчас!), компания FutureWave подготовила собственную программу FutureSplash Animator, выход которой приходится на лето 1996 года. Это приложение предназначалось для создания линейной анимации на базе векторной графики. Данный продукт сразу привлек к себе внимание и был использован при разработке Web-версии MSN компанией Microsoft, а так же при разработке электронной версии Disney Daily Blast компанией Disney. В ноябре 1996 года FutureWave получила предложение о совместной работе от компании Macromedia. Для небольшой фирмы, штат которой состоял лишь из шести человек, это был прекрасный шанс. В декабре 1996 года компания Macromedia купила технологию FutureSplash у FutureWave, а уже в начале 1997 года выпустила первую версию Flash 1.

После передачи эстафеты компании Macromedia началось интенсивное развитие Flash. С каждой новой версией пополнялся арсенал инструментов и функций этого программного средства:

- ▶ **Flash 1.** Приложение Flash 1 было по сути переименованной версией программы FutureSplash с довольно ограниченным набором функций (с точки зрения современных стандартов) и предназначалось для анимирования векторных изображений с использованием временной шкалы. Главное достоинство данной программы заключалось в возможности воспроизведения анимации в сети с использованием подключаемого модуля Netscape и элементов управления ActiveX Internet Explorer.
- ▶ **Flash 2.** Версия Flash 2 характеризовалась рядом существенных усовершенствований. Ее выход послужил началом превращения Flash из простой анимационной программы в интерактивное медиаприложение. В программе появились возможности применения библиотечных элементов, внедренных графических изображений и векторных шрифтов, а также обработки стереозвука. Данная версия также поддерживала импорт изображений и многочисленные форматы файлов: EPS, GIF, JPEG, AutoCAD DFX, BMP, Enhanced Metafile, AIFF, Windows Metafile и Shockwave.
- ▶ **Flash 3.** Одним из наиболее значительных усовершенствований Flash 3 (наряду с улучшенным интерфейсом) следует считать внедрение действий (actions). Базирующиеся на JavaScript действия (позже на их основе был разработан язык ActionScript) обеспечили некоторый контроль над фильмами, а

также определенный уровень их интерактивности. Такие новинки, как маски, промежуточные изображения и прозрачность позволили пользователям заметно улучшить внешний вид изображений, подготавливаемых средствами Flash.

- ▶ **Flash 4.** Важнейшая характеристика Flash 4 — возможность воспроизведения сжатых MP3-файлов в фильмах. В данной версии заметно усовершенствован язык ActionScript, что открыло возможности создания интерактивных игр и интерфейсов. Заслуживают упоминания также редактируемые текстовые поля, улучшенный пользовательский интерфейс и упрощенный процесс публикации продуктов.
- ▶ **Flash 5.** Самым существенным шагом вперед в Flash 5 является усовершенствованный язык ActionScript. Приведенный в соответствие стандарту ECMA-262, новый язык позволил компании Macromedia объявить, что ее Flash и ActionScript готовы конкурировать на рынке с самыми именитыми продуктами в данной области. Другие усовершенствования касались пользовательского интерфейса. В их числе — дополнительные инструменты и панель Movie Explorer, возможности получения помощи через Интернет, обновления продуктов и создания пользовательских комбинаций клавиш.

Программа Flash преодолела длинный путь. Те свойства, которые считались революционными в одной версии, со временем казались примитивными по сравнению с мощными функциями последующей версии. Как вы вскоре убедитесь, Flash MX — это еще один шаг вперед.

FLASH MX: Что означает название?

Пользователям последней версии Flash нетрудно заметить отклонение от традиционной схемы именования продукта. Компания Macromedia приняла решение назвать новую версию «MX». Такое решение обусловлено желанием проинформировать разработчиков о том, что данное программное обеспечение предоставляет интегрированные решения для цифровых медиатехнологий на базе Интернета. Поскольку все продукты Macromedia (Flash, Dreamweaver, ColdFusion и т. д.) могут быть тесно интегрированы, компания решила, что правильно включить в имена авторских приложений общий элемент. MX — это метка, присваиваемая программным средствам семейства Macromedia. Название проигрывателя Flash Player (служит для воспроизведения фильмов Flash; может быть как встраиваемым, так и автономным), как и прежде, снабжено номером текущей версии (Flash Player 6).

Буквы MX не несут смысловой нагрузки, это лишь обозначение. Главное, что новая версия программы Flash MX является прекрасным продуктом.

Сравнительный анализ растрового и векторного форматов

Графические изображения могут храниться и выводиться в двух форматах: *векторном* и *растровом*. Чтобы понять принципы работы Flash и по достоинству оценить преимущества этой программы перед другими анимационными приложениями, необходимо изучить различия между указанными двумя форматами.

В векторном формате графическая информация представляется с помощью кривых, описываемых математическими формулами. Этот способ обеспечивает возможность трансформации изображения без потери качества. Каждый векторный объект имеет две цветовые характеристики: обводку (stroke) и заливку (fill) (рис. 1.1).

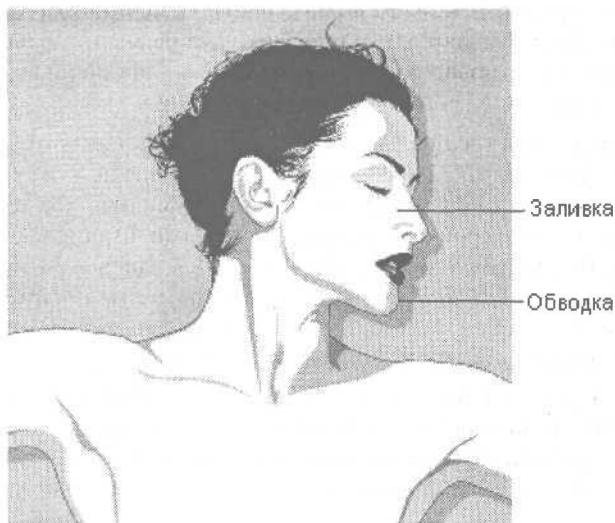


Рис. 1.1. Обводка и заливка в векторном изображении

Обводка и заливка в векторном изображении рассчитываются математически. Это очень важно при анимировании изображения.

В одном графическом объекте растровые изображения (рис. 1.2) состоят из пикселей (цветных точек), которых может насчитываться несколько миллионов. Английское слово «pixel» образовано от слов «picture» (картинка) и «element» (элемент). Можно провести аналогию между растровым изображением и мозаикой: мельчайший составной элемент занимает определенный объем памяти и влияет на внешний вид и цветовую гамму целого изображения.

Одно из важнейших различий между векторными и растровыми изображениями проявляется при их масштабировании. Поскольку элементы векторного изображения (обводки, заливки и т. д.) строятся на основе результатов вычисления математических выражений, такое изображение можно масштабировать, растягивать и изменять без ущерба для его четкости и разрешения. Выполнение перечисленных операций по отношению к растровым изображениям приводит к снижению их качества из-за того, что изменяется сетка, в узлах которой размещены пиксели. При увеличении размера растрового изображения компьютер должен интерполировать (с помощью различных алгоритмов) дополнительные пиксели, необходимые для формирования модернизированной сетки. В результате на увеличенном растровом рисунке могут возникнуть нежелательные фрагменты либо может проявиться эффект «ступенек».

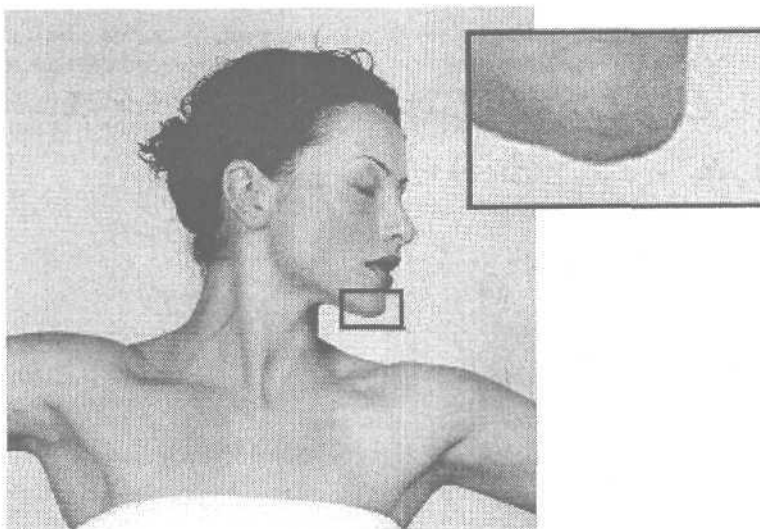


Рис. 1.2. Растровое изображение состоит из мозаичных цветных элементов, пикселей, размещенных в узлах сетки

Области применения Flash

Как указывалось, Flash — это не просто средство анимации. Тогда что же? Вместо того чтобы давать обширное определение, предлагаем сразу приступить к рассмотрению основных возможностей этой программы.

Использование Flash в электронных средствах массовой информации

Хотя приложение Flash, возможно, больше всего приспособлено для работы в Web, в последние годы его все чаще используют в электронных средствах массовой информации. И это не удивительно. В конце концов, Flash имеет репутацию мощного средства анимирования, а его файлы можно экспортировать в различные форматы.

Технология Flash применялась в нескольких электронных проектах как в Европе, так и в США. Например, в Великобритании компания Kerb, специализирующаяся в области Web-дизайна, создала одну из первых серий мультфильмов, полностью подготовленных с помощью технологии Flash. Более подробную информацию об этой серии, повествующей о приключениях шумной компании говорящих овощей, вы найдете в практикуме главы 28.

ПРИМЕЧАНИЕ

В США приложение Flash использовалось при разработке заставки для шоу The Rosie O'Donnell. За более детальной информацией обращайтесь по адресу www.macromedia.com/proom/2000/converge.html.

Flash и системы мобильной связи

Компьютеры приобретают все меньшие размеры и становятся все более приспособленными для мобильной связи. Flash не отстает от этого процесса. Приложение Flash Player способно функционировать с широкой гаммой Web-совместимых устройств и аппаратных средств. На этих устройствах Flash Player поддерживает работу бизнес-приложений, позволяет просматривать рекламу, новости и запускать развлекательные приложения.

Одной из первых платформ, поддерживавших Flash, была система Microsoft Pocket PC. Детальная информация о Flash для Pocket PC опубликована по адресу www.macromedia.com/software/flashplayer/pocketpc.

Приложение Flash способно также функционировать на мобильных телефонах и других переносных вычислительных устройствах. Сведения об этом вы почерпнете с узла, расположенного по адресу www.macromedia.com/software/flashplayer/resources/devices.

Применение Flash в игровых приставках

Совершенно неожиданно технология Flash нашла себе применение в развлекательной индустрии, точнее — в игровых приставках.

Примером может служить пользовательский интерфейс игры Starfighter от компании LucasArts для приставок PlayStation 2 и Xbox. Разработав интерфейс начала игры для Starfighter, компания столкнулась с серьезными проблемами на заключительном этапе, когда стало очевидно, что не хватает функционального пользовательского интерфейса выхода из игры. Такой интерфейс был создан с использованием Flash при сотрудничестве компаний Secret Level (www.secretlevel.com) и Orange Design (www.orangedesign.com). Выпуск игры в феврале 2001 года стал важным этапом в истории развития Flash. Популярность новой технологии стремительно возросла, что способствовало дальнейшему ее развитию. О том, как использовалась технология Flash в Star Wars: Starfighter, рассказывается в статье www.gamasutra.com/features/20010801/corry_01.htm.

В мае 2001 года компании Macromedia и Sony приступили к разработке проигрывателя Flash Player для игровой приставки Sony PlayStation 2. Основной целью было создание интерфейса начала игры, как в Star Wars: Starfighter.

Flash в Интернете

Изначально программа Flash предназначалась для разработки анимированного наполнения Web-узлов. И какие бы новые горизонты ни открывались перед этой технологией, будь то коммуникации, реклама или интерактивные приложения, Flash всегда останется инструментальным средством для Web. Подтверждением послужат многочисленные примеры, предлагаемые в нашей книге, которые продемонстрируют, насколько обширны возможности Flash в области Web-дизайна.

К традиционным областям применения Flash, таким как передача информации и развлечения, можно добавить еще одну — рекламу. Узнать больше о том, как Flash используется в рекламе, вы сможете, прочитав статью по адресу www.macromedia.com/resources/richmedia/tracking.

Web-игры и мультфильмы

Flash действительно является эффективным средством разработки мультимедийных продуктов и интерактивных приложений, но ему *нет равных* среди инструментов по созданию развлекательного наполнения Web.

Не будем создавать обманчивый образ серьезного приложения: следует признать, что большинство разработчиков используют Flash для чего угодно, но только не для решения серьезных задач. Можно смело утверждать, что сегодня одна из основных областей применения Flash — игры и мультипликация. Настоящая книга содержит несколько замечательных примеров, подтверждающих этот факт.

Разработка киосков

Киоском называют пользовательское аппаратное средство, применяемое для передачи порций информации посредством дружелюбного интерфейса. Киоски, как правило, используются для предоставления пользователям доступа к информации (чаще всего с помощью сенсорного или управляемого мышью интерфейса), предназначенной только для специальных служащих. Поскольку Flash позволяет создавать интерактивные приложения, в которых сочетаются привлекательный интерфейс и многофункциональность, вполне естественно, что эту программу применяют для создания киосков.

Перечень киосков, разработанных с помощью Flash, слишком велик, чтобы публиковать его в книге. Мы ограничимся одним примером, который, по правде говоря, выделяется из общего ряда и представляется наиболее интересным. В конце 2000 года на выставке CeBit, которая ежегодно проводится в Ганновере (Германия), компания *Моссси* представила прототип заправочной колонки с сенсорным экраном. Разработка позволяет водителям получать информацию о маршрутах движения и ситуациях на дорогах, делать небольшие покупки и даже отправлять электронные сообщения во время заполнения баков горючим. Хотя данная система пока не нашла применения на практике, она является прекрасным примером использования продуктов Flash в некомпьютерной среде.

Резюме

Надеемся, по прочтении этой главы вы получили представление о возможностях Flash и теперь полностью готовы к увлекательному путешествию в мир этой замечательной технологии. Удачи!

Глава 2

Новинки Flash MX

Добавив множество мощных и полезных инструментов, Macromedia в очередной раз расширила границы возможностей своего продукта. В настоящей главе дан обзор самых интересных нововведений. В процессе их изучения будут затронуты следующие темы:

- ▶ улучшенный процесс разработки и практичность;
- ▶ новые инструменты для создания визуальных эффектов;
- ▶ расширенные медиавозможности;
- ▶ новые возможности ActionScript;
- ▶ новые средства публикации и распространения.

Улучшенный процесс разработки и практичность

Хорошее программное обеспечение удобно в работе и легко поддается настройке. Чтобы обеспечить соответствие Flash этому определению, компания Macromedia ввела в данный программный продукт следующие новшества:

Новый пользовательский интерфейс. Новый интерфейс Flash MX, все панели которого можно перемещать, растягивать и настраивать, в значительной степени способствует повышению производительности работы и позволяет сконцентрироваться на самом важном — процессе разработки. Подробное рассмотрение элементов интерфейса Flash MX содержится в главе 3.

Усовершенствованная временная шкала. Самый важный инструмент создания анимации — временная шкала — теперь снабжен такой новинкой, как каталог слоев. Детальную информацию об улучшенной временной шкале вы найдете в главе 9.

Шаблоны. Flash MX предоставляет в распоряжение пользователей библиотеки шаблонов, которые избавят от необходимости выполнения многих рутинных операций, сопровождавших ранее начало создания каждого нового фильма. В виде шаблона может быть сохранен готовый фильм.

Панель Properties. Многочисленные панели из версии Flash 5 теперь объединены в панель Properties, которая представляет собой динамический инструмент, обеспечивающий централизованное редактирование всех параметров фильма. Более близкое знакомство с панелью Properties состоится в главе 3, кроме того, мы будем неоднократно возвращаться к ней на протяжении всей книги.

Редактирование символов. Прошли те времена, когда для создания и редактирования символов приходилось оставлять основную временную шкалу. Теперь вы можете работать с символами непосредственно в главном окне. Вопросы редактирования символов освещаются в главе 7.

Усовершенствование совместно используемой библиотеки. Все пользователи Flash MX могут вооружиться новыми средствами библиотеки Shared Library, которые интегрируются в фильм либо во время воспроизведения (при подключении к Web-серверу), либо в заданное автором время (при создании библиотеки на локальном компьютере). Полный обзор усовершенствований библиотеки Shared Library представлен в главе 7.

Панель Components. Компоненты сослужат хорошую службу разработчикам, которые создают фильмы с высокой степенью интерактивности (управляемым содержимым). Они позволяют добавлять элементы управления стандартного интерфейса — флажки, ниспадающие меню, меню и панели с прокруткой, кнопки, переключатели, полосы прокрутки — путем выполнения простейшей процедуры перетаскивания мышью. Flash MX предоставляет возможность настройки панели компонентов, а также разрешает создавать новые компоненты для решения специфических задач. Обстоятельный рассказ о панели компонентов предлагается вниманию читателей в главе 16.

Уроки и средства обучения. Компания Macromedia, предполагая, что изучение новой версии Flash окажется непростой задачей, усовершенствовала интерактивные средства обучения. Информацию о возможностях справочной системы предоставляет команда Tutorials меню Help.

Новые инструменты для создания визуальных эффектов

В помощь своим пользователям Flash MX предлагает множество инструментов для рисования и разработки проектов:

Панель Color Mixer. Поскольку выбор правильного цветового решения является одним из наиболее важных аспектов при создании фильма, компания Macromedia усовершенствовала цветовой микшер, благодаря которому процедуры смешивания и использования цветов и градиентов стали намного проще. Подробное описание панели Color Mixer дано в главе 4.

Инструмент Free Transform. Это эффективный и универсальный инструмент для преобразования объектов и манипулирования ими. Вращение, искажение и масштабирование — все эти операции можно осуществлять с помощью инструмента Free Transform. Подробные сведения — в главе 6.

Инструмент Envelope Modifier. Данный инструмент, называемый также Edit Envelope, представляет собой упрощенный вариант инструмента Free Transform. Он служит для коррекции цвета и формы конверта объекта, а следовательно — и самого объекта. Более подробно об инструменте Envelope Modifier мы расскажем в главе 6.

Команда Distribute to Layers. А это средство особенно удобно при создании сложной анимации: оно позволяет перенести любое число выбранных объектов на их собственные автоматически создаваемые слои. Детальные сведения об этом вы почерпнете из главы 8.

Инструмент Text. Инструмент Text в новой версии Flash MX (в сочетании с элементами панели Properties) обеспечивает возможность полного контроля того, как будет выглядеть текстовая информация в фильме. Благодаря новым возможностям текст можно расположить по вертикали, что удобно в случае использования языков с иероглифической системой письма. Полный обзор возможностей усовершенствованного инструмента Text дан в главе 5.

Расширенные медиавозможности

Что такое Flash без медиавозможностей? Flash MX поддерживает множество медиаформатов:

Аудиоформаты. Программа Flash MX снабжена дополнительными элементами ActionScript, которые значительно расширяют возможности при озвучивании Flash-фильмов. Объект Sound обеспечивает предварительную загрузку отдельных потоковых звуковых файлов, делает возможным мониторинг продолжительности звука и позиции воспроизведения, а также управляет событиями при загрузке звуковых файлов или по окончании воспроизведения. Подробно об инструментах Flash MX, предназначенных для работы со звуком, мы расскажем в части VI.

Внешние графические файлы. Flash MX позволяет загружать JPEG-файлы из внешних источников, что особенно удобно разработчикам, создающим интерактивные слайд-шоу и другие фильмы с большим числом картинок. Описание процесса загрузки JPEG-изображений содержится в главе 17.

Видео. Впервые Flash предлагает реальную поддержку видео. Программа способна импортировать информацию во многих цифровых видеоформатах для встраивания в фильм Flash или связывания с ним. Для воспроизведения встроенного видео необходим кодек Sorenson Spark, который позволяет воспроизводить видео в рамках Flash-фильма. Управление видео как элементом фильма осуществляется с помощью языка ActionScript. Работе с видео посвящена глава 28.

Новые возможности ActionScript

Язык сценария ActionScript для версии MX является, возможно, одним из наиболее мощных инструментов, доступных разработчикам при создании фильмов. Значительные усовершенствования этого языка предопределили его применение для решения самых сложных задач.

Далее перечисляются некоторые усовершенствованные средства ActionScript:

Улучшенная среда разработки. В Flash MX представлена новая панель Actions, которая настолько удобна в применении, что понравится даже самым придирчивым разработчикам. Усовершенствования коснулись также панели инструментов ActionScript, системы подсказок, генерируемых при написании сценария, настраиваемого окна сценария и окна Debugger, в котором производится проверка сценария в построчном режиме.

Написание сценариев для текста и текстовых полей. Flash MX предлагает множество элементов ActionScript, предназначенных для манипулирования полями

и изменения параметров форматирования текста. Эти модернизированные средства в значительной степени расширяют возможности вывода динамического текста в фильме.

Рисование с помощью объекта Movie Clip. Усовершенствованный объект Movie Clip теперь имеет несколько новых методов, позволяющих рисовать и чертить, используя ActionScript. За детальной информацией по этой теме обращайтесь к главе 16.

Написание сценария для кнопок. Кнопки в Flash MX являются объектами и могут быть дополнены элементами ActionScript, повышающими их функциональность и интерактивность. Подробнее об объекте Button рассказывается в главе 16.

Поддержка воспроизведения. Для управления воспроизведением фильма и изменения его содержимого в соответствии с требованиями разработчика созданы новые объекты — Capabilities и Stage, полные описания которых включены в главу 16.

Новые возможности публикации и распространения

В новой версии Flash MX расширены возможности публикации.

Создание фильмов с возможностью доступа. Используя панель Accessibility, вы можете создавать и публиковать фильмы, которые будут доступны тем пользователям, которые используют адаптивную технологию, например screen reader. Эта тема освещена в главе 27.

SWF-сжатие. Кодек для сжатия Z-Lib позволяет значительно сократить время загрузки SWF-файлов со сложным содержанием.

Сохранение в формате Flash 5. Многие пользователи все еще работают с Flash 5. Благодаря команде Save as Flash 5 обладатели Flash MX смогут обмениваться продуктами с коллегами, не обновившими версию своей программы.

Резюме

Данная глава представляет собой обзор новых возможностей и средств, предоставляемых Flash MX. Усовершенствования коснулись главным образом процесса разработки, инструментов для создания визуальных эффектов, медиасредств, языка ActionScript и возможностей распространения фильмов. В следующей главе мы на практике испытаем новые элементы интерфейса Flash MX.

Глава 3

Интерфейс

В результате выполнения двойного щелчка на значке программы Flash появляется множество панелей и окон. Не следует пугаться изобилия различных элементов. Прочитав данную главу, вы сможете не только легко ориентироваться в интерфейсе Flash, но и производить его настройку. Знакомство с интерфейсом мы начнем с изучения главного меню программы и панелей инструментов. Затем последуют описания области действия и временной шкалы — элементов, используемых для создания конечных продуктов. После этого вашему вниманию будет предложен обзор специальных панелей, назначение которых — ускорение процесса разработки. В завершение мы поговорим о том, как производить настройку интерфейса в соответствии с индивидуальными требованиями пользователя, и познакомимся со справочной системой Flash.

В главе рассматриваются следующие темы:

- ▶ главное меню программы;
- ▶ инструментальные панели;
- ▶ панель инструментов Tools;
- ▶ область действия и рабочая область;
- ▶ временная шкала Timeline;
- ▶ панель Properties;
- ▶ работа с панелями и окнами;
- ▶ справочная система.

Освоение рабочей среды Flash

Одно из важнейших достоинств Flash MX — усовершенствованный интерфейс, содержащий невероятное число инструментов, доступ к которым осуществляется легко и быстро. Богатый выбор специальных и рабочих стилей нового интерфейса обеспечивает максимально эффективное использование возможностей программы по созданию фильмов. Давайте вкратце рассмотрим инструменты, которые вы будете использовать.

ПРИМЕЧАНИЕ

По умолчанию программа Flash загружается со стандартной конфигурацией инструментов. Опыт, необходимый для пользовательской настройки интерфейса, вы приобретете по прочтении настоящей главы.

Строка главного меню

Как и во многих других приложениях, в Flash строка главного меню расположена в верхней части окна программы (рис. 3.1) и служит для получения доступа к различным командам.

Строка
главного
меню
программы

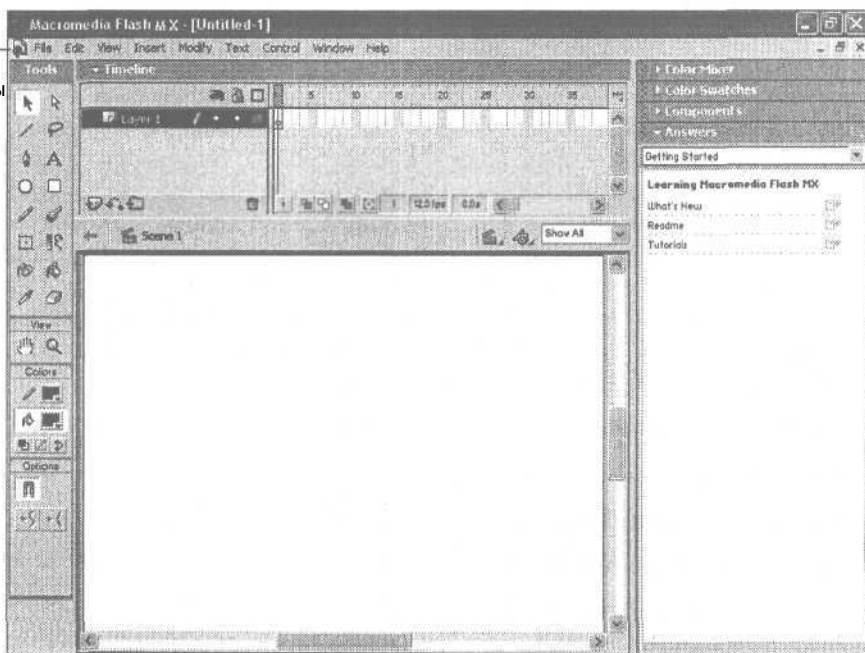


Рис. 3.1. Строка главного меню программы

Со многими командами связаны горячие клавиши, которые являются настраиваемыми. Подробно процесс настройки горячих клавиш мы рассмотрим в разделе «Выбор комбинации клавиш для вызова команды» этой главы. Получить практические навыки в настройке горячих клавиш позволит упражнение, приведенное в конце главы.

Команды меню File предназначены для работы с файлами. Работа с любым проектом начинается и завершается активизацией одной из этих команд. Кроме команд, которые являются стандартными для всех приложений, в этом меню находятся команды, специфичные для Macromedia Flash MX: Open as Library (Открыть как библиотеку), Revert (Вернуться к прежнему состоянию), Import to Library (Импорт в библиотеку), Export Movie (Экспорт видео), Export Image (Экспорт графики), Publish Settings (Параметры публикации), Publish Preview (Предварительный просмотр публикации) и Publish (Публикация).

Меню Edit содержит команды, позволяющие выполнять как стандартные операции редактирования в процессе обработки данных, так и специализированные команды Flash, предназначенные для копирования, перемещения и удаления кадров, правки символов, назначения комбинаций клавиш и отображения шрифтов.

В меню View в распоряжение пользователя предоставляются команды для управления внешним видом фильмов, а также инструментальными панелями.

Меню Insert предназначено для добавления элементов в процессе разработки Flash-анимации, а меню Modify позволяет их модифицировать. Это такие элементы как символы, кадры, слои, различные фигуры и элементы управления.

Назначение команд меню Text — редактирование и упорядочение текста. В состав этого меню входят команды для выбора начертания и размера шрифта, выравнивания текста и тому подобные.

Меню Control содержит все необходимые команды для управления воспроизведением фильмов Flash, в том числе: Play (Воспроизведение), Rewind (Перемотать), Go to End (Переход в конец), Step Forward (Шаг вперед), Step Backward (Шаг назад), Test Movie (Тестирование Фильма), Debug Movie (Отладка Фильма), Test Scene (Тестирование сцены), Loop Playback (Автоматический повтор), Play All Scenes (Воспроизведение всех сцен), Enable Simple Frame Actions (Разрешить действия простых кадров), Enable Simple Buttons (Разрешить простые кнопки), Mute Sounds (Выключить звук) и Enable Live Preview (Разрешить прямой предварительный просмотр).

Меню Window позволяет получить доступ к наиболее важным инструментам Flash — панелям и диалоговым окнам. Список открытых в настоящий момент окон отображается в нижней части меню.

Доступ к инструментальным панелям в среде Windows

Инструментальные панели Flash доступны только при работе в среде Windows (к сожалению, версия Flash MX для Macintosh лишена инструментальных панелей). Инструментальные панели содержат кнопки вызова популярных команд меню.

Инструментальная панель Main

Панель Main, доступ к которой предоставляет команда Window ▶ Toolbars ▶ Main, похожа на панель Edit ▶ Production многих графических программ. Данная панель по умолчанию содержит следующие инструменты (слева направо): New (Создать), Open (Открыть), Save (Сохранить), Print (Печать), Print Preview (Предварительный просмотр), Cut (Вырезать), Copy (Копировать), Paste (Вставить), Undo (Отменить), Redo (Повторить), Snap to Objects (Привязка к объектам), Smooth (Сглаживание), Straighten (Выравнивание), Rotate (Поворот), Scale (Масштаб) и Align (Выровнять).



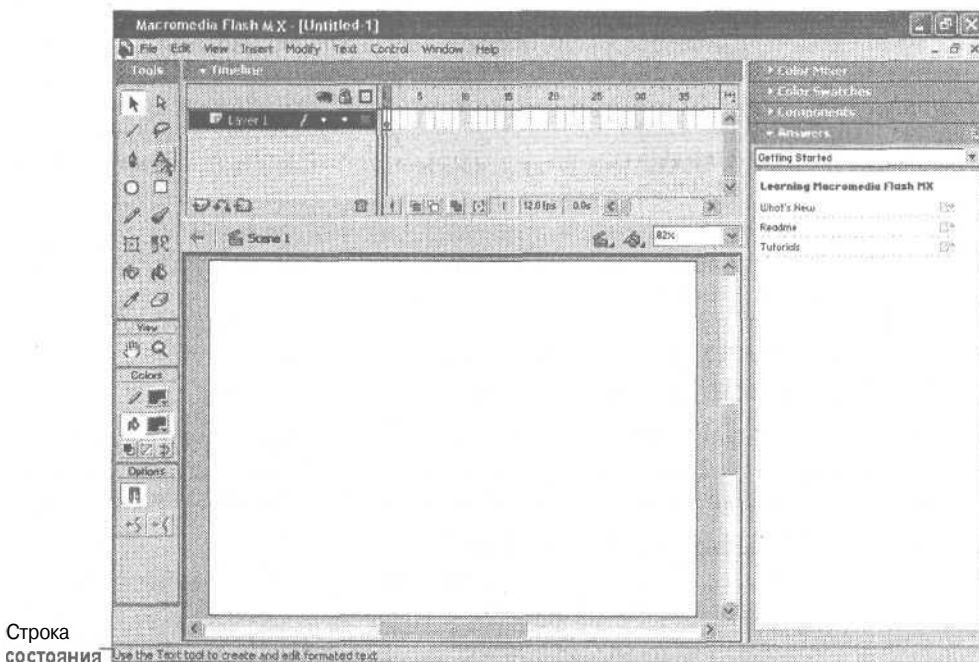
Инструментальная панель Main изначально располагается над временной шкалой и может быть перемещена в любое место окна программы. Переместить панель несложно: нужно установить указатель мыши в ее свободной области (за пределами кнопок) и, удерживая левую кнопку манипулятора нажатой, перетащить панель в надлежащую позицию.

ВНИМАНИЕ

Многие команды инструментальной панели Main, такие как Copy, Paste, Smooth, Straighten, Rotate и Scale, доступны только в том случае, если в области действия выбран объект.

Строка состояния

В отличие от панели Main, для панели Status (Строка состояния) отведена область в нижней части окна программы. Основное предназначение панели Status — получение информации о различных элементах интерфейса Flash. Сделать панель видимой позволяет команда Window ▸ Toolbars ▸ Status.



Строка
состояния

Рис. 3.2. Строка состояния

Инструментальная панель Controller

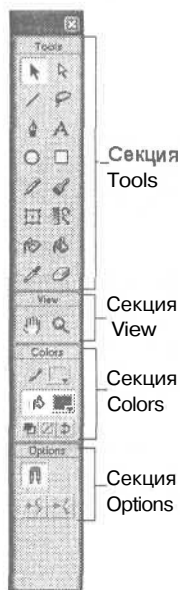
Панель Controller, загружаемая посредством команды Window ▸ Toolbars ▸ Controller, содержит серию кнопок (напоминающих кнопки проигрывателя), которые позволяют управлять фильмами и тестировать их с использованием Flash Movie Editor (Редактор фильмов Flash). Панель Controller включает следующие кнопки



(слева направо): Stop (Остановка), Rewind (Перемотка), Step Back (Шаг назад), Play (Воспроизведение), Step Forward (Шаг вперед) и Go to End (Переход в конец).

Как и многие другие инструментальные панели, панель Controller легко перемещать. Чтобы изменить положение панели, поместите указатель мыши в любое место за пределами кнопок, а затем, удерживая левую кнопку манипулятора, перетяните ее в другую позицию.

Знакомство с панелью инструментов Tools



Представьте, что вы оказались в обществе художника и мастера на все руки. Таким мастером является панель Tools — набор инструментов как для создания художественных произведений, так и для их ремонта и модификации.

Панель инструментов Tools загружается с помощью команды Tools меню Window. Данная панель состоит из четырех секций с инструментами для рисования, черчения, выбора и модификации. Давайте рассмотрим каждую секцию в отдельности.

Секция Tools

Секция Tools содержит инструменты черчения, рисования и выбора. Вопросы о том, как функционируют эти инструменты, отложим до следующих глав.

Секция View

Секция View содержит инструменты, необходимые для изменения масштаба и панорамирования Flash-фильма.

Инструмент Hand полезен в тех случаях, когда выбранный масштаб области действия не позволяет просматривать картинку целиком. «Что такое область действия?» — спросите вы. Ответ дается в разделе «Область действия и рабочая область».

Чтобы панорамировать Flash-фильм, выполните следующие действия.

1. Выберите инструмент Hand в секции View либо нажмите клавишу H.
2. Установив указатель мыши в любой позиции области действия, нажмите левую кнопку манипулятора и, удерживая ее, перетащите область действия в нужном направлении. По окончании процесса отпустите кнопку мыши.

Далее представлена последовательность действий для увеличения масштаба изображения определенной области в фильме:

1. Активизируйте инструмент Zoom из секции View либо нажмите клавишу M или Z.
2. Разместите инструмент Zoom в той части области действия, которую вы хотите увеличить. Нажав и удерживая левую кнопку мыши, переместите указатель таким образом, чтобы линии охватили интересующую вас зону. Отпустите кнопку мыши.

Чтобы уменьшить масштаб изображения при выполнении описанных действий, следует нажать и удерживать клавишу Option (Macintosh) или Alt (Windows). Вместо плюса, находящегося рядом с указателем мыши, появится минус.

СОВЕТ

Альтернативное средство уменьшения масштаба изображения — кнопка Reduce (Уменьшение), которую вы найдете в секции Options панели инструментов Tools, если выбран инструмент Zoom.

Секция Colors

Данная секция используется исключительно для управления цветами как обводки, так и заливки объектов. Инструменты секции Colors дублируются инструментами Stroke (Обводка) и Fill (Заливка) панели Properties. Подробно об этих инструментах мы поговорим в главе 4.

Секция Options

Секция Options, в отличие от других секций панели инструментов Tools, не имеет постоянного содержимого. В ней выводятся дополнительные элементы для инструментов, выбранных в других секциях. Например, при выборе инструмента Rectangle (Прямоугольник), который находится в секции Tools, в секции Options появляется кнопка Round Rectangle Radius (Радиус углов закругленного прямоугольника). Подробно о модификаторах рассказывается в последующих главах.

Область действия и рабочая область

Область действия Flash — это белая зона, где оживают элементы фильма, где происходит то, что увидят зрители.

Серая зона, окружающая область действия, называется рабочей областью. Здесь протекает закулисная жизнь, если развивать театральную тему.

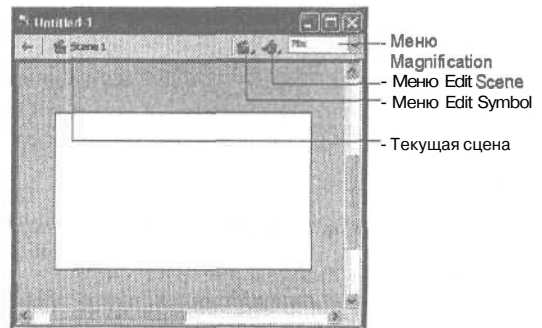
Элементы могут быть размещены в обеих областях. Разница состоит в том, что элементы, находящиеся за пределами области действия, не видны в фильме. В остальном возможности по работе с объектами одинаковы в обеих областях. Благодаря такому разграничению можно создавать анимацию, в которой некоторый объект (например, сфера) появляется постепенно перед глазами зрителей. На самом деле он начинает свой путь в рабочей области и заканчивает движение в области действия. На рис. 3.3 показаны область действия и рабочая область.

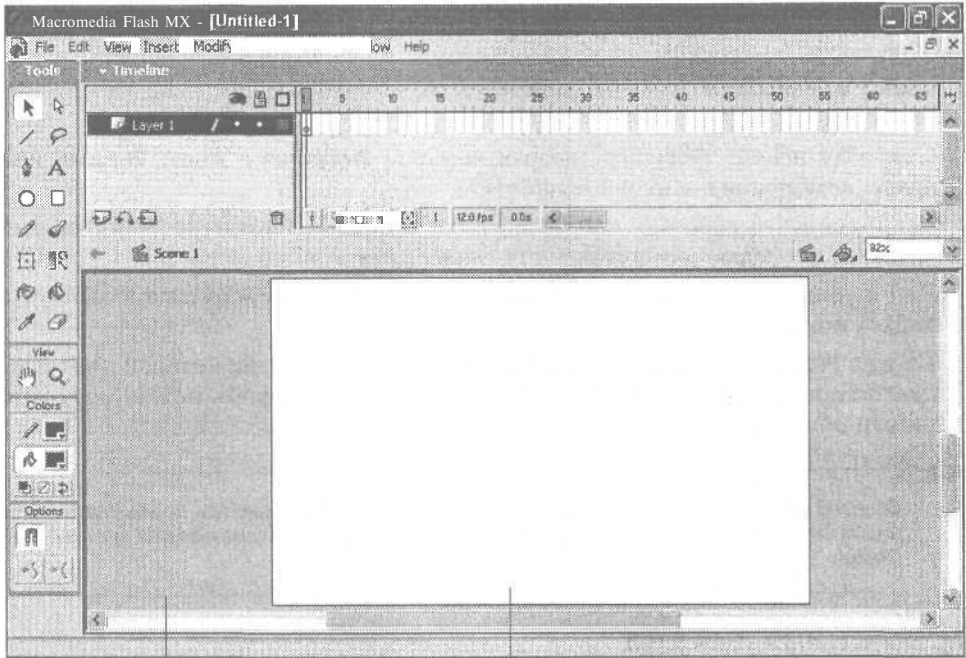
Сделать рабочую область видимой позволяет команда Work Area из меню View. Альтернативные средства — комбинации клавиш `Cmd+Shift+W` (Macintosh) и `Ctrl+Shift+W` (Windows).

После активизации команды Work Area рядом с ее названием в меню появляется метка, указывающая, что рабочая область в данный момент видима.

Строка Scene and Symbol (Сцена и символ) расположена непосредственно над рабочей зоной. В ней выводятся сообщения о том, с какой сценой или с каким символом ведется работа в данный момент. Эта строка содержит два меню — Edit Scene (Правка сцены) и Edit Symbol (Правка символа), которые позволяют осуществлять переход от сцены к сцене и от символа к символу.

Обратите внимание на то, что в правом верхнем углу строки имеется меню Magnification, предназначенное для выбора масштаба сцены.





Рабочая область

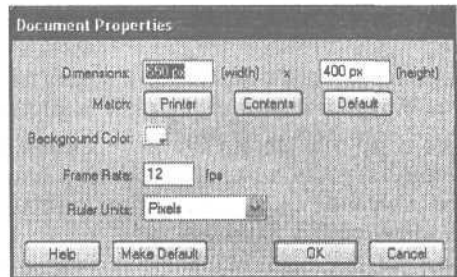
Область действия

Рис. 3.3. Область действия и рабочая область

Изменение размера области действия

При запуске программы область действия имеет вид чистого листа бумаги, ожидающего, когда пользователь начнет создавать и анимировать объекты. Безусловно, программа Flash не стала бы такой популярной, если бы все фильмы создавались в области действия одного размера. Итак, прежде чем приступить к разработке проекта, поговорим о том, как задавать размер области действия:

1. Активизируйте команду Document меню Modify или нажмите клавиши в комбинации **Cmd/Ctrl+J**, чтобы открыть окно свойств документа.
2. В поля Width и Height введите значения ширины и высоты в пикселах. Нажмите кнопку ОК. Размеры области будут тут же изменены.



Изменение цвета области действия

Фон фильма имеет тот цвет, который установлен для фона области действия. Процедура его изменения та же, что и при выборе размера для области действия, только вместо указания значений в полях Width и Height надлежит щелкнуть

сначала на палитре Background Color (Цвет фона), а затем — на понравившемся образце цвета.

Внешний вид области действия можно также изменить с помощью панели Properties:

1. Откройте панель Properties, выбрав команду Properties в меню Window, или нажав комбинацию клавиш Cmd/Ctrl+F3.
2. Если в области действия нет выделенных объектов, на панели Properties отобразятся элементы для управления свойствами этой области.
3. Для изменения фонового цвета щелкните на палитре Background и выберите подходящий цвет.
4. Панель Properties содержит также кнопку Size, щелчок на которой вызывает диалоговое окно Document Properties, с помощью которого можно изменить размер области действия.

СОВЕТ

Если вы хотите, чтобы сделанные вами изменения устанавливались по умолчанию для новых фильмов Flash, в диалоговом окне Document Properties щелкните на кнопке Make Default.

Временная шкала

Если область действия — это место, где все «происходит», то временная шкала (см. главу 9) — это то, что «управляет» действием. Временная шкала представляет собой инструмент, позволяющий анимировать изображение.

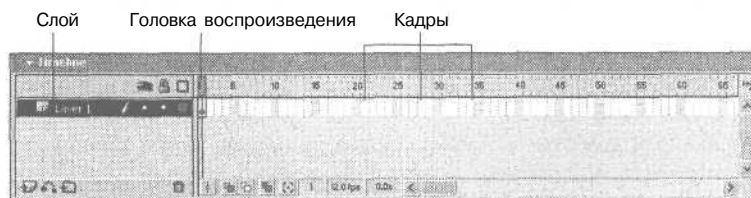


Рис. 3.4. Временная шкала

Временная шкала занимает верхнюю часть окна Flash и состоит из трех основных элементов: головки воспроизведения (Playhead), кадров (Frames) и слоев (Layer).

Как и другие элементы интерфейса Flash MX, временная шкала может быть свернута, расположена автономно или вдоль границы другого элемента. Детальное описание приемов позиционирования временной шкалы вы найдете в разделе «Настройка панелей» этой главы. (Хотя временная шкала не является панелью, работа с ней осуществляется по тем же принципам.)

Головка воспроизведения

При демонстрации фильмов Flash головка воспроизведения перемещается по временной шкале (по горизонтали) с постоянной скоростью. Во время прохождения по очередному кадру его содержимое выводится в области действия, за счет чего создается анимация.

Головку воспроизведения можно перемещать вручную. Для этого необходимо либо выбрать кадр, щелкнув на верхней полосе временной шкалы, на которой размечены кадры с шагом 5, либо перетащить головку воспроизведения в требуемое положение традиционным способом с помощью мыши.

Кадры

Кадры — это душа временной шкалы. Каждый кадр анимации представляет определенный момент времени. Содержимое изменяется от кадра к кадру. При перемещении головки воспроизведения вдоль временной шкалы содержимое каждого кадра выводится в области действия.

Частота смены кадров определяет фактическую скорость воспроизведения анимации. Более полно данная тема освещается в главе 9.

Слои

Анимация Flash имеет не только горизонтальную структуру (кадры), но и вертикальную — слои. Фильм состоит из множества слоев, содержащих различные его элементы.

Слои позволяют разделить содержимое. В особо сложных Flash-фильмах это дает вам возможность следить за ходом процесса разработки. Подробно о слоях и об их использовании мы расскажем в главе 8.

Знакомство с панелью Properties

MX Панель Properties, новинка Flash MX, несомненно, станет наиболее часто используемым средством разработки фильмов. На данную панель возложены функции проводника при доступе к свойствам различных объектов (будь то текст, форма, кнопка или клип).



Как и другие элементы интерфейса Flash MX, панель Properties может быть свернута, расположена автономно или вдоль границы другого элемента. Детальное описание приемов позиционирования данной панели вы найдете в разделе «Настройка панелей» этой главы.

Огромное достоинство панели Properties состоит в том, что она является динамической. Это означает, что представленные на ней параметры изменяются в зависимости от того, какой объект выбран. Например, при работе со строкой текста можно изменить ее шрифт, цвет и размер, а при работе с фигурой — обводку, заливку и размеры. Специфика параметров каждого выделенного объекта влияет на состав элементов панели Properties. Загрузить панель Properties позволяет команда Properties меню Window, клавиатурный аналог которой — Cmd/Ctrl+F3.

СОВЕТ

Для получения доступа к параметрам объекта, которые реже подвергаются модификации, щелкните на значке со стрелкой в нижнем правом углу панели Properties.

Панели Flash

Панели, безусловно, представляют собой важнейшие компоненты интерфейса Flash MX. Благодаря им мы получаем доступ к полным наборам инструментов, объединенных по функциональному или иному признаку.

Особенностью панелей в среде Flash MX является то, что они могут быть не только плавающими, но и интегрированными в общий интерфейс за счет *стыковки*. Благодаря стыковке облегчается доступ к инструментам и информации при максимальном увеличении рабочего пространства (рис. 3.5).

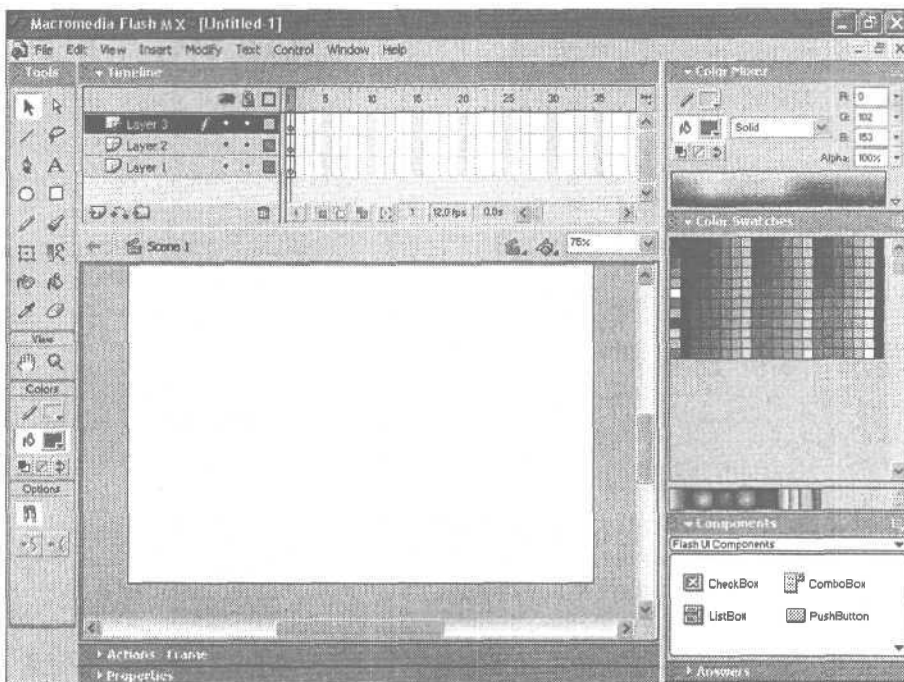


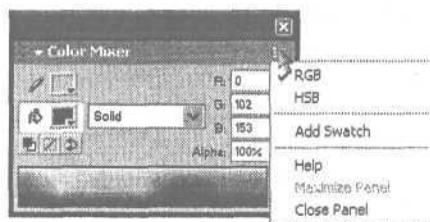
Рис. 3.5. Установленный по умолчанию способ организации панелей

О процессе стыковки (а также о различных нюансах при работе с панелями) вы узнаете из раздела «Настройка панелей» этой главы.

Освоив базовые операции, вы легко сможете открывать, закрывать, передвигать и масштабировать панели и таким образом оптимально настраивать интерфейс для решения рабочих задач.

Многие панели снабжены выпадающим меню Options, доступ к которому можно получить, щелкнув на значке в верхнем правом углу. Число пунктов в этом меню специфично для каждой панели.

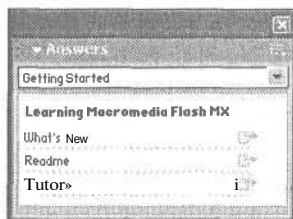
Давайте рассмотрим каждую панель в отдельности.



При первом запуске Flash действует способ организации панелей, заданный по умолчанию. Дополнительно загружаемые панели не состыкуются и могут быть свободно перемещены в пределах рабочего пространства. Вскоре вы узнаете, как устанавливать удобный для себя способ упорядочения панелей.

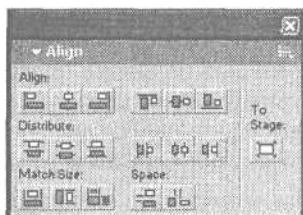
Панель Answers

MX Панель Answers, загружаемая командой Answers меню Window, обеспечивает централизованный доступ к систематизированным документам справочной системы Flash MX (Tutorials (Обучение), What's New (Новые возможности), Readme (Ознакомительная информация) и т. д.).



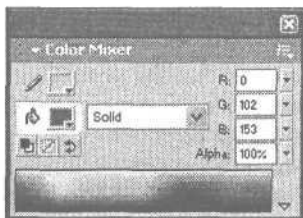
Панель Align

Панель Align, доступ к которой предоставляет команда Align меню Window (клавиатурный аналог — **Cmd/Ctrl+K**), позволяет выровнивать объекты (или группы выбранных объектов) в соответствии с рядом установленных критериев, каждый из которых представлен отдельной кнопкой. Подробно о выравнивании объектов с использованием панели Align мы расскажем в главе 6.



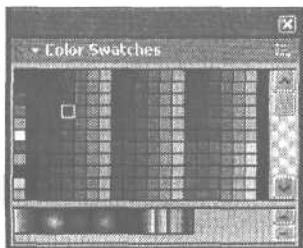
Панель Color Mixer

Панель Color Mixer, открываемая посредством одноименной команды меню Window или при нажатии клавиш **Shift+F9**, предназначена для создания цветов с использованием модели RGB или HSB либо шестнадцатеричного кода, а также для их последующего сохранения в виде образцов в каталоге. Панель Color Mixer позволяет также изменять цвета заливки или обводки объектов. Детальная информация о данной панели содержится в главе 4.



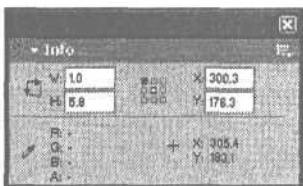
Панель Color Swatches

Назначение панели Color Swatches — организация цветов текущей палитры, их загрузка, сохранение и удаление. Открыть ее можно с помощью команды Color Swatches меню Window или комбинации клавиш **Ctrl+F9**. Подробнее о панели Color Swatches рассказывается в главе 4.



Панель Info

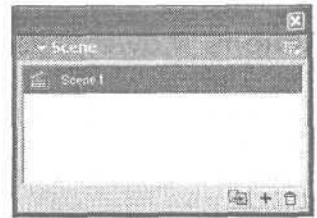
Панель Info, загружаемая вследствие вызова команды Info из меню Window или нажатия клавиш **Cmd/Ctrl+I**, позволяет задать размеры (ширину и высоту) выбранного объекта (поля Width и Height) и его положение (поля X и Y). В нижней части панели Info выводится информация о цвете выбранного объекта



(в формате модели RGB). Наконец, в нижнем правом углу рассматриваемой панели демонстрируются текущие координаты указателя мыши.

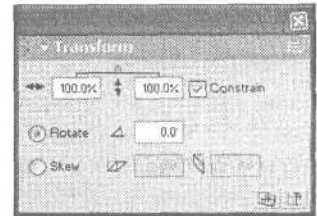
Панель Scene

Панель Scene можно открыть с помощью команды Scene из меню Window. Элементы данной панели предоставляют возможность переименовывать, добавлять и удалять сцены, а также переходить от одной сцены к другой. Мы вернемся к изучению панели Scene в главе 11.



Панель Transform

Подобно панели Info, панель Transform позволяет манипулировать выбранными объектами путем указания числовых значений. Данная панель открывается с помощью команды Transform из меню Window, клавиатурный аналог которой — **Cmd/Ctrl+T**. В верхней части панели находятся два поля, в которых можно задавать размер объекта по вертикали и по горизонтали.

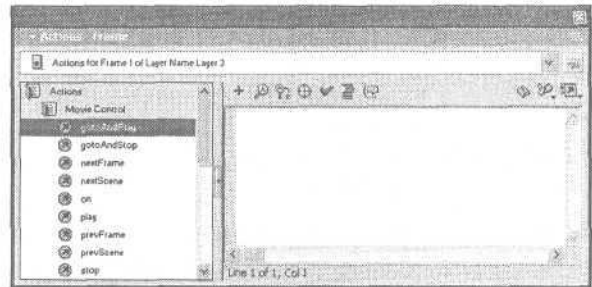


При изменении масштаба выбранного объекта с помощью панели Transform вы сможете сохранить пропорции объекта, если установите переключатель Constrain.

Поля, расположенные в нижней части панели Transform, предназначены для определения угла поворота и степени перекашивания выделенного объекта.

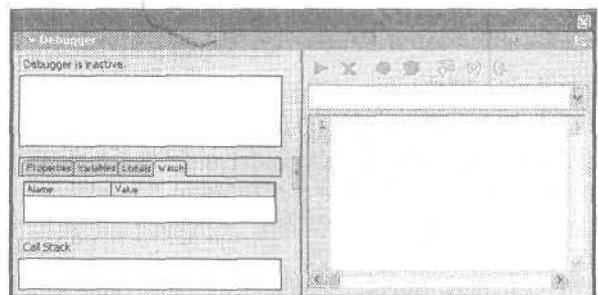
Панель Actions

Панель Actions открывается с помощью команды Actions из меню Window либо клавиши **F9**. Она позволяет ассоциировать команды **ActionScript** с объектами или кадрами. Вы можете либо выбрать команду в списке, либо создать собственную команду, работая в режиме Normal (Обычный) или Expert (Эксперт). Подробнее о панели Actions — в главе 12.



Панель Debugger

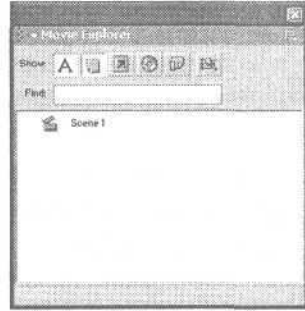
Поскольку ActionScript — это язык написания сценариев, вполне закономерно, что компания Macromedia включила отладчик ActionScript в состав программы Flash. Отладчик ActionScript, который впервые появился в Flash 5, позволяет



обнаруживать и исправлять неисправности в сценариях ActionScript. Открыть окно отладчика позволяет команда Debugger из меню Window. Знакомиться с возможностями отладчика ActionScript мы продолжим в главе 18.

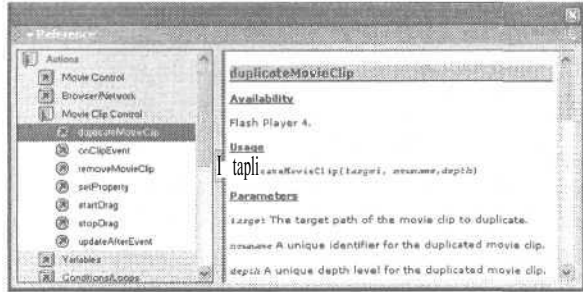
Панель Movie Explorer

Панель Movie Explorer — это полезное средство, впервые представленное в Flash 5, с помощью которого обеспечивается доступ к элементам настройки всех параметров фильма. Открыть панель можно, выбрав команду Movie Explorer в меню Window или нажав клавиши Option/Alt+F3. Данная панель позволяет производить поиск объектов и элементов по имени, выводить и изменять свойства заданного элемента, а также заменять шрифты. Детально панель Movie Explorer рассматривается в главе 7.



Панель Reference

MX С выходом каждой новой версии Flash язык сценариев ActionScript становится все более сложным. Ввиду этого компания Macromedia предложила в помощь разработчику панель Reference — интегрированный архив, в котором для каждого элемента ActionScript собраны документы и записи (описания параметров и характеристик совместимости версий Flash, примеры). Данная панель загружается с помощью команды Reference из меню Window.

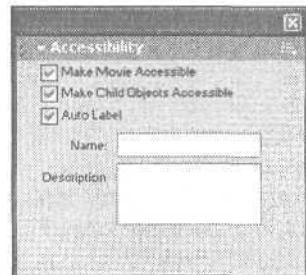


Окно Output

Окно Output, которое не может быть состыковано с другими панелями, является весьма полезным инструментом. После экспорта фильма в данном окне выводится информация о размере файла для всех сцен, объектов, текстов, символов и копий. Открыть окно Output можно с помощью команды Output из меню Window.

Панель Accessibility

MX Появление в Flash MX панели Accessibility следует рассматривать как значительный шаг вперед в процессе создания продукта, доступного для аудитории с различными физическими недостатками. Для загрузки панели Accessibility предназначена одноименная команда из меню Window. В принципах использования данной панели мы познакомимся в главе 27.



Панель Components

MX Панель Components, загрузка которой также осуществляется из меню Window, окажется полезной при создании сложных клипов с заранее определенными параметрами. Данная панель обеспечивает доступ к компонентам (окно с прокруткой или раскрывающееся меню), которые применяются при разработке сложного пользовательского интерфейса.

Панель Component Parameters

MX Панель Component Parameters служит для настройки параметров тех компонентов, которые были добавлены с использованием панели Components. Данную панель можно открыть, активизировав команду Component Parameters в меню Window.

Панель Library

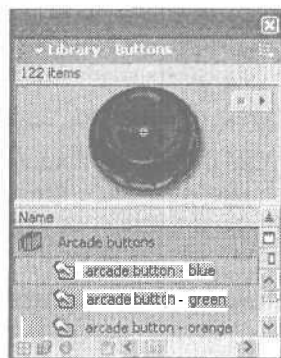
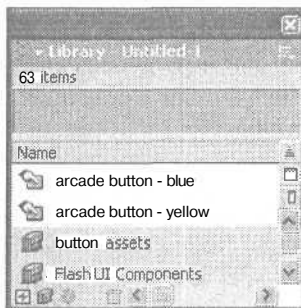
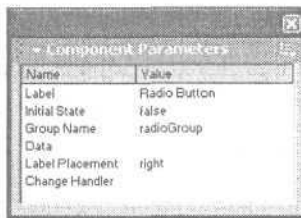
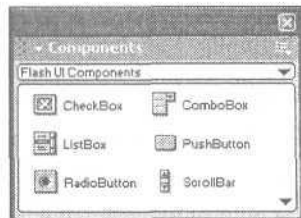
На панели Library (загружается при вызове одноименной команды меню Window или нажатии клавиши F11) предоставляется доступ ко всем символам, которые вы создаете или используете в своем Flash-фильме. Это может быть Movie Clip (Клип), Button (Кнопка) или Graphic (Графический символ). Подробнее об использовании этой панели мы расскажем в главе 7.

Подменю Common Libraries

Разделы библиотеки содержит не только те символы, которые ассоциированы с открытым в настоящий момент фильмом, но и общие наборы символов (Buttons (Кнопки), Learning Interactions (Элементы интерактивного обучения) и Sounds (Звуки)). Общее количество поставляемых с Flash уже разработанных символов достаточно велико, так что вам нет необходимости создавать их самостоятельно.

Панель Sitespring

MX При выборе команды Window ► Sitespring пользователь попадает на Web-узел Sitespring. Этот узел разработан компанией Macromedia как средство автоматизации коллективной работы, позволяющее участникам группы следить за процессом разработки, общаться друг с другом, осуществлять управление версиями и т. д.



Настройка панелей

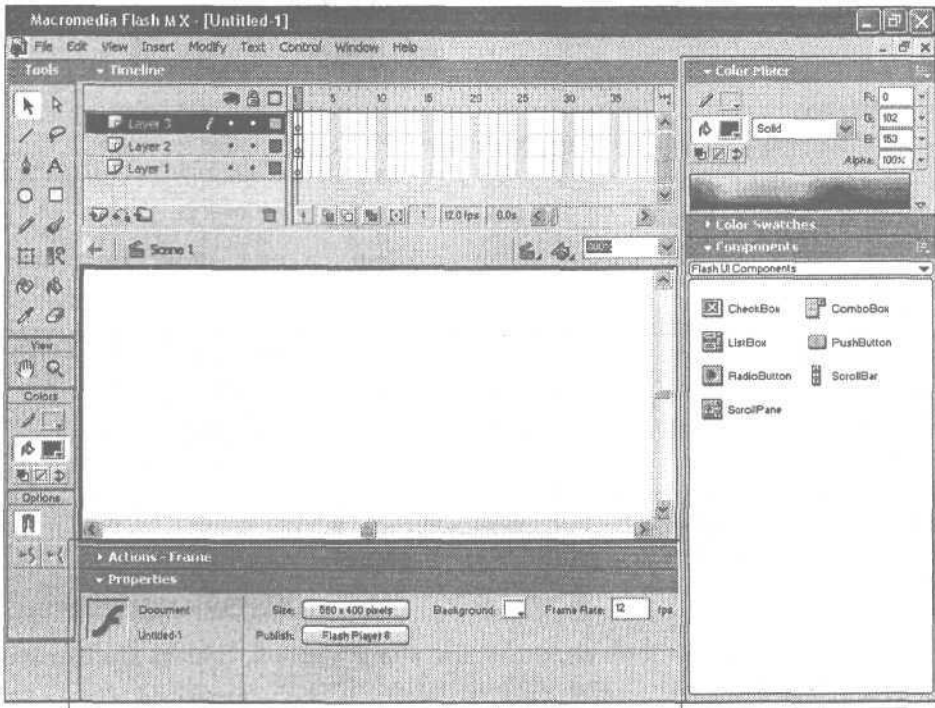
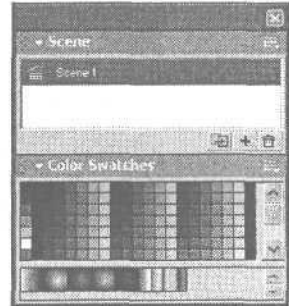
Одной из сильных сторон нового пользовательского интерфейса Flash MX является возможность цельного размещения всех панелей, что делает среду разработки особенно удобной.

Как уже отмечалось, панели в новой версии Flash MX можно располагать автономно или вдоль сторон других панелей, сворачивать, разворачивать и, наконец, упорядочивать таким образом, который наиболее удобен при решении конкретной задачи. Далее мы поговорим о том, как производить все эти операции.

Стыковка панелей

При стыковке панель комбинируется с другим элементом интерфейса. Если состыковать две плавающие панели, получится плавающая мегапанель. Как показано на рис. 3.7, вы можете состыковать панель с частью базового интерфейса.

Стыковка панелей значительно облегчает доступ к инструментам и информации при максимально эффективном использовании рабочего пространства.

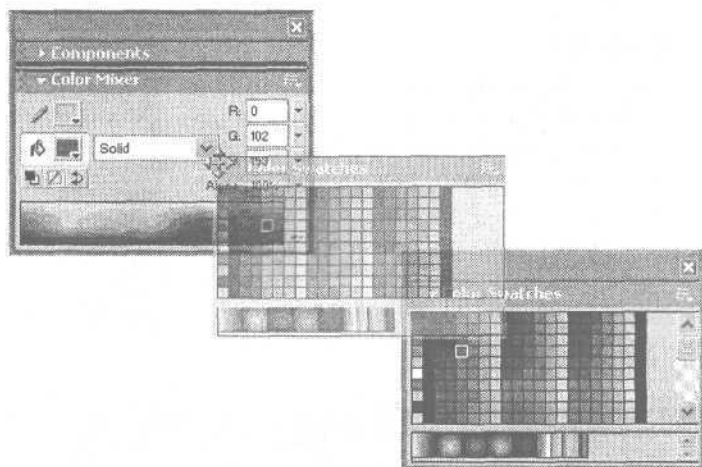


Состыкованные панели

Рис. 3.7. Состыкованные панели

Чтобы состыковать две панели, выполните следующие действия:

1. Откройте две панели, используя меню Window.
2. Установите указатель мыши в области захвата заголовка панели (точки в левой части строки заголовка). Нажмите левую кнопку манипулятора и, удерживая ее, переместите панель в направлении к другой панели. В процессе перетаскивания отображается контур, по которому легко определить текущую позицию панели.



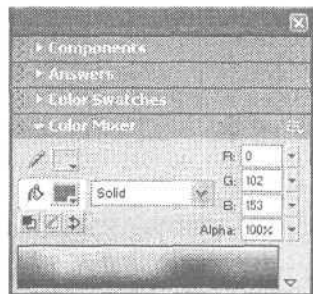
ПРИМЕЧАНИЕ

В области захвата заголовка панели изображение указателя мыши меняется.

3. Расположите указатель мыши над второй панелью.
4. Когда вокруг второй панели появится темный ореол, отпустите кнопку мыши.

Далее рассказывается, как состыковать панель с базовым интерфейсом:

1. Откройте панель, которую вы хотите состыковать с базовым интерфейсом, используя команду из меню Window.
2. Переведите указатель мыши в область захвата панели. Нажмите левую кнопку манипулятора и, удерживая ее, переместите панель в направлении к той области интерфейса, с которой вы хотите состыковать ее.
3. Когда появится темный ореол, отпустите кнопку мыши. Панель автоматически будет состыкована с указанной областью интерфейса.



В Flash MX разрешается стыковка панелей с левой, нижней и правой границами окна программы.

Свертывание и разворачивание панелей

Возможность свертывать и разворачивать панели обеспечивает быстрый доступ ко многим необходимым инструментам и способствует эффективному использованию рабочего пространства (рис. 3.8).

Чтобы развернуть свернутую панель (или свернуть развернутую), достаточно щелкнуть на имени панели в строке заголовка.

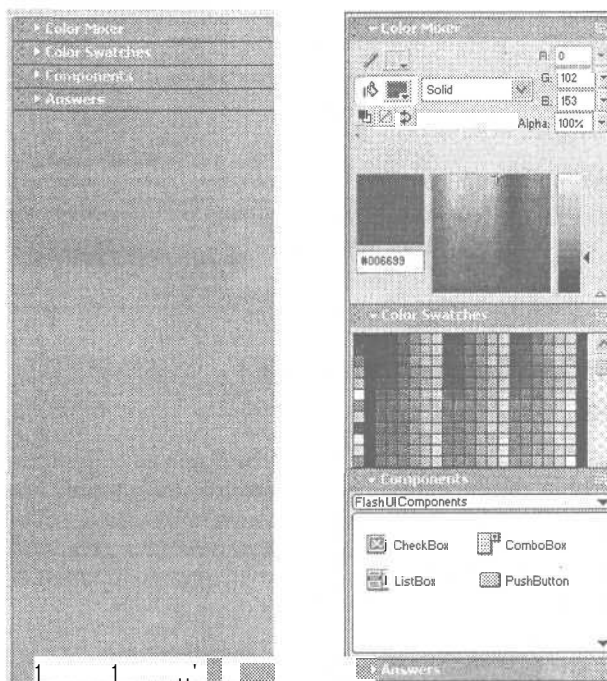


Рис. 3.8. Слева — группа свернутых панелей, справа — те же панели в развернутом виде

Выбор схемы расположения элементов интерфейса

Поскольку интерфейс программы Flash MX гибок в настройке, существует множество вариантов расположения элементов интерфейса. Вы можете создать собственную схему, отвечающую специфике вашего рабочего процесса, или выбрать одну из схем, предлагаемых компанией Macromedia.

Чтобы выбрать стандартную схему, откройте подменю Window ▶ Panel Sets и выберите один из семи вариантов.

Для каждой схемы характерен специфический набор панелей, зависящий от разрешения экрана: чем выше разрешение, тем больше панелей.

Создание пользовательской схемы расположения элементов интерфейса

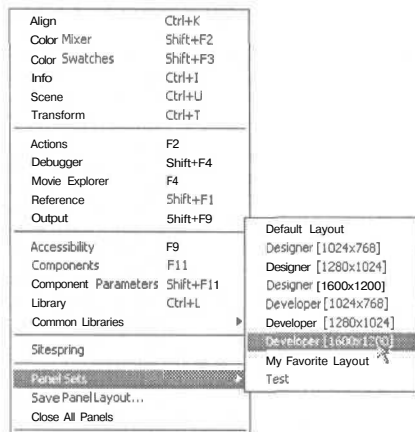
По прошествии некоторого времени, необходимого для освоения элементов интерфейса программы Flash MX, вы сами сможете создавать оптимальные схемы,

каждая из которых будет предназначена для решения определенной задачи. Чтобы каждый раз не перестраивать интерфейс, введена возможность сохранения схем для дальнейшей загрузки. Сохраняемые пользовательские схемы помещаются в то же меню, что и стандартные.

Рассмотрим пошагово процесс сохранения пользовательской схемы расположения панелей:

1. Выберите команду Save Panel Layout в меню Window.
2. Когда появится диалоговое окно Save Panel Layout, введите имя вашей схемы в поле Name.
3. Щелкните мышью на кнопке ОК.

Загрузить данную схему вы сможете посредством команды Window ▶ Panel Sets ▶ название вашей схемы.



Выбор комбинации клавиш для вызова команды

Комбинации клавиш — это средство быстрого доступа к функциям программы. Как и многие другие приложения, выпускаемые компанией Macromedia, Flash предлагает специальный редактор, позволяющий создавать собственные комбинации клавиш. Чтобы загрузить редактор Keyboard Shortcuts, используйте команду Keyboard Shortcuts меню Edit (рис. 3.9).



Рис. 3.9. Редактор Keyboard Shortcuts

Для того чтобы побольше узнать о возможностях редактора Keyboard Shortcuts, выполните упражнение из практикума 1.

Использование справочной системы Flash

Справочная система Flash содержит огромное число документов и статей, призванных помочь разработчику при возникновении затруднительных ситуаций. Меню Help включает следующие команды:

- ▶ Using Flash (Help ▶ Using Flash) — обеспечивает доступ к справочному руководству, информацию в котором можно искать, используя содержание (Content List), алфавитный указатель (Index) или поисковую систему (Search).
- ▶ ActionScript Dictionary (Help ▶ ActionScript Dictionary) — открывает окно справки по синтаксису языка ActionScript.
- ▶ Lessons (Help ▶ Lessons) — предоставляет доступ к обучающей системе, поставляемой в комплекте с программой и охватывающей темы от введения во Flash до использования Flash с программами Freehand и Illustrator.
- ▶ Samples (Help ▶ Samples) — демонстрирует интересные технические приемы, которые вы сможете использовать в своих разработках.
- ▶ Flash Support Center (Help ▶ Flash Support Center) — дает возможность получить через Интернет самую свежую информацию о Flash и техническую поддержку. (Иногда данный Web-узел называют Developers Resource Center.)
- ▶ Flash Tutorials (Help ▶ Tutorials) — предоставляет доступ к серии учебных пособий, исходные файлы которых были установлены на вашем жестком диске (в каталоге Tutorials) при инсталляции программы Flash MX.

Резюме

В данной главе состоялось знакомство с интерфейсом программы Flash MX. Мы поговорили о том, для чего предназначены и как используются главное меню, панели инструментов, область действия, рабочая область и временная шкала. Особый интерес представляет материал о настройке интерфейса Flash с целью увеличения эффективности работы. Завершил меню обзор разделов справочной системы.

Знание интерфейса Flash необходимо для успешного изучения возможностей программы. Помните: чем основательнее вы изучите среду разработки, тем быстрее достигнете успешных результатов в создании Flash-фильмов.

Практикум 1

Пользовательские комбинации клавиш

Некоторые стандартные комбинации клавиш могут показаться неудобными. Разработчики Flash предусмотрели такую ситуацию и создали редактор Keyboard Shortcuts, который позволяет самостоятельно определять сочетания клавиш.

Изменение клавиш быстрого вызова команд

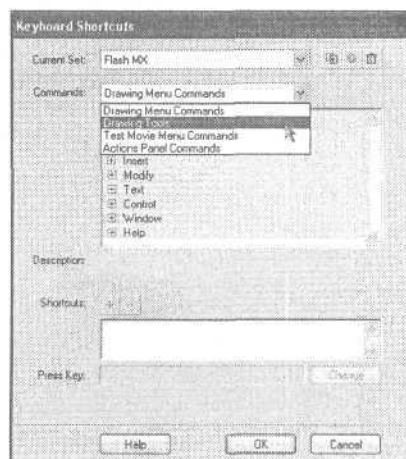
За многими командами закреплены клавиши быстрого вызова. Подбор наиболее удобных для вас комбинаций сделает работу с редактором намного эффективнее. Давайте ознакомимся с процедурой создания комбинации клавиш для вызова команды:

1. Откройте окно редактора Keyboard Shortcuts, вызвав команду Keyboard Shortcuts меню Edit.
2. Выберите пункт Flash MX в выпадающем меню Current Set (Текущий набор), которое расположено в верхней части окна редактора Keyboard Shortcuts.
3. Щелкните на кнопке Duplicate Set (Дублировать набор) справа от меню Current Set.

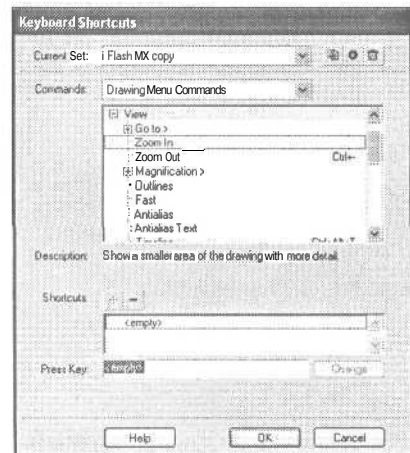
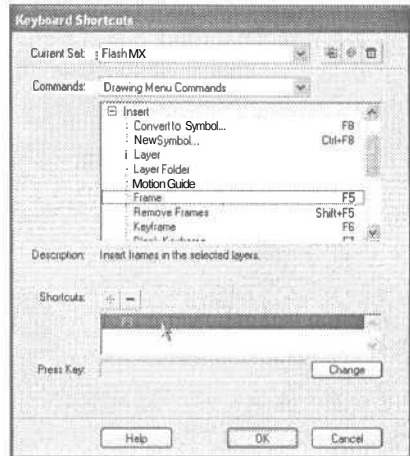
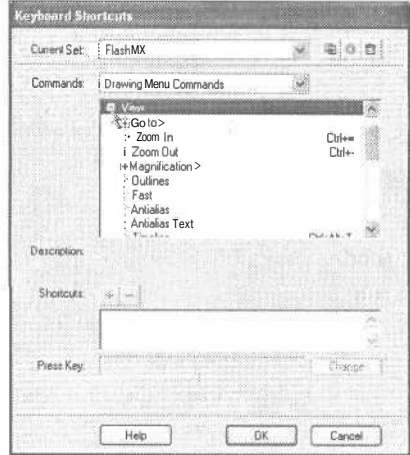
ПРИМЕЧАНИЕ

Поскольку нельзя изменять комбинации клавиш Flash MX, установленные по умолчанию, вы создаете дубликаты, которые и подвергаются редактированию.

4. В диалоговом окне Keyboard Shortcuts введите имя дубликата в поле Duplicate Name и щелкните на кнопке ОК. Имя только что созданного дубликата появится в меню Current Set.
5. Укажите в меню Commands группу команд, с которой хотите работать.



6. При выборе категории команд — Drawing Menu Commands (Команды меню рисования) или Test Movie Menu Commands (Команды меню тестирования фильма) — щелкните па знаке «плюс», чтобы развернуть весь список. Если вы выбрали команды Drawing Tools (Инструменты для рисования) или Actions Panel Commands, перейдите к пункту 9.
7. Укажите команду. Обратите внимание на то, что справа от команд отображаются комбинации клавиш, предусмотренные для них изначально (они также выводятся в иоле Shortcuts, расположенном ниже).
8. Выберите существующую комбинацию клавиш в поле Shortcut.
9. Щелкните па кнопке Remove Shortcut (-) (Удалить комбинацию клавиш).
10. Выполните щелчок па кнопке Add Shortcut (+) (Добавить комбинацию клавиш), вследствие чего поле Press Key будет подсвечено.
11. Нажмите комбинацию клавиш, которую вы хотите добавить. Ее обозначение появится в поле Press Key.
12. Щелкните па кнопке Change (Изменить). Новая комбинация клавиш будет закреплена за выбранной командой.
Если введенная комбинация уже задействована, Flash сообщит вам об этом и предложит либо изменить, либо удалить ее.
13. Произведя все необходимые изменения, подтвердите их нажатием кнопки ОК.



ЧАСТЬ 2

Содержимое фильма: создание и управление

С помощью Flash MX вы создадите множество замечательных интерактивных продуктов (традиционных Web-страниц, киосков и других приложений), работа которых в большинстве случаев связана с воспроизведением видео. В этом нет ничего удивительного, поскольку приложение Flash MX, используя возможности векторной графики, позволяет добиваться интересных и сложных видеоэффектов.

Таким образом, в процессе разработки вам придется уделять особое внимание визуальным аспектам фильмов. Для этого в вашем распоряжении имеются все необходимые средства. Компания Macromedia снабдила свой замечательный продукт обширным набором мощных инструментов, позволяющих создавать визуальные эффекты, а также управлять и манипулировать ими.

Глава 4. Рисование и черчение.....	62
Глава 5. Работа с текстом.....	87
Глава 6. Работа с объектами.....	100
Глава 7. Многократно используемые объекты.....	121
Глава 8. Слои.....	148

Глава 4

Рисование и черчение

В арсенале Flash — множество средств для работы как с векторной, так и с растровой графикой. Знакомство с ними мы начнем в этой главе с обзора инструментов, предназначенных для рисования. Это — Pencil (Карандаш), Brush (Кисть), Pen (Перо), Ink Bottle (Чернильница), Paint Bucket (Заливка) и Eyedropper (Пипетка). Но прежде изучим *инструменты выбора*, ведь именно с выделения объекта начинается большинство процессов. В заключительной части главы мы поработаем с инструментом Eraser (Ластик), а также с панелью Properties, где осуществляется настройка параметров обводки и заливки.

Глава охватывает такие темы:

- ▶ инструменты выбора;
- ▶ рисование линий;
- ▶ рисование основных фигур;
- ▶ использование инструментов Pencil, Brush, Pen, Subselection, Ink Bottle и Paint Bucket;
- ▶ копирование цветов с помощью инструмента Eyedropper;
- ▶ использование инструмента Eraser;
- ▶ изменение обводки объекта с помощью панели Properties;
- ▶ изменение заливки объекта с помощью панели Properties;
- ▶ цифровое представление цвета.

Выбор объектов

Программа Flash отличается разнообразием инструментов выбора, действие которых распространяется на линии, объекты, их фрагменты, группы объектов, символы, кнопки и многие другие элементы. Перейдем непосредственно к обзору инструментов, в частности Arrow, Subselection и Lasso.

ПРИМЕЧАНИЕ

Поскольку инструменты Pen и Subselection взаимосвязаны, трудно провести границу между ними при рассмотрении возможностей и характеристик. Подробно о функциональных возможностях инструмента Subselection (особенно в сочетании с инструментом Pen) рассказывается в разделе «Инструмент Pen».

Инструмент Arrow

Инструмент Arrow (Стрелка), расположенный в верхнем левом углу панели Tools, позволяет выбирать и перемещать один или несколько элементов в пределах рабочей области. В его функции входит также изменение формы невыбранной линии, обводки или объекта.

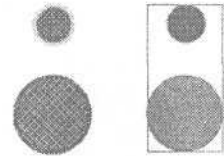
Выбор объекта посредством инструмента Arrow — это первый шаг при редактировании свойств объекта с помощью панели Properties.

Выбор объекта

Выбор объекта с помощью инструмента Arrow — простая процедура: достаточно установить указатель мыши на объекте и выполнить щелчок. Однако этот инструмент не так прост, как может показаться на первый взгляд. В работе с ним есть множество нюансов, с которыми мы и хотим вас познакомить.

Если выделенный объект не входит в состав группы, над ним появляется сетчатая заливка — признак выделения. Вопросы группировки объектов освещаются в главе 6.

При выделении объекта, входящего в группу, появляется тонкая рамка. На рисунке демонстрируются различия при выделении инструментом Arrow двух независимых объектов (слева) и сгруппированных объектов (справа).



СОВЕТ

Многие объекты, особенно те, которые созданы с помощью одного из инструментов рисования стандартных фигур, имеют как обводку, так и заливку. Для выделения такого объекта необходимо выбрать и заливку, и обводку. Это можно сделать, выполнив двойной щелчок на заливке. Другой путь — выбор обводки и заливки с последующей активизацией команды Group меню Modify. Для выделения группы достаточно щелкнуть на одном из составляющих ее объектов.

Выделение объекта, выбранного посредством инструмента Arrow, снимается в результате выполнения щелчка в любом другом месте рабочей области. К тому же результату приводит активизация команды Deselect All меню Edit, нажатие клавиши Esc либо комбинации клавиш Cmd+Shift+A (Macintosh) или Ctrl+Shift+A (Windows).

Перемещение объекта

Чтобы переместить объект с помощью инструмента Arrow, установите указатель в его пределах, затем нажмите левую кнопку мыши и, не отпуская ее, перетащите объект в нужное место, после чего отпустите кнопку манипулятора. Если объект входит в состав группы, перемещена будет группа целиком.

ВНИМАНИЕ

Как вы уже знаете, многие объекты имеют и обводку, и заливку. Будьте внимательны! При выборе только одного компонента после перемещения другой компонент останется на прежнем месте.

Изменение формы объекта

Вероятно, изменять форму объекта удобнее с помощью других инструментов Flash, однако инструмент **Arrow** также предоставляет такую возможность, которую мы и рассмотрим далее.

1. Активизируйте инструмент **Arrow** на панели **Tools**. Убедитесь, что объект (в рассматриваемом примере — овал) не включен в группу и не выбран.
2. Установите инструмент **Arrow** на таком расстоянии от края овала, чтобы вид указателя мыши изменился.
3. Нажмите и удерживайте левую кнопку мыши. Перемещая указатель, измените форму овала. Режим предварительного просмотра позволит вам следить за трансформацией объекта.
4. Если требуемая форма получена, отпустите кнопку мыши.



Привязка к объектам

При нажатии кнопки **Snap to Objects** в момент, когда выбран инструмент **Arrow**, осуществляется привязка выделенных объектов к другим объектам, находящимся в области действия. Данная кнопка расположена в секции **Options** панели инструментов. Выделенные объекты можно привязывать к чему угодно: к линиям, контурам, обводкам и т. д. Существует даже возможность привязки объектов к сетке области действия. В режиме привязки удобно выравнивать объекты в области действия и располагать их в определенном порядке.

Чтобы сделать сетку видимой, активизируйте команду **View ▸ Grid ▸ Show Grid**. Привязать объекты к сетке позволяет команда **View ▸ Grid ▸ Snap to Grid**. Возможность редактировать свойства сетки предоставляет команда **View ▸ Grid > Edit Grid**.

В режиме **Snap to Objects** при перемещении объекта с помощью инструмента **Arrow** в его центре появляется небольшой кружочек — *точка регистрации*.

Мы осуществляем привязку именно точки регистрации, а не самого объекта. Поскольку точка регистрации и объект являются единым целым, данное утверждение достаточно спорно. При перемещении точки регистрации (а следовательно, и самого объекта) на другой объект (например,



на прямую линию, созданную с помощью инструмента Line) кружочек становится жирным и черным, что является признаком привязки к данному объекту.

В данном разделе мы не ставили целью дать вам полную информацию о применении точек регистрации. Рассмотрение этой темы будет продолжено в главе 6.

ПРИМЕЧАНИЕ

Режим Snap to Objects особенно удобен при построении промежуточных изображений в процессе превращения исходного объекта в заданный. Подробная информация о создании промежуточных изображений содержится в главе 9.

Привязка к сетке пикселей

Привязка к сетке пикселей осуществляется посредством отдельной команды Snap to Pixels меню View, поэтому соответствующая кнопка не появляется в секции Options панели инструментов Tools, когда выбран инструмент Arrow. Однако рассмотрение этой операции уместно именно на данном этапе.

В результате активизации команды Snap to Pixels все объекты привязываются к узлам сетки пикселей, которая становится видимой только при увеличении масштаба отображения как минимум до 400 процентов. Сетку пикселей на время можно сделать невидимой, нажав клавишу X на клавиатуре.

Сглаживание

Сглаживание — уменьшение числа изгибов в выбранной кривой — производится посредством кнопки Smooth, которая появляется в секции Option при выборе инструмента Arrow.

Чтобы выполнить сглаживание, выделите кривую с помощью инструмента Arrow, а затем щелкните на кнопке Smooth.

Еще один способ — активизация команды Smooth из меню Modify после выбора кривой.

Выпрямление

Помимо сглаживания линию можно подвергнуть выпрямлению, которое осуществляется посредством кнопки Straighten, появляющейся в секции Option при выборе инструмента Arrow.

Альтернативное средство, команду Straighten, вы найдете в меню Modify.

Выбор с помощью инструмента Lasso

В то время как инструмент Arrow используется для выбора отдельных объектов, инструмент Lasso позволяет выделить все объекты (или их фрагменты) в указанной области. Инструмент Lasso прост в применении. Все, что вам необходимо сделать, — активизировать этот инструмент и, удерживая нажатой левую кнопку мыши, нарисовать линию вокруг той области, которую необходимо выделить.

Инструмент Lasso позволяет выделять фрагменты объектов (например, угол квадрата), которые впоследствии можно перемещать с помощью инструмента Arrow. В рассматриваемом случае эффект будет таким, как будто угол квадрата отрезан ножницами и сдвинут в сторону.

В секции Option для инструмента Lasso выводятся три дополнительных инструмента: Magic Wand (Волшебная палочка), Magic Wand Properties (Свойства волшебной палочки) и Polygon mode (Режим многоугольника).

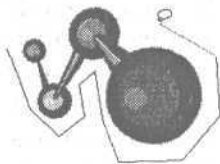
ВНИМАНИЕ -

Инструмент Lasso автоматически замыкает контур выделения, если вы не сделали этого.

Режим многоугольника

В режиме Polygon mode выделение области осуществляется с более высокой точностью, чем при обычном использовании инструмента Lasso.

Процедура выделения в данном случае заключается в рисовании множества прямых линий вокруг области. Каждый раз при нажатии кнопки мыши создается опорная точка. Затем вы рисуете очередную прямую линию, которая соединяется с только что созданной опорной точкой. Процесс продолжается до тех пор, пока вы не замкнете контур, выполнив двойной щелчок мышью.



ПРИМЕЧАНИЕ —

При использовании инструмента Lasso, как с инструментом Polygon mode, так и без него, концы незамкнутого контура автоматически соединяются по окончании выбора.

Волшебная палочка

Параметр Magic Wand инструмента Lasso позволяет выделить область, окрашенную в определенный цвет, после разбивки растрового изображения.

При разбивке растрового изображения па участки с помощью команды Break Apart вы указываете Flash, что изображение необходимо рассматривать как набор отдельных участков цвета. Подробно работа с растровыми изображениями в среде Flash MX рассматривается в главе 6.

После того, как растровое изображение будет разбито на участки (что осуществляется путем выделения изображения и активизации команды Break Apart из меню Modify), вы можете выбрать области близких цветовых оттенков, щелкнув на них инструментом Lasso с активным параметром Magic Wand.

Признаком выделения области того или иного цвета является нанесенная поверх сетчатая заливка.

Свойства волшебной палочки

Кнопка Magic Wand Properties, которая находится в секции Option, если выбран инструмент Lasso, позволяет изменять свойства волшебной палочки. В результате щелчка на этой кнопке открывается диалоговое окно Magic Wand Settings, где и производится настройка. Рассмотрим, какими свойствами волшебной палочки мы можем управлять:

- ▶ **Threshold** (пороговая величина). В этом поле можно изменить цветовой порог — диапазон оттенков, охватываемых при выделении. Диапазон допустимых значений — от 0 до 200. Если значение Threshold равно 0, в выделяемую

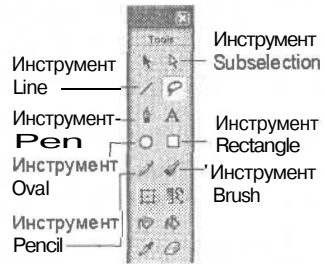
область будут включаться пиксели, которые по цветовым характеристикам соответствуют первому выбранному пикселу. При увеличении значения Threshold количество цветов, включаемых в выделяемую область, возрастет.

- Smoothing (сглаживание). Значение в данном поле определяет, насколько сглаженными будут края выбираемого участка. Установить можно следующие значения: Normal (Обычное), Pixels (По пикселям), Rough (Грубое) и Smooth (Плавное).

Рисование графических элементов

В приложении Flash имеется богатый выбор инструментов, с помощью которых вы сможете нарисовать все, на что хватит фантазии, будь то любимый цветок или автопортрет в манере Пикассо.

Все инструменты рисования представлены на панели Tools. В процессе работы с ними вы сможете изменять характеристики обводки и заливки с помощью панели Properties до создания объектов.



Инструмент Line

Инструмент Line (Линия) предназначен для черчения прямых линий. Рассмотрим, как выполняется данная операция: активизируйте инструмент Line, установите указатель в начальной точке линии и, удерживая нажатой левую кнопку мыши, переместите его к конечной точке, после чего отпустите кнопку манипулятора. Толщина и стиль создаваемых линий определяются на панели Properties. За подробной информацией о манипуляциях с обводкой объектов обращайтесь к разделу «Изменение обводки с помощью панели Properties».

ВНИМАНИЕ

Если во время использования инструмента Line нажать и удерживать клавишу Shift, линия будет проведена под углом 45 градусов.

Рисование геометрических фигур

Для рисования геометрических фигур Flash предлагает два инструмента: Oval (Овал) и Rectangle (Прямоугольник). Эти инструменты во многом похожи, но в тоже время каждый из них имеет специфические характеристики.

Эллипсы

Эллипсы и окружности создаются посредством инструмента Oval. Он достаточно прост в применении. Выполните щелчок в той точке, с которой должно начинаться рисование фигуры, и, удерживая нажатой левую кнопку мыши, перемещайте указатель, пока не получите фигуру требуемых формы и размера. После этого отпустите кнопку мыши. Созданный овал будет иметь как обводку, так и заливку. О том, как настраивать свойства обводки и заливки овала, рассказывается

в разделах «Изменение обводки с помощью панели Properties» и «Изменение заливки с помощью панели Properties».

СОВЕТ

Если во время использования инструмента Oval удерживать нажатой клавишу Shift, программа начертит правильную окружность.

Прямоугольники

Прямоугольники и квадраты создаются посредством инструмента Rectangle. Рисование этих объектов осуществляется по тем принципам, которые описывались в предыдущем разделе. Готовый прямоугольник имеет как обводку, так и заливку. О том, как настраивать свойства обводки и заливки прямоугольника, рассказывается в разделах «Изменение обводки с помощью панели Properties» и «Изменение заливки с помощью панели Properties».

СОВЕТ

Чтобы получить квадрат, в процессе применения инструмента Rectangle следует удерживать нажатой клавишу Shift.

Закругление углов прямоугольника

В секции Option для инструмента Rectangle дополнительно выводится кнопка Rounded Rectangle Radius (Радиус скругленных углов прямоугольника). Если заполнить щелчок на ней, откроется диалоговое окно Rectangle Setting (Параметры прямоугольника). В поле Corner Radius этого окна определяется величина радиуса закругления углов прямоугольника (от 0 до 100). Чем выше значение, тем больше радиус. Введите несколько значений и проследите, как изменяется изображение.

Инструмент Pencil

Инструмент Pencil предназначен для рисования произвольных линий (что отличает его от инструмента Line). Работать с ним легко. Активизируйте инструмент, выполните щелчок в начальной точке и, удерживая нажатой левую кнопку мыши, начинайте рисовать контур, как карандашом на листе бумаги. По завершении отпустите кнопку манипулятора.



Выбор режима работы

Режим работы инструмента Pencil определяется в меню Pencil mode (Режим карандаша). Кнопка, предоставляющая доступ к нему, расположена в секции Options панели Tools. Указанное меню содержит три команды: Straighten (Выпрямление), Smooth (Сглаживание) и Ink (Чернила). Заданный режим действует при рисовании новых линий, однако вы можете представить в нем и существующие линии. С этой целью выделите линию посредством инструмента Arrow и щелкните на кнопке Straighten или Smooth либо активизируйте соответствующую команду меню Modify.

Рассмотрим особенности каждого режима работы инструмента Pencil.

- ▶ **Straighten.** Выпрямление закруглений на линии, вследствие чего она становится угловатой.
- ▶ **Smooth.** Закругление всех углов линии. К сожалению, в данном режиме нельзя управлять степенью закругления.
- ▶ **Ink.** В этом режиме линия выводится в том виде, в котором она нарисована, без сглаживания и выпрямления.

Инструмент Brush

Инструмент Brush (Кисть), как и инструмент Pencil, предназначен для рисования отдельных линий произвольной формы. Его особенность состоит в том, что результирующие линии имеют заливку. Таким образом, чтобы изменить цвет линий, созданных с помощью кисти, следует использовать панель Properties.

ПРИМЕЧАНИЕ

Режимы Straighten и Smooth доступны и для инструмента Brush. Чтобы представить в одном из них линию, нарисованную кистью, следует выделить ее и нажать кнопку Straighten или Smooth (либо активизировать соответствующую команду меню Modify).

Специфика инструмента Brush заключается еще и в возможности использования графических планшетов, которые позволяют изменять плотность линии в процессе рисования. Нажав кнопку Use Pressure (Давление кисти) в секции Options, вы сможете изменить давление кисти на стило.

Размер кисти

Возможности инструмента Brush мы бы признали ограниченными, если бы размер кисти нельзя было изменить. К счастью, компания Macromedia позаботилась о том, чтобы в распоряжении разработчика оказались кисти самых разных размеров. Выбор размера кисти осуществляется в списке Brush Size, находящемся в секции Option панели Tools.

Размер кисти необходимо устанавливать до рисования линии. В процессе черчения нельзя изменить толщину линии, выбрав кисть другого размера.

Форма кисти

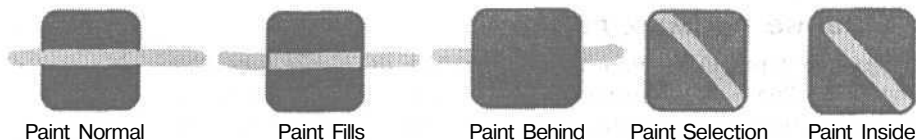
Форму кисти также можно изменить. Возможные варианты представлены в списке Brush shape, расположенном в секции Option. Задавать форму кисти надлежит до рисования линии. К сожалению, в среде Flash по-прежнему не предусмотрены средства для создания пользовательских кистей.

Режим работы с кистью

Программа Flash позволяет определить способ взаимодействия кисти с существующими объектами. Для этой цели предназначено меню Brush mode (Режим кисти), которое вы найдете в секции Option.

Всего насчитывается пять режимов работы инструмента Brush:

- ▶ Paint Normal. Кисть рисует в любом месте.
- ▶ Paint Fills. Кисть рисует только в области заливки, не затрагивая контур.
- ▶ Paint Behind. Кисть закрасивает незаполненные зоны рабочей области на текущем слое.
- ▶ Paint Selection. Кисть рисует только внутри выделенных областей.
- ▶ Paint Inside. Кисть рисует только поверх области заливки, где находится начальная точка линии, и не воздействует на контур.



Инструмент Pen

Инструмент Pen (Перо) считается основным средством создания векторных изображений произвольной формы. Безусловно, большинство инструментов рисования программы Flash (в том числе Line и Pencil) также позволяют создавать векторную графику, но перо обладает рядом серьезных преимуществ.

Инструмент Pen часто применяют совместно с инструментом Subselection (дополнительный выбор), о котором рассказывается в разделе «Изменение позиции точки с помощью инструмента Subselection».

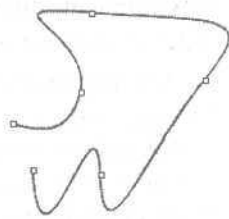
Те из вас, кто знаком с такими программами, как Macromedia FreeHand и Adobe Illustrator, знают свойства инструмента Pen. Он служит для построения сложных контуров путем расстановки опорных точек (узлов), которые автоматически соединяются сегментами.

Опорная точка, без излома соединяющая две кривые, называется гладкой опорной точкой.

Опорная точка, соединяющая две кривые «на изгиб», называется угловой опорной точкой.

Выделенные гладкие опорные точки выглядят как полые кружочки, а выделенные угловые опорные точки — как полые квадраты.

Опорные точки можно свободно перемещать. В зависимости от их позиций меняется и путь кривой. Это позволяет изменять любые сегменты кривой и получать контуры самых разнообразных форм.



Настройка инструмента Pen

Доступ к параметрам инструмента Pen предоставляет команда Preferences меню Edit. Вследствие ее вызова загружается диалоговое окно, в котором вкладка

Editing (Редактирование) содержит три флажка для определения режимов работы пера. Рассмотрим, к каким результатам приводит их установка:

- ▶ Show Pen Preview. Активизируется режим предварительного просмотра сегментов линии во время рисования (перемещения указателя мыши по рабочей области).
- ▶ Show Solid Points. Невыбранные опорные точки отображаются с заливкой, а выбранные — без.
- ▶ Show Precise Cursors. В процессе работы с пером указатель приобретает вид перекрестья.

Рисование прямых линий

С помощью пера легко рисовать прямые линии. Активизируйте инструмент Реп и щелкните в начальной точке линии. В результате будет создана первая опорная точка. Выполните еще один щелчок, на этот раз — в позиции конечной точки. Программа автоматически соединит обе точки прямой линией.



Рисование кривых

Рисование кривых — это область, где в полной мере проявляется мощь инструмента Реп. Начертить прямую линию могут многие инструменты Flash, а вот создавать самые сложные произвольные контуры и с легкостью манипулировать ими способен только этот инструмент.

Рассмотрим пример кривой. Такую кривую можно построить, указав начальную и конечную опорные точки и отрегулировав ее форму с помощью *управляющих* точек. Управляющей называется также линия, проходящая через опорную точку и соединяющая две управляющие точки.



Давайте попрактикуемся в создании кривых:

1. Активизируйте инструмент Реп.
2. Щелкните в том месте рабочей области, где должна находиться начальная точка линии. В указанной позиции появится первая опорная точка.
3. Установите указатель в конечной точке кривой.
4. Нажмите левую кнопку мыши, что приведет к созданию второй опорной точки, и, не отпуская ее, сместите указатель в каком-либо направлении. Заслуживают внимания следующие нюансы:
 - > Во-первых, окажется, что вы перемещаете одну из управляющих точек, связанных со второй опорной точкой сегмента (положение парной управляющей точки автоматически изменяется соответствующим образом).
 - > Во-вторых, чем дальше вы смещаете управляющую точку от опорной точки, тем больше изгиб кривой. Следует также отметить, что управляющая линия растягивается при перемещении указателя мыши.
5. Чтобы придать кривой требуемую форму, перемещайте выбранную управляющую точку.
6. Отпустите кнопку мыши. Управляющая линия исчезнет.

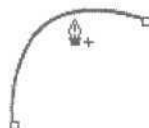


Добавление опорных точек

Каждый раз, когда вы нажимаете кнопку мыши при использовании инструмента Реп, вы добавляете опорную точку к кривой. По завершении процесса черчения вы получаете набор сегментов, которые соединяют опорные точки. Путь кривой определяется позициями опорных точек. Для изменения формы кривой достаточно переместить опорные точки. Чтобы обеспечить полный контроль над формой кривой, необходимо добавить как можно больше опорных точек *после* ее создания.

Процедура добавления опорных точек состоит из таких этапов:

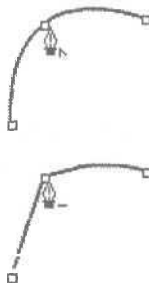
1. Создайте контур. Убедитесь, что выбран инструмент Реп.
2. Установите указатель в том месте контура, где должна быть добавлена точка. Вид указателя изменится: рядом с ним появится знак «плюс».
3. Выполните щелчок, чтобы вставить точку.



Удаление опорных точек

Удалять опорные точки так же просто, как и вставлять. Однако удаление опорных точек выполняется в два этапа. Сначала вы преобразовываете опорную точку в угловую и только после этого удаляете ее.

1. Удостоверьтесь, что контур с многочисленными опорными точками действительно существует в рабочей области. Активизируйте инструмент Реп.
2. Расположите указатель над точкой, подлежащей удалению. Вид указателя изменится: рядом с ним появится знак «V».
3. Выполните щелчок. Опорная точка станет угловой, а вид указателя мыши снова изменится — теперь это перо со знаком «минус».
4. Щелкните еще раз для окончательного удаления точки.



Изменение позиции точки с помощью инструмента Subselection

Как вам уже известно, для изменения пути кривой достаточно переместить опорные точки. Но как это сделать? Такую возможность предоставляет инструмент Subselection (Дополнительный выбор).

1. Создайте линию, состоящую из нескольких сегментов. Активизируйте инструмент Subselection, который расположен справа от инструмента Arrow на панели Tools.
2. Щелкните на кривой, вследствие чего она приобретет светло-голубой оттенок.
3. Установите на точке указатель в виде стрелки и небольшого черного прямоугольника, вследствие чего он несколько изменится: прямоугольник станет полым. Это признак того, что выбрана редактируемая опорная точка.

4. Нажмите левую кнопку мыши и, удерживая ее, переместите точку в требуемую позицию.
5. Отпустите кнопку мыши.

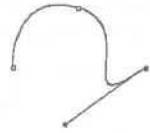
ПРИМЕЧАНИЕ -

Если выбранная вами точка является опорной, то при ее перемещении с помощью инструмента Subselection появляются направляющие.

Редактирование сегмента кривой с помощью инструмента Subselection

Мы неоднократно отмечали, что с помощью пера можно легко и быстро создавать кривые. А что делать, если необходимо отредактировать сегменты кривой, которая уже создана? Для этих целей предназначен инструмент Subselection.

1. Начертите линию, состоящую из нескольких сегментов. Активизируйте инструмент Subselection.
2. Щелкните на сегменте, который требует редактирования. В результате цвет контура станет светло-голубым.
3. Выполните одинарный щелчок на опорной точке. В зависимости от того, является данная опорная точка конечной или промежуточной, появится либо одна, либо две управляющие точки.
4. Щелкните на одной из них и переместите ее таким образом, чтобы кривая приобрела требуемую форму.



Преобразование угловых опорных точек в гладкие

Угловые опорные точки хороши при создании контура с прямолинейными сегментами. В остальных случаях эти точки ограничивают ваши возможности, поскольку сегменты между ними нельзя искривлять. Если в процессе работы возникнет необходимость сделать прямолинейные сегменты изогнутыми, угловые точки придется преобразовать в гладкие. Рассмотрим подробно эту процедуру.

1. С помощью инструмента Subselection выберите угловую точку, которую нужно преобразовать. В результате из полого квадрата она превратится в заполненный квадрат.
2. Нажмите клавишу Option/Alt.
3. Удерживая клавишу Option/Alt, щелкните на точке и немного сместите ее. Рядом моментально появятся управляющие точки.
4. Придайте кривой требуемую форму, перемещая управляющие точки.

Инструмент Ink Bottle: управление обводкой

Инструмент Ink Bottle (Чернила) предназначен для изменения цвета, толщины и стиля обводки. Эти параметры можно задавать также на панели Properties. Важное достоинство инструмента Ink Bottle состоит в том, что он позволяет изменять свойства нескольких объектов одновременно.

Давайте поработаем с инструментом Ink Bottle.

1. Убедитесь, что в области действия нет выделенных объектов. Активизируйте инструмент Ink Bottle.
2. Загрузите панель Properties посредством одноименной команды меню Windows. Вы увидите, что на этой панели отображаются параметры инструмента Ink Bottle.
3. Выберите цвет обводки в поле Stroke color, толщину обводки в поле Stroke height и стиль обводки в раскрывающемся списке Stroke style.

ПРИМЕЧАНИЕ —

Подробную информацию об обводке вы найдете в разделе «Изменение обводки с помощью панели Properties».

4. Расположите указатель мыши (который теперь выглядит как чернильница) в области редактируемой обводки и выполните щелчок. Обводка автоматически изменится в соответствии со значениями параметров, заданными на панели Properties.

Значения параметров инструмента Ink Bottle останутся действительными. Это означает, что вы сможете многократно изменять обводки других объектов в соответствии с установками, выполненными на панели Properties.

Изменение заливки с помощью инструмента Paint Bucket

Изменять заливку объектов, а также закрашивать пустые области позволяет инструмент Paint Bucket (Заливка). Заливка может быть одного из трех видов: сплошная, градиентная или растровая.

Поэкспериментируйте с инструментом Paint Bucket.

1. Убедитесь, что в области действия отсутствуют выделенные объекты. Активизируйте инструмент Paint Bucket.
2. Откройте панель Properties. Вы обнаружите на ней элементы, позволяющие задавать параметры, необходимые для инструмента Paint Bucket.
3. Выберите цвет заливки.
4. Расположите указатель мыши (который теперь выглядит как ведро с краской) внутри пустой или заполненной области, которой назначается новая заливка, и выполните щелчок. Указанная область будет заполнена в соответствии с установками на панели Properties.

Заполнение незамкнутых контуров

Интересным свойством инструмента Paint Bucket является возможность закрашивать незамкнутые области. Для указанного инструмента в секции Option выводится меню Gap Size (Размер зазора), в котором вы определяете, области с какой

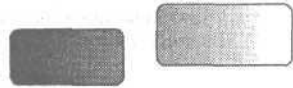
величиной зазора делать доступными для заполнения. Вот перечень возможных вариантов:

- ▶ Close Small Gaps — небольшие зазоры;
- ▶ Close Medium Gaps — средние зазоры;
- ▶ Close Large Gaps — большие зазоры.

Запретить заполнение областей внутри незамкнутых контуров позволяет пункт Don't Close Gaps (Не закрывать зазоры).

Режим Lock Fill

Режим Lock Fill (Заблокировать заливку) инструмента Paint Bucket применяется в тех случаях, когда в качестве заливки используются градиенты. В этом режиме создается иллюзия, что все заполненные области (независимо от того, расположены они рядом или далеко друг от друга) являются частью единого градиента (о том, как создавать градиенты, рассказывается в разделе «Панель Color Mixer»). Кнопка Lock Fill становится доступной в секции Option после активизации инструмента Paint Bucket.



Если вас заинтересовал режим Lock Fill, следуйте нашим инструкциям:

1. Выберите или создайте градиентную заливку, используя панель Color Mixer.
2. Активизируйте инструмент Paint Bucket.
3. Нажмите кнопку Lock Fill.
4. Расположите указатель (в виде ведерка с замочком) во внутренней области объекта с заливкой или без нее. Выполните одинарный щелчок. Область автоматически будет заполнена градиентом, который вы создали на панели Color Mixer.
5. Повторите шаг 4 для заполнения дополнительных областей. Требуемый эффект достигнут!

Инструмент Fill Transform

MX Инструмент Fill Transform (Преобразование заливки), который ранее относился к числу дополнительных функций инструмента Paint Bucket, в Flash MX стал самостоятельным. Предназначение данного инструмента — управление параметрами, влияющими на внешний вид градиентов и растровых заливок.

Рассмотрим приемы работы с инструментом Fill Transform:

1. Активизируйте инструмент Fill Transform.
2. Щелкните на области, имеющей градиентную или растровую заливку. Обратите внимание на то, что при выборе заливки для редактирования появляется рамка выделения с маркерами и центральной точкой. Тип маркеров зависит от того, работаете вы с линейным градиентом, с радиальным градиентом или с растровой заливкой (рис. 4.1).

- > Рамка выделения линейного градиента имеет вид прямоугольника с центральной точкой и круглым маркером в верхнем правом углу.
- > Рамка выделения растровой заливки также имеет прямоугольную форму, но снабжена двумя отдельными наборами маркеров: тремя круглыми (вверху, в правом верхнем углу и справа) и тремя квадратными (слева, в левом нижнем углу и внизу).
- > Рамка выделения радиального градиента имеет овальную форму и снабжена одним квадратным и двумя круглыми маркерами.

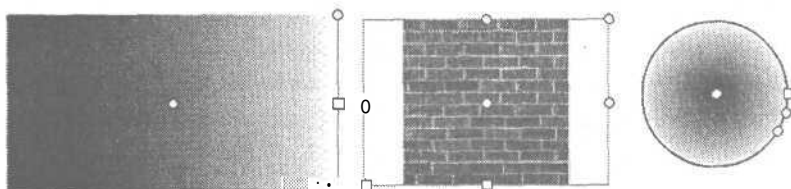


Рис. 4.1. Зависимость рамки выделения от типа заливки

ПРИМЕЧАНИЕ

Когда указатель мыши находится над одним из маркеров, он видоизменяется, сигнализируя о возможности выполнения той или иной операции.

3. На данном этапе вы можете производить следующие действия (рис. 4.2):
 - > *Перемещать центральную точку радиального или линейного градиента.* Установите указатель на центральной точке и, удерживая нажатой левую кнопку мыши, сместите ее в требуемом направлении.

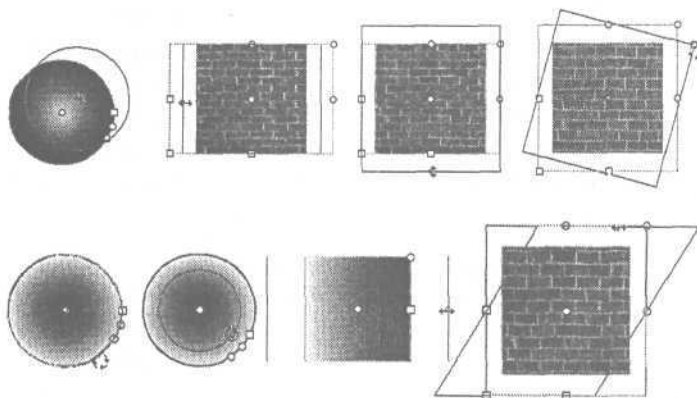


Рис. 4.2. Способы изменения заливки

- ▷ *Изменять ширину линейного градиента или растровой заливки.* Щелкните на квадратном маркере, который расположен на боковой грани рамки, и переместите его. Учтите, что при изменении ширины градиентной или растровой заливки размер объекта остается прежним.

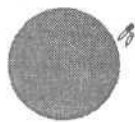
- > *Изменять высоту растровой заливки.* Щелкните на квадратном маркере, который расположен на нижней грани рамки, и переместите его. Если изменится высота градиентной или растровой заливки, размер объекта останется прежним.
- > *Вращать линейный градиент, радиальный градиент или растровую заливку.* Если вы работаете с линейным градиентом или растровой заливкой, переместите с помощью мыши круглый маркер, расположенный в углу рамки выделения. Но если градиент радиальный, перемещать необходимо нижний круглый маркер.
- > *Растягивать и сжимать градиент или растровую заливку.* Перетащите квадратный маркер, расположенный в центре рамки выделения. Если градиент радиальный, перемещать следует верхний круглый маркер.
- > *Перекашивать и наклонять растровую заливку.* Перетащите один из круглых маркеров, расположенных на верхней или на правой грани рамки выделения.

Копирование цветов с помощью инструмента Eyedropper

Инструмент Eyedropper (Пипетка) позволяет копировать цвета заливки и обводки определенного объекта и присваивать их другим объектам.

Копирование цветовых параметров обводки осуществляется следующим образом:

1. Активизируйте инструмент Eyedropper. Удостоверьтесь, что в области действия есть объект, имеющий обводку.
2. Расположите указатель на обводке. В результате его вид изменится: рядом с пипеткой появится небольшой карандаш.
3. Выполните одинарный щелчок. Программа заменит инструмент Eyedropper инструментом Ink Bottle. Это означает, что указанный в области действия цвет можно присвоить обводке другого объекта.
4. Переместите указатель мыши на обводку другого объекта и выполните одинарный щелчок. Свершилось! Обводка второго объекта приобрела такой же цвет, как у первого.



Давайте теперь поработаем с цветом заливки:

1. Активизируйте инструмент Eyedropper и проверьте, содержит ли область действия объект с заливкой.
2. Установите указатель на заливке, вследствие чего его вид изменится: рядом с пипеткой появится кисточка.
3. Выполните одинарный щелчок. Программа заменит инструмент Eyedropper инструментом Paint Bucket. Это означает, что указанный в области действия цвет можно присвоить заливке другого объекта.
4. Переместите указатель мыши на заливку другого объекта и щелкните мышью один раз.



СОВЕТ

Если во время щелчка на объекте инструментом Eyedropper удерживать нажатой клавишу Shift, указанный цвет будет присвоен как обводке, так и заливке других объектов.

Инструмент Eraser

Инструмент Eraser гораздо удобнее в использовании, чем настоящий ластик, — никакой резиновой крошки. Выберите его на панели Tools и, удерживая нажатой левую кнопку мыши, проведите указателем по той области, которую хотите очистить. В процессе перемещения инструмента содержимое области будет удаляться.

Режимы работы инструмента Eraser

Подобно инструменту Brush, инструмент Eraser имеет несколько режимов работы, выбор которых осуществляется в секции Option панели Tools.

Всего насчитывается пять режимов. В каждом из них ластик по-особому влияет на объекты:

- ▶ Erase Normal — удаление всех закрашенных областей и контуров;
- ▶ Erase Fills — удаление заливки с сохранением контуров;
- ▶ Erase Lines — удаление контуров без воздействия на заливку;
- ▶ Erase Inside — удаление только той закрашенной области, в которой начинался процесс, контуры в этом режиме сохраняются;
- ▶ Erase Selected Fills — удаление только выделенных заливок.

Кнопка Faucet

Кнопка Faucet доступна в секции Option панели Tools, если выбран инструмент Eraser. Она позволяет удалять любые заливки или обводки. Для этого необходимо выбрать кнопку Faucet и щелкнуть мышью на обводке или заливке, подлежащей удалению.

Форма ластика

Размер и форму ластика можно изменять. Доступные варианты предлагаются на выбор в выпадающем списке секции Option, когда активен инструмент Eraser. Выполнять данные установки следует *перед* использованием инструмента Eraser.

Изменение обводки с помощью панели Properties

Вы уже знаете, что цвет обводки определяется на панели Color Mixer. Но если наряду с цветом нужно задать и другие параметры обводки, например толщину и стиль, вам потребуется панель Properties.

1. Откройте панель Properties, активизировав команду Properties меню Window.
2. Выберите объект, обводку которого вы хотите отредактировать.

ПРИМЕЧАНИЕ

При обработке несгруппированного объекта убедитесь, что вы не выбрали его заливку.

- Щелкните на образце цвета Stroke Color. После открытия палитры выберите новый цвет для обводки.

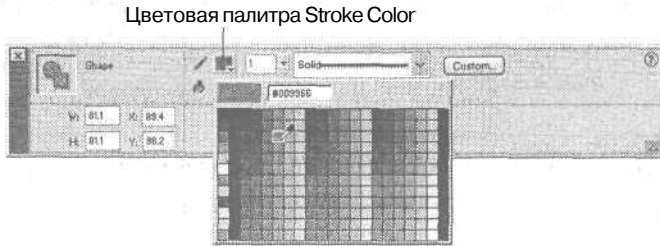


Рис. 4.3. Выбор цвета обводки

- Толщина обводки указывается в поле Stroke Height. Введите числовое значение с клавиатуры либо измените текущую установку путем перемещения ползунка, который появляется в результате выполнения щелчка на небольшой кнопке с направленной вниз стрелкой.

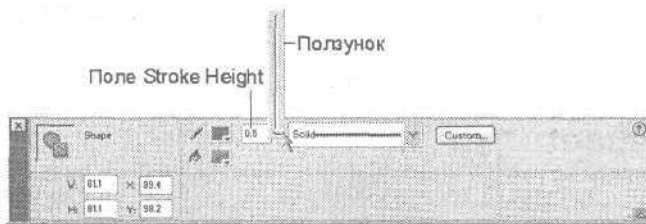


Рис. 4.4. Установка толщины обводки

ПРИМЕЧАНИЕ

Минимальное значение толщины обводки — 0.1, а максимальное — 10.

- Перечень доступных стилей обводки представлен в списке Stroke Style. Если вас не удовлетворяет выбор стилей, которые предлагает Flash, разработайте собственный стиль. Для этого щелкните на кнопке Custom, расположенной справа от списка Stroke Style. В появившемся диалоговом окне подберите подходящие тип и толщину обводки (рис. 4.5).

Производимые изменения будут демонстрироваться в окне предварительного просмотра, которое находится в левой верхней части диалогового окна. Поэкспериментируйте, задавая различные значения для параметров, и проследите за результатами в окне предварительного просмотра. От того, какой тип обводки выбран, зависит набор специфических параметров в диалоговом окне Stroke Style.

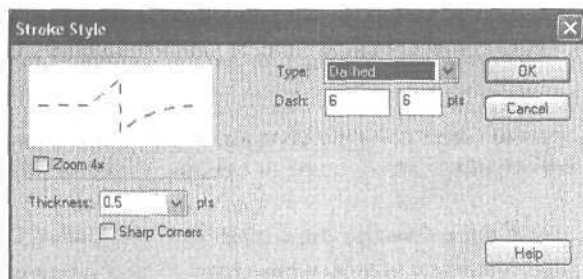


Рис. 4.5. Диалоговое окно выбора стиля обводки

ПРИМЕЧАНИЕ

Цвет обводки можно также изменять с помощью палитры, которая появляется после щелчка на кнопке Stroke Color, расположенной на панели Tools.

Изменение заливки с помощью панели Properties

МХ Существует два способа определения цвета заливки. Один из них мы обсудим позже, в разделе «Панель Color Mixer». А пока научимся задавать цвет заливки с помощью панели Properties. Следует отметить, что возможности панели Properties при изменении цвета заливки довольно ограничены, поскольку доступ предоставляется только к стандартным цветам. Панель Color Mixer интересна тем, что позволяет создавать градиентную и растровую заливки.

ПРИМЕЧАНИЕ

Еще одно средство выбора цвета заливки — палитра Fill Color панели Tools.

Для изменения цвета заливки с помощью панели Properties необходимо выполнить следующие действия:

1. Откройте панель Properties, активизировав команду Properties меню Window.
2. Выберите объект, цвет заливки которого хотите изменить.
3. Щелкните мышью на образце цвета Fill Color (Цвет заливки). После открытия цветовой палитры укажите новый цвет для заливки объекта.



Рис. 4.6. Выбор цвета заливки

Основные положения теории цвета

Управление цветом — важнейший аспект разработки продуктов в среде Flash. За исключением случаев, когда продукт создается для аудитории с неполноценным зрением (что требует большого опыта и специальных знаний), вам необходимо продумать, как максимально эффективно использовать цветовую гамму. Этот раздел познакомит вас с тонкостями процессов выбора, создания и использования цветов.

До появления глобальной сети Интернет графические приложения были ориентированы главным образом на создание изображений для вывода на печать. Для установки соответствия между цветами на мониторе и цветами чернил принтера специально была разработана модель CMYK (Cyan — голубой, Magenta — пурпурный, Yellow — желтый, Black — Черный), в которой каждый цвет задается значениями четырех компонентов в процентном соотношении. Например, в данном формате черный цвет представляется как 0-0-0-100.

При разработке изображений для вывода на экран модель CMYK не используется. Вместо нее применяется модель RGB (Red — красный, Green — зеленый, Blue — синий), в которой каждый цвет определяется комбинацией значений трех базовых цветов.

Для передачи изображений по сети существует еще одна модель цветов — *Hexadecimal notation*, или *Hex* (шестнадцатеричная система). На самом деле это, скорее, не модель, а формат представления RGB-цветов. В данном случае цвет описывается с использованием шести символов (например, OODDFF). Первые два символа представляют канал красного цвета, последующие два — канал зеленого цвета, последние два — канал синего цвета. Каждый цифровой (0–9) и буквенный символ (A–F) представляет целочисленное значение от 0 до 16. В результате, в шестнадцатеричном формате в точности передается RGB-значение.

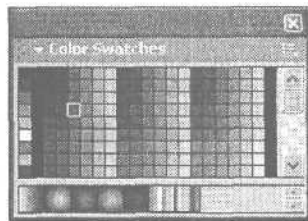
Использование цветов для Web

Большинство компьютеров способны выводить графическую информацию, используя 256 цветов (в современных компьютерах применяются миллионы цветов). Проблема состоит в том, что нельзя точно сказать, *какие* это 256 цветов. Такая неопределенность является проблемой для Web-дизайнеров. К счастью, большинство современных Web-браузеров (Netscape Navigator, Internet Explorer и т. д.) использует стандартизированную (безопасную) цветовую палитру — *browser-safe palette*, называемую иногда *web-safe palette*. Благодаря этой палитре у разработчиков есть гарантия, что цветовая гамма изображения будет корректно воспроизведена на другом компьютере.

Если выбранный цвет не входит в состав цветовой палитры Web 216, браузер конвертирует его в ближайший по значению цвет из своей палитры. В результате при передаче изображения на другой компьютер цветовая гамма может искажаться.

Работа с панелью Color Swatches

Каждый Flash-фильм имеет свою цветовую палитру. Flash выводит цветовую палитру файла в виде набора образцов (небольших цветных квадратиков) на панели Color Swatches.



Хотя по умолчанию на панели Color Swatches (открыть ее можно, активизировав команду Color Swatches меню Window) выводится безопасная палитра цветов, вы можете добавлять, удалять, редактировать и дублировать цвета в соответствии со своими потребностями. Наряду с этим существует возможность импортировать и экспортировать пользовательские палитры. Все эти операции производятся посредством команд всплывающего меню, доступ к которому можно получить, щелкнув мышью на значке в правом верхнем углу цветовой палитры.

Для редактирования палитры цветов доступны следующие команды:

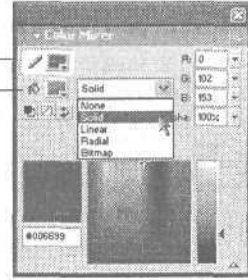
- ▶ Duplicate Swatch (Дублировать образец). Автоматически дублирует выбранный образец.
- ▶ Delete Swatch (Удалить образец). Удаляет выбранный образец.
- ▶ Add Colors (Добавить цвета). Позволяет импортировать цветовые палитры, которые были сохранены в формате CLR (Flash Color Set) или ACT (Color Table).
- ▶ Replace Colors (Замена цветовой палитры). Позволяет заменить текущую палитру цветов импортируемой. Выполните данную команду, а затем в появившемся окне Import Color Swatch укажите файл CLR.
- ▶ Load Default Colors (Загрузить цветовую палитру по умолчанию). Если вы уже изменяли цветовую палитру, используйте данную команду для загрузки цветовой палитры, установленной по умолчанию.
- ▶ Save Colors (Сохранить цветовую палитру). Сохраняет экспортированную цветовую палитру. Активизируйте команду и в появившемся диалоговом окне выберите каталог, в котором должен быть сохранен файл. Укажите тип файла и его имя, а затем щелкните на кнопке Save (Сохранить).
- ▶ Save as Default (Сохранить по умолчанию). Позволяет сделать текущую цветовую палитру палитрой по умолчанию, которая будет загружаться при выполнении команды Load Default Colors.
- ▶ Clear Colors (Очистить цвета). Автоматически удаляет все цвета из текущей палитры, за исключением черного и белого.
- ▶ Web 216. Загружает вместо текущей палитру Web 216.
- ▶ Sort by Color (Сортировать по цвету). Упорядочивает образцы цветов по оттенкам согласно с цветовым спектром.

Панель Color Mixer

Панель Color Mixer является средством создания и редактирования цветов, градиентов и растровых заливок. Ее основное предназначение — управление характеристиками заливок и обводок объектов.

Чтобы начать работу с панелью Color Mixer, активизируйте команду Color Mixer меню Window. Затем перейдите в режим редактирования обводки, щелкнув на палитре Stroke Color, либо в режим обработки заливки, щелкнув на палитре Fill Color. Оба значка расположены в верхнем левом углу панели Color Mixer. Далее следуйте инструкциям, изложенным ниже.

палитра
stroke Color
палитра
Fill Color



Смешение чистых цветов

Панель Color Mixer можно использовать для создания чистых цветов в моделях RGB и HSB (Hue — оттенок, Saturation — насыщенность, Brightness — яркость), а также в шестнадцатеричном формате. Она позволяет изменять и редактировать заливку как до, так и после создания объекта. Если необходимо изменить заливку существующего объекта, выделите данный объект посредством инструмента Arrow и измените характеристики его заливки на панели Color Mixer.

Новый цвет вы можете добавить в текущую палитру или в каталог цветов. Для этого надлежит вызвать команду Add Swatch меню Options на панели Color Mixer.

Создание цвета в формате RGB

Для получения RGB-цвета выполните следующие действия:

1. Выберите пункт Solid в меню Fill Style.
2. Выберите пункт RGB в меню Options на панели Color Mixer.
3. Введите числовые значения в поля R, G и B. Эти значения можно также задать с помощью ползунков, находящихся справа от полей. Альтернативный вариант — выбор цвета в палитре с помощью мыши. Соответствующие значения автоматически будут вставлены в полях цветовых каналов.
4. Чтобы изменить прозрачность цвета, введите требуемое значение в поле Alpha. Значение 0 соответствует полной прозрачности, а значение 100 — полной непрозрачности.

Создание цвета в формате HSB

Процесс создания цвета в формате HSB аналогичен описанному. Выберите пункт HSB в меню Options на панели Color Mixer. Введите числовые значения в поля H, S и B или щелкните на подходящем цвете в палитре. Определите прозрачность цвета в поле Alpha.

Смешение цвета в шестнадцатеричном формате

Смешение цвета в шестнадцатеричном формате также является простой процедурой. Все что вам необходимо сделать — это ввести шестнадцатеричное значение в поле Hex.

ПРИМЕЧАНИЕ

Шестнадцатеричное значение любого цвета, создаваемого с использованием модели RGB или HSB, автоматически подставляется в поле Hex.

Создание объектов без заливки

В случае выбора пункта None (Без заливки) в меню Fill Style все создаваемые объекты не будут иметь заливки. В отличие от других пунктов меню Fill Style, данный параметр нельзя присвоить объекту, который был создан с заливкой. Следовательно, если вы хотите, чтобы объект не имел заливки, параметр None необходимо выбрать до его создания.

Создание линейного градиента

Линейный градиент представляет собой заливку, в которой происходит смена цветов вдоль вертикальной или горизонтальной оси.

Далее рассматривается каждый этап процесса создания и редактирования линейного градиента с использованием панели Color Mixer.

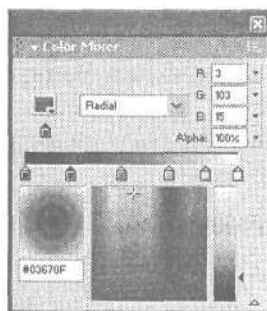
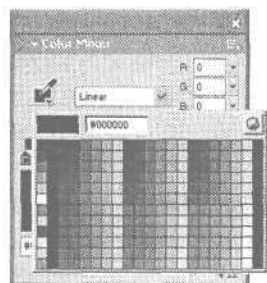
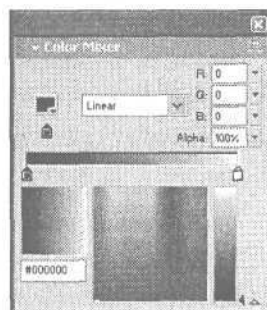
1. Выделите объект в области действия и откройте панель Color Mixer. В меню Fill Style (Стиль заливки) активизируйте команду Linear (Линейный градиент).

Учтите, что если пункт Linear выбрать *до* создания объекта, новый объект будет автоматически заполнен линейным градиентом, указанным ранее.

2. Чтобы изменить цвет в градиенте, щелкните на одном из бегунков ниже полосы градиента. Изначально имеются два бегунка для начального и конечного цветов.
3. Щелкните на образце цвета выше полосы градиента. Откроется цветовая палитра, в которой и осуществляется выбор нового цвета для градиента.
4. Переместите один из ползунков вдоль линейки. Чем дальше друг от друга находятся ползунки, тем более плавным является градиент. И наоборот: чем ближе друг к другу ползунки, тем резче границы между составными частями градиента.

При смещении ползунков или смене цветов градиента все изменения автоматически отображаются как в полосе градиента, так и в поле предварительного просмотра.

5. Для получения сложного и эффектного градиента добавьте в него новые цвета. Для этого выполните щелчок под полосой градиента, вследствие чего появится еще один ползунок. Измените цвет ползунка, выполнив действия, описанные в пункте 4.



Отредактированный градиент можно добавить на панель Color Swatches и впоследствии применять для оформления различных объектов. С этой целью активизируйте команду Add Swatch меню Options на панели Color Mixer.

Создание радиального градиента

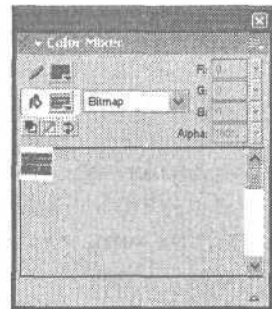
В радиальном градиенте смена цветов происходит от центра к краям. Он создается и редактируется так же, как и линейный.

Создание растровой заливки

Еще один вид заливки, которую вы можете создать с помощью панели Color Mixer, — растровая. Вот как это осуществляется:

1. Выделите объект в области действия и откройте панель Color Mixer. В меню Fill Style (Стиль заливки) выберите пункт Bitmap (Растровая заливка).

Учтите, что если команду Bitmap активизировать до создания объекта, он будет автоматически заполнен предварительно разработанной растровой заливкой.



2. Если вы еще не импортировали растровые изображения в свой фильм, откроется диалоговое окно Import to Library (Импортировать в библиотеку). Найдите файл, содержащий необходимое изображение, выделите его и щелкните на кнопке Open.
3. Если же растровые изображения импортировались ранее, они появятся в окне Bitmap Fill (Растровая заливка) в виде миниатюр.
4. Выберите необходимое изображение.
5. Для изменения положения растровой заливки используйте инструмент Fill Transform (Преобразование заливки).

Разработки

Условием разработки удачного проекта Flash является знание характеристик и возможностей инструментов рисования, черчения и управления цветом. Многие Flash-фильмы могут служить примером мастерского применения перечисленных инструментов. Один из них представлен в этом разделе. Это — Web-узел Peter Grafik Retro Experience (www.petergrafic.dk/dubonet_retro_experience/), разработанный Питером Холмом, участником проекта Titoonic, о котором мы расскажем в одноименном разделе главы 26 (рис. 4.7). Вашему вниманию предложена первая версия данного узла, с новой версией можно ознакомиться по адресу www.petergrafik.dk. Узел интересен удачной интеграцией великолепной графики в стиле мультипликации и общего интерфейса. Совершите путешествие по страницам рассматриваемого Web-узла, и вы убедитесь, насколько остроумно он организован.

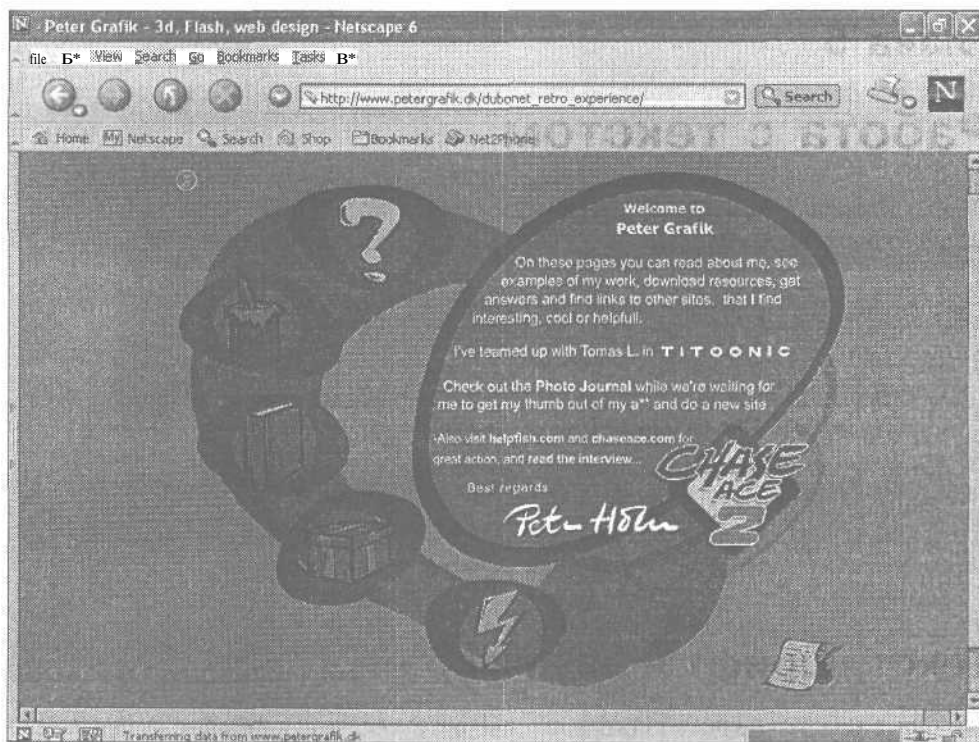


Рис. 4.7. Web-узел Peter Grafik Retro Experience — удачное использование цвета и иллюстраций

Резюме

В этой главе рассказано о возможностях инструментов Flash.

Любой процесс начинается с выделения объекта. Вам были представлены все предназначенные для этого инструменты: Arrow (Стрелка), Subselection (Дополнительный выбор) и Lasso (Лассо).

Совершая первые шаги в рисовании, вы научились чертить элементарные геометрические фигуры посредством инструментов Oval (Овал) и Rectangle (Прямоугольник).

От стандартных форм к произвольным мы перешли, рассмотрев инструменты рисования Pencil (Карандаш), Pen (Перо), Brush (Кисть) и Line (Линия).

Следующим шагом к профессиональной графике было изучение приемов цветового оформления объектов. Эти вопросы освещались в разделах, посвященных инструментам Eyedropper (Пипетка), Paint Bucket (Заливка), Ink Bottle (Чернила), Transform Fill (Преобразование заливки) и Eraser (Ластик).

Завершает главу важный материал о панелях Properties и Color Mixer, назначение которых — управление характеристиками обводки и заливки.

Глава 5

Работа с текстом

Текст — важная составляющая фильмов. Время, которое вы потратите на поиск удачной формы подачи текстового материала, с лихвой компенсирует успех всего проекта.

В главе рассматриваются следующие темы:

- ▶ текст, шрифт и гарнитура;
- ▶ ограничения при использовании текста в Flash-фильмах;
- ▶ использование инструмента Text;
- ▶ манипулирование текстом: панель Properties;
- ▶ разбивка текста;
- ▶ изменение формы текстовых символов;
- ▶ создание символов шрифта.

Текст, шрифт и гарнитура

Работая с текстом в среде Flash, вы обнаружите, что слова *текст*, *шрифт* и *гарнитура* часто используются как синонимы. Однако не стоит путать эти понятия, поскольку между ними имеются существенные различия.

Текстом называют любые комбинации символов (слова и предложения), которые составляют какой-либо документ. *Шрифт* — это набор символов (букв, цифр и специальных знаков, вводимых с помощью клавиш Shift, Option и Cmd/Ctrl) определенного размера и стиля. *Гарнитура* представляет собой комплект шрифтов, которые имеют одинаковый рисунок, но различаются начертанием и размером. Например, гарнитура Arial охватывает шрифты Arial (обычный), Arial Bold (полужирный), Arial Italic (курсив) и Arial Bold Italic (полужирный курсив).

СОВЕТ

Более подробную информацию о шрифтах в частности и печатном деле в целом вы можете найти на Web-узлах [www.typereview](http://www.typereview.com), www.rsub.com/typographic/ и <http://counterspace.motivo.com/>.

Ограничения при использовании текста в Flash-фильмах

Использование шрифтов при разработке Flash-фильмов связано с рядом проблем, которые требуют обсуждения. Эти проблемы не так сложны, как те, что характерны для проектов HTML, но все же достаточно важны и влияют на конечный результат.

В фильмах Flash можно использовать шрифты Type 1 PostScript и TrueType, а в среде Macintosh — еще и растровые шрифты. Для использования шрифтов PostScript в вашей системе должно быть установлено приложение Adobe Type Manager (ATM).

ПРИМЕЧАНИЕ

Если вы работаете в операционной системе Windows 2000, то для использования шрифтов Type 1 PostScript нет необходимости устанавливать Adobe Type Manager (ATM).

В большинстве случаев при публикации окончательного варианта проекта Flash вкладывает необходимую информацию о шрифтах, что обеспечивает корректный вывод текста перед зрителями. Однако текстовая информация все равно может выводиться неправильно, если на компьютере зрителя не установлены необходимые шрифты.

Предотвратить описанную проблему можно двумя способами. Во-первых, если для вас нет принципиальной разницы, какой шрифт использовать, выберите один из шрифтов типа `_sans`, `_serif` или `_typewriter`, доступных в меню панели Properties. Они называются *шрифтами устройства*. Используя их, вы указываете Flash применять эквивалентные шрифты из числа тех, что установлены на компьютере зрителя. Это гарантирует, что зритель увидит текст в надлежащей форме. Во-вторых, вы можете разбить текст на части и преобразовать их в отдельные объекты. Как это делается, рассказывается в разделе «Разбивка текста» этой главы.

Помимо того, что на машине пользователя могут отсутствовать шрифты, которые применяются в Flash-фильме, существуют и другие проблемы, связанные со шрифтами, о которых вам следует знать. На компьютерах с операционной системой Windows шрифты PostScript часто отображаются некорректно. Поэтому пользователям Windows мы рекомендуем использовать только шрифты TrueType.

Пользователи компьютеров с операционной системой Macintosh, в свою очередь, должны быть осмотрительны при использовании шрифтов Adobe PostScript. В большинстве случаев шрифты PostScript функционируют правильно, однако иногда возникают ситуации, когда использование таких шрифтов приводит к возникновению проблем. Порой шрифты PostScript выводятся правильно в процессе создания Flash-фильма, а после публикации работы отображаются некорректно.

Использование инструмента Text

Теперь, когда вам известно, с какими ограничениями связано использование шрифтов в фильмах Flash, можно приступить к рассмотрению процесса создания текста с помощью инструмента Text. Данный инструмент расположен на панели Tools в секции Tools.

Хотя большая часть операций с текстом производится посредством панели Properties (о которой мы расскажем позже), на первых этапах вы всегда будете использовать инструмент Text.

Режим сглаживания текста включается/отключается с помощью команды View ► Antialias.

Чтобы начать ввод символов с клавиатуры, щелкните мышью сначала на инструменте Text, а затем — в позиции ввода на рабочей области. Не стоит на первой стадии создания беспокоиться о том, как выглядит текст. С приемами оформления текста вы познакомитесь в разделе «Манипулирование текстом: панель Properties».

СОВЕТ

Если возникнет необходимость в редактировании текста, активизируйте инструмент Text и выполните щелчок на текстовом поле (либо двойной щелчок, если выбран инструмент Arrow). В текстовом поле появится курсор — признак того, что надпись можно изменить.

Текстовые поля

Текст в среде Flash создается в блоках, которые называются текстовыми полями. Текстовое поле — это прямоугольная рамка с небольшим кружочком в правом верхнем углу.

Поле, содержащее текст, называется текстовым полем

Его можно перемещать по рабочей области. Различают три типа текстовых полей. Каждый из них обладает определенными свойствами в процессе создания и манипулирования текстом в среде Flash.

Расширяемое текстовое поле

Фиксированное текстовое поле

Текстовое поле ввода

Расширяемое текстовое поле увеличивается по мере ввода текста. Такое поле легко узнать по небольшому кружочку в правом верхнем углу. Поля этого типа создаются по умолчанию.

Фиксированное текстовое поле не расширяется по горизонтали при вводе текста. Признак полей этого типа — квадратик в правом верхнем углу. Чтобы создать фиксированное текстовое поле, выберите инструмент Text, щелкните мышью в любом месте рабочей области и растяните текстовое поле до требуемого размера. Обратите внимание на то, что поле растягивается только по горизонтали. При выполнении ввода с клавиатуры горизонтальный размер текстового поля останется неизменным, а слова будут автоматически переноситься в следующую строку, тем самым увеличивая размер поля по вертикали.

Текстовое поле *ввода* содержит динамический текст, который может быть изменен зрителем вашего фильма.

Подробно о том, как создавать и работать с динамическим текстом, рассказывается в разделе «Динамический текст».

Поля третьего типа применяются в тех случаях, когда необходимо создать элемент для ввода данных пользователем, например пароля, открывающего доступ

к конфиденциальной информации на Web-узле. Для создания текстового поля ввода недостаточно инструмента Text. Чтобы сделать текст динамическим, необходимо выбрать команду Dynamic Text (Динамический текст) или Input Text (Текст ввода) в раскрывающемся списке Text Type (Тип текста) на панели Properties. Текстовое поле ввода обозначается квадратиком в правом нижнем углу прямоугольной рамки.

Манипулирование текстом: панель Properties

MX В процессе разработки фильма текст создается посредством инструмента Text, а его редактирование производится с помощью панели Properties. В отличие от предыдущих версий Flash, в которых для управления различными свойствами текста использовались разные панели, в Flash MX существует универсальная панель Properties. Данная панель открывается при выборе команды Properties в меню Window.

Текст в фильмах Flash MX может быть трех различных типов: статический, динамический и вводимый. Текст каждого из этих типов имеет свои свойства, с которыми мы познакомимся далее.

Давайте пройдем все этапы процессов создания и редактирования текста каждого типа.

1. С помощью инструмента Text создайте на рабочей области какой-либо текст.
2. Если текстовое поле осталось «активным», выберите текст с помощью курсора. Если же вы отменили выделение текстового поля, выберите его повторно посредством инструмента Arrow (вокруг текстового поля должна появиться светло-голубая рамка).
3. Откройте панель Properties, активизировав команду Properties меню Window.
4. С помощью раскрывающегося списка Text Type на панели Properties выберите требуемый тип текста.

Раскрывающийся список Text Type



С этого момента на панели Properties будут отображаться все специфические свойства выбранного вами типа текста.

СОВЕТ -

Далее описано, как манипулировать текстом после его создания. Если вы желаете создать текст с заранее определенными характеристиками, выберите инструмент Text на панели Tools, откройте панель Properties, установите все параметры и только после этого создайте текст в рабочей области. Вводимому тексту будут автоматически присваиваться атрибуты (шрифт, цвет, размер и т. д.), установленные на панели Properties.

Работа со статическим текстом

Статический текст создается по умолчанию и используется исключительно для передачи информации. Это самый распространенный тип текста. Данный раздел содержит обзор всех свойств и характеристик статического текста, доступ к которым предоставляется на панели Properties.



Параметры, рассматриваемые далее, доступны для настройки только на панели Properties.

Раскрывающийся список Font

В списке Font (Шрифт) перечисляются все установленные на компьютере пользователя шрифты. Если активен инструмент Text, то в связанном со списком поле указывается шрифт выделенного в настоящий момент текста. Выбрать необходимый шрифт в раскрывающемся списке Font можно либо перед вводом текста, либо после выделения существующего текста (или всего поля). Изменить шрифт выделенного текста можно и в подменю Font меню Text.

Поле Font Size

Данное поле служит для установки размера шрифта. Вы можете либо ввести значение размера шрифта в пунктах, либо задать его с помощью бегунка, щелкнув на небольшой стрелке, указывающей вниз, справа от поля Font Size. Возможность изменить размер шрифта предоставляется также в подменю Size меню Text.

Кнопки Text style

Указанные в заголовке кнопки позволяют сделать шрифт полужирным или курсивным. Как и другие параметры, устанавливаемые на панели Properties, стиль текста можно определить либо до его создания, либо после, выделив текст с помощью инструмента Aggow или Text и нажав соответствующую кнопку.

К сожалению, при использовании панели Properties доступен ограниченный набор стилей шрифта. Расширенный список отображается в результате вызова команды Style (Стиль) меню Text.

Палитра Text Color

Благодаря палитре Text Color вы можете задать необходимый цвет для текста. Если вы не удовлетворены имеющимся выбором, создайте собственный цвет, используя цветовое поле, доступ к которому можно получить, щелкнув на небольшом цветном кружочке в правом верхнем углу палитры цветов.

Межзнаковый интервал

Данное значение характеризует расстояние между символами в строке текста. Чем больше значение межсимвольного интервала, тем дальше друг от друга расположены символы. На рис. 5.1 демонстрируются различные межзнаковые интервалы. Регулирование межзнаковых интервалов иногда называют *трекингом*.

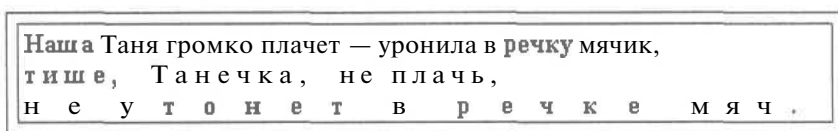


Рис. 5.1. Верхняя строка — трекинг 0, средняя строка — трекинг 10, нижняя строка — трекинг 15

На панели Properties можно установить значение межзнакового интервала для любого блока текста (либо до, либо после его создания). Если текст уже создан, необходимо выделить текстовое поле или фрагмент текста, а затем ввести значение трекинга в поле Character Spacing (Межзнаковый интервал). Кроме того, вы можете использовать бегунок, щелкнув на небольшой стрелке, указывающей вниз, справа от поля Character Spacing.

Расположение символа

Речь идет о положении символа относительно базовой линии. В графических программах этот параметр часто называют *baseline shift* (смещение относительно базовой линии). В отличие от большинства других графических программ, в которых разрешается устанавливать точное значение смещения относительно базовой линии, Flash предлагает на выбор только три варианта расположения: Normal (Обычный), Superscript (Верхний индекс) и Subscript (Нижний индекс).



Рис. 5.2. Три варианта смещения относительно базовой линии

Чтобы изменить положение символов, выделите строку текста и выберите один из пунктов раскрывающегося списка Character Position. На панели Properties значение Character Position (Положение символов) можно установить до создания текста с помощью инструмента Text.

Преобразовать символы в верхний или нижний индексы позволяют также команды Text ▶ Style ▶ Superscript и Text ▶ Style ▶ Subscript.

Режим Auto Kern

Данный режим служит для автоматического выравнивания интервалов между отдельными символами строки. При выборе флажка Auto Kern (Автоматическая установка межзнакового интервала) активизируется режим автоматического выравнивания межсимвольных интервалов для многих (но не для всех) шрифтов.

ПРИМЕЧАНИЕ

Функция автоматического выравнивания межсимвольных интервалов может быть применена только к тем шрифтам, информация о которых включена в файл Flash-фильма.

Кнопка Change Direction of Text

Существуют несколько вариантов расположения текста. Доступ к их перечню предоставляет кнопка Change Direction of Text (Изменение направления текста). По умолчанию текст размещается по горизонтали слева направо, что соответствует варианту Horizontal (Горизонтально). Режимы Vertical, Left to Right (Вертикально, слева направо) и Vertical, Right to Left (Вертикально, справа налево) предназначены для ввода текста по вертикали. Рассмотрим, в чем заключается различие между ними. При выборе режима Vertical, Left to Right в блоке текста, содержащем больше одной строки, первая строка остается первой вертикальной строкой. В режиме Vertical, Right to Left таковой становится последняя строка текста.

Кнопка Rotation

С помощью кнопки Rotation можно изменить ориентацию символов в вертикальных блоках текста. Если данная кнопка отключена, символы отображаются один над другим обычным образом, а если нажата — поворачиваются на 90 градусов по часовой стрелке.

Поле URL

В Flash предлагается несколько способов связывания текста с URL (Uniform Resource Locator — унифицированный указатель информационного ресурса). Наиболее простой — указать адрес в поле URL панели Properties.

Выберите строку текста, которую нужно преобразовать в ссылку, и введите URL-адрес в поле URL. С URL можно связывать только горизонтальный текст. Любой Flash-текст, с которым вы связали URL, выделяется пунктирным подчеркиванием.

Поле URL Target

Поскольку существует вероятность, что ваш Flash-фильм будет перенесен в структурированный HTML-документ, в раскрывающемся списке Target предусмотрена возможность указать, где должен загружаться URL-объект. В этом списке на выбор предлагаются четыре варианта:

- ▶ `_blank` — документ загружается в новом окне браузера; окно с гиперссылкой располагается под вновь открытым окном.
- ▶ `_parent` — документ загружается в родительском фрейме текущего кадра.



- ▶ `_self` (установка по умолчанию) — документ загружается в том окне, где находится ссылка.
- ▶ `_top` — документ загружается в верхнем фрейме текущего окна.

Вы можете самостоятельно ввести имя фрейма в поле списка Target, если хотите указать имя определенного фрейма в HTML-документе.

Кнопка Selectable

Обычно статический текст является частью Flash-фильма и, следовательно, не может быть выбран в документе HTML. Но если вы хотите, чтобы ваши зрители имели возможность выбрать статический текст в фильме (что позволит им скопировать и вставить его в другой документ), убедитесь, что кнопка Selectable (Выбираемый) нажата.

Режим Use Device Fonts

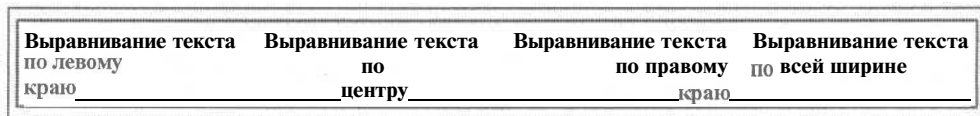
Выбрав флажок Use Device Fonts (Использовать шрифты устройства), вы указываете Flash не использовать определенный шрифт в заданном текстовом поле. Вместо этого Flash для вывода текстовой информации будет применять наиболее подходящий шрифт, установленный на компьютере зрителя. Поскольку информация о шрифте не будет добавлена в файл SWF, его размер немного сократится.

ПРИМЕЧАНИЕ

Режим Use Device Fonts доступен только при работе с горизонтальным текстом.

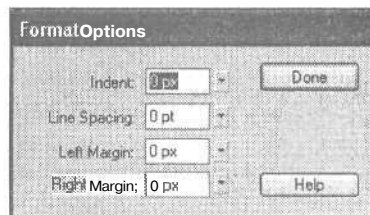
Кнопки выравнивания текста

В правой верхней части панели Properties расположены четыре кнопки, предназначенные для выравнивания текста: Left/Top (По левому краю), Center (По центру), Right/Bottom (По правому краю) и Fully Justify (По ширине). Чтобы выровнять текст, выберите текстовое поле или строку в пределах текстового поля и выполните щелчок на той кнопке, которая соответствует подходящему способу выравнивания.



Параметры форматирования

Щелкнув на кнопке Format (Изменить настройки форматирования), вы получаете доступ к диалоговому окну Format Options (Опции форматирования), которое содержит несколько полей со значениями, влияющими на внешний вид всего текстового поля (а не отдельных символов).



В поле Left Margin (Левое поле) определяется расстояние (в пикселах) между левой стороной текстового поля и самим текстом. Числовое значение в это поле либо вводится с клавиатуры, либо устанавливается посредством бегунка. Поле Right Margin (Правое поле) предназначено для указания интервала (в пикселах) между правой стороной текстового поля и самим текстом. Приемы работы с ним те же.

В поле Indent (Отступ) задается отступ первой строки текстового блока.

С помощью поля Line Spacing (Межстрочное расстояние) регулируют величину интервала между строками текста. И в этом случае значение либо вводится с клавиатуры, либо устанавливается посредством бегунка.

Работа с динамическим текстом

Динамический текст в поле обновляется при внесении изменений в источник. Источником может быть база данных или текстовый файл, расположенный на вашем сервере. Динамический текст создается таким же способом, что и статический. То, что динамический текст не является фиксированным, значительно расширяет сферу его применения. Его используют для решения разнообразных задач. Вы можете, например, создать постоянно обновляющийся список любимых шуток или ежедневных маршрутов в сети.

Более подробно об использовании динамического текста мы расскажем в главе 17.

Если вы хотите создать и обработать динамический текст, вам необходимо выбрать пункт Dynamic Text в раскрывающемся списке Text Type на панели Properties.



С этого момента с помощью панели Properties вы получаете доступ к специфическим свойствам динамического текста, описанным ниже.

Раскрывающийся список Line Type

Данный список содержит три пункта: Single Line (Однострочный), Multiline (Многострочный) и Multiline No Wrap (Многострочный без переносов). В режиме Single Line текст выводится в одной строке. Различие между режимами Multiline и Multiline No Wrap состоит в том, что в последнем режиме переход в следующую строку осуществляется только в том случае, если последним символом является символ разрыва строки, например Return/Enter.

Поле Variable

Поле Variable (Переменная) предназначено для указания имени текстового поля. Данный идентификатор необходим для того, чтобы обеспечить точное соответствие между динамическим текстом и отведенным для него полем. Имя, вводимое в поле Variable, должно быть уникальным.

Кнопка Selectable

В случае применения данной кнопки зрители получают возможность выделять и копировать текст, находящийся в динамическом текстовом поле.

Кнопка Render Text as HTML

Данная кнопка применяется для сохранения стилей форматирования HTML. К сожалению, не все теги HTML удастся сохранить, поскольку поддерживаются только некоторые теги: bold (полужирный), italic (курсив), underline (подчеркивание), font face (направление шрифта), font color (цвет шрифта), paragraph break (конец абзаца) и font size (размер шрифта).

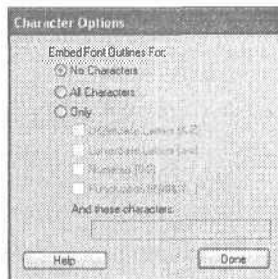
Кнопка Show Border around Text

Чтобы заключить текстовое поле в рамку, следует активизировать кнопку Show Border around Text (Показывать рамку вокруг текста). По умолчанию динамический текст на экране окружается пунктирной рамкой, которая не выводится на печать.

Кнопка Edit Character Options

В результате выполнения щелчка на кнопке Edit Character Options (следует отметить, что на данной кнопке видна только надпись Character...) открывается диалоговое окно Character Options (Опции символов).

В этом окне можно точно определить, сколько шрифтовых символов будет вложено в Flash-файл. При выборе переключателя No Characters (Нет символов) в файл не будут добавляться символы. Переключатель All Characters (Все символы) обеспечивает вложение в файл всех символов. Наконец, переключатель Only (Только) позволяет точно указать, какие именно символы должны быть включены: Uppercase (Символы верхнего регистра), Lowercase (Символы нижнего регистра), Numbers (Цифры) или Punctuation (Знаки пунктуации).



ПРИМЕЧАНИЕ

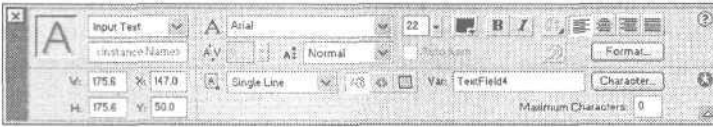
Чем меньше символов вложено в файл, тем меньше его размер.

Работа с текстовыми полями ввода

Единственное различие между текстом ввода и динамическим текстом связано с тем, в каком направлении передаются данные.

Текст ввода передается с пользовательского компьютера на сервер, а динамический текст — с сервера на компьютер пользователя. В связи с этим большинство

параметров текста ввода и динамического текста совпадают. Текст ввода обладает двумя дополнительными атрибутами. Рассмотрим их.



Режим Password

Для текста ввода в раскрывающемся списке Line Type (Тип строки) наряду с режимами Single Line (Однострочный), Multiline (Многострочный) и Multiline No Wrap (Многострочный без переносов) доступен еще один режим — Password (Пароль). Данный режим необходим, чтобы вводимые пользователем символы в поле замещались звездочками.

Действие команды Password распространяется только на символы, вводимые пользователем. Чтобы обеспечить корректную обработку вводимого пароля, вам необходимо написать сценарий с помощью языка ActionScript для передачи информации приложению на сервере.

Поле Maximum Characters

В данном поле определяется максимальное количество символов, которые пользователь может ввести в текстовое поле.

Разбивка текста

Для получения полного контроля над каждым символом текстового блока необходимо произвести его разбивку, вследствие которой все символы преобразуются в объекты. В этом случае вы сможете выполнять независимое форматирование каждого символа (например, изменять шрифт или увеличивать размер), трансформировать символы различными способами и т. д.

ПРИМЕЧАНИЕ

Выполняя разбивку текста, необходимо помнить о двух вещах. Во-первых, разбивка приводит к значительному увеличению размера файла Flash-фильма. Во-вторых, после разбивки нельзя редактировать текст.

Данная проблема легко решается:

1. Выберите текстовое поле с помощью инструмента Arrow.
2. Активизируйте команду Break Apart меню Modify.
3. В результате текстовый блок будет разбит на отдельные символы, которые можно редактировать, перемещать и трансформировать по отдельности.



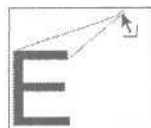
Разбивка особенно полезна, когда необходимо анимировать отдельные символы текстового блока большого объема.

Изменение формы текстовых символов

Одним из замечательных свойств Flash является возможность изменять форму отдельных символов заданного шрифта. Если вам, например, не нравится, как выглядит какая-либо буква в шрифте Arial, вы можете изменить ее вид. Вот как решается эта задача:

1. Создайте блок текста.
2. Выберите текстовое поле с помощью инструмента Arrow.
3. Активизируйте команду Break Apart меню Modify, вследствие чего текст будет разбит на множество групп, каждая из которых содержит отдельный символ.
4. Еще раз вызовите команду Break Apart.
5. Отмените выделение, щелкнув в любом месте области действия либо выбрав команду Deselect All меню Edit.

В результате все символы будут преобразованы в объекты с обводкой и заливкой. Теперь можно изменять форму любого символа с помощью инструмента Arrow.



Интересного эффекта можно добиться, назначив новый цвет обводке каждого символа посредством инструмента Ink Bottle (Чернила). Такая возможность существует, если произведена разбивка текста.

ПРИМЕЧАНИЕ

Команду Break Apart следует применять только к шрифтам TrueType. Растровые шрифты исчезают после разбивки, а шрифты PostScript можно разбить лишь в системах Macintosh, где запущен диспетчер печати компании Adobe (ATM).

Создание шрифтовых символов

На практике удобны совместно используемые библиотеки. Такие библиотеки хранят на сервере. К ним может быть подключено любое число Flash-фильмов, в файлы которых не вложены шрифты.

Если вы планируете создать библиотеку для совместного использования (эта тема освещается в главе 7), вам необходимо знать, как создаются шрифтовые символы (символы типа Font):

1. Откройте фильм, в котором вы хотите применять символы из совместно используемой библиотеки.
2. Загрузите окно Library (Библиотека), активизировав команду Library меню Window.
3. Выберите пункт New Font в меню Options.
4. Когда появится диалоговое окно Font Symbol Properties (Свойства символа Font), введите имя нового шрифта в поле Name. Указанное вами имя будет не официальным именем шрифта, а идентификатором.



5. Выберите шрифт в раскрывающемся списке Font, а также, в случае необходимости, полужирное или курсивное начертание.
6. Щелкните на кнопке ОК.

Разработки

Важную роль при создании Web-узла играет грамотное применение текстовых объектов. Ярким примером удачного их использования является Web-узел *typographic* (www.typographic.com). Его автор — Джимми Чен, талантливый дизайнер, проживающий в Лос-Анджелесе. Каждый квадратный сантиметр этого проекта представляет собой образец замечательного дизайна (рис. 5.3). Комбинации титров и надписей способны надолго задержать внимание посетителя.



Рис. 5.3. Web-узел *typographic*

Резюме

В главе освещались вопросы работы с текстовыми объектами в среде Flash. Обработка текста в программе Flash связана с рядом ограничений, которые необходимо знать, чтобы с точностью воплощать в фильмах свои творческие замыслы. Основные средства работы с текстом — инструмент Text и панель Properties. Большая часть операций с текстом осуществляется посредством панели Properties, но на этапе создания всегда используется инструмент Text. Символы текста можно редактировать как графические объекты, для чего требуется произвести разбивку текстового блока.

Глава 6

Работа с объектами

Успех Flash-проекта во многом зависит от вашего умения управлять внешним видом объектов. Что же такое объект? Это любой графический элемент, который вы создаете или импортируете: блок текста, растровый рисунок или геометрическая фигура.

Программа Flash обладает мощным арсеналом инструментов, предназначенных для модификации объектов. Разработчики внедрили в свой продукт средства для выполнения самых разнообразных операций: трансформации объектов, перемещения их по рабочей области, группирования с другими объектами и т. д.

Тема настоящей главы — способы управления внешним видом объектов в среде Flash. Вашему вниманию предлагается материал по следующим темам:

- ▶ понятие объекта;
- ▶ работа с растровыми изображениями;
- ▶ группирование объектов;
- ▶ позиционирование объектов;
- ▶ трансформация объектов;
- ▶ точки трансформации.

Понятие объекта

Все, что вы создаете или импортируете, Flash рассматривает как объекты. Таким образом, объектом может быть как фотография в формате JPEG, так и схема Солнечной системы, созданная вами с помощью инструментов рисования и черчения.

В последующих разделах мы расскажем об инструментах и процедурах, которые позволят вам манипулировать внешним видом отдельных объектов.

ПРИМЕЧАНИЕ —

Термин object (объект) используется также в языке ActionScript. Не следует путать эти два понятия. Объекты ActionScript и объекты, обсуждаемые в этой главе, различны. Подробно об объектах ActionScript рассказывается в главе 14.

Работа с растровыми изображениями

Хотя Flash-фильмы основываются на использовании векторной графики, в процессе разработки вы можете также применять растровые изображения. Программа Flash снабжена многочисленными инструментами, позволяющими импортировать растровые изображения и манипулировать ими. Важно не только освоить приемы работы с этими инструментами, но и понять природу растровых изображений, а также то, как Flash работает с ними.

Растровые, или побитовые, изображения состоят из отдельных точек, *пикселей*, имеющих фиксированный размер и занимающих определенный объем памяти. Качество растрового рисунка зависит от его разрешения. В большинстве случаев эта величина выражается в точках на квадратный дюйм (dpi — dots per inch). Растровые рисунки с высоким разрешением (например, 300 dpi) отличаются лучшим качеством от рисунков с низким разрешением (допустим, 72 dpi).

Еще одной особенностью растрового изображения является его зависимость от установленного разрешения. Рисунок сохраняет свое качество только при заданных размере и разрешении. При изменении размеров изображения ухудшается его качество, что проявляется в возникновении эффекта *ступенек*. С другой стороны, значительное увеличение растрового изображения приводит к тому, что становятся видимыми отдельные пиксели; и в этом случае проявляется эффект ступенек, поскольку число пикселей, приходящихся на один квадратный дюйм, неизменно. В процессе масштабирования векторных рисунков производятся математические вычисления, благодаря которым изображение не искажается. На рис. 6.1 демонстрируется различие между масштабированными растровым и векторным рисунками.

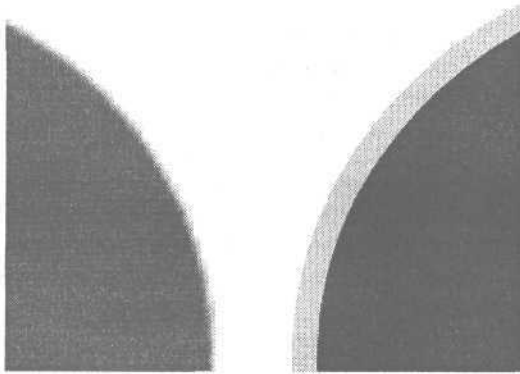


Рис. 6.1. При увеличении пиксели растрового рисунка (слева) становятся видимыми, а края векторного рисунка (справа) остаются ровными

Из-за этих свойств растровые изображения занимают больше памяти, чем векторные. Таким образом, в процессе создания Flash-фильмов следует отдавать предпочтение векторной графике.

Однако в ряде случаев растровая графика предпочтительнее. Векторы не позволяют получать изображения, по реалистичности близкие к фотографиям, а также создавать цветовые переходы с богатой палитрой оттенков.

Так как все инструменты черчения и рисования Flash рассчитаны на векторную графику, растровые рисунки придется импортировать. К счастью, Flash поддерживает множество форматов растровых изображений. После импортирования растровыми рисунками можно манипулировать, чтобы уменьшить занимаемый ими объем памяти.

Импортирование растровых изображений

В Flash существует два способа импортирования растровых изображений: посредством команды **Import** меню **File** и путем вставки изображения в область действия.

Flash позволяет импортировать файлы растровых изображений следующих форматов: **Windows Bitmaps (BMP/DIB)**, **GIF (GIF)**, **JPEG (JPG)**, **PNG (PNG)** и **PICT (PCT/PIC)**. Если на вашем компьютере установлено приложение **QuickTime** версии 4 или более поздней версии, список доступных форматов таков: **MacPaint (PNTG)**, **Quick-Time Image (QTIF)**, **Photoshop (PSD)**, **Silicon Graphics Images (SGI)**, **TGA (TGA)** и **TIFF (TIF)**.



Установочные файлы приложения **QuickTime 5** вы найдете на прилагаемом компакт-диске.

Давайте импортируем первое растровое изображение:

1. Активизируйте команду **Import** меню **File** или нажмите клавиши **Cmd/Ctrl+R**.
2. Когда откроется диалоговое окно **Import**, укажите файл, который вы хотите импортировать, и щелкните на кнопке **Open**.

СОВЕТ

Чтобы в диалоговом окне отображались файлы только определенного типа, отметьте расширение искомого файла в меню **Files of Type**.

3. После импортирования растровое изображение автоматически помещается как в область действия, так и в библиотеку (в качестве символа). Если вы хотите, чтобы импортируемое растровое изображение было помещено только в библиотеку, вызовите команду **Import to Library** меню **File**.

Обязательным условием импортирования **TIFF**-файлов является наличие на компьютере приложения **QuickTime**. Если у вас установлено данное приложение, при попытке импортировать **TIFF**-файл, например **image.tiff**, Flash сгенерирует дополнительный запрос: «Flash doesn't recognize the file format of image.tiff. Would you like to try importing via QuickTime?» («Flash не может распознать формат файла **image.tiff**. Хотите ли вы попытаться импортировать файл посредством приложения **QuickTime**?»). Выберите **Yes (Да)**, и ваш файл будет успешно импортирован. Если же в вашей системе приложение **QuickTime** отсутствует, Flash выдаст такое сообщение: «One or more files were not imported because there were problems reading them» («Один или несколько файлов не были импортированы, поскольку возникли проблемы при их чтении»).

Чтобы скопировать растровое изображение из другого приложения, выполните следующие действия.

1. Скопируете изображение в приложении-источнике, активизировав команду **Copy** меню **Edit** или нажав клавиши **Cmd/Ctrl+C**.
2. Перейдите в окно **Flash** и произведите вставку растровой графики посредством команды **Paste** меню **Edit** или комбинации клавиш **Cmd/Ctrl+V**.

Трассировка растровых изображений

Команда Trace Bitmap (Трассировка растрового изображения) позволяет преобразовывать растровые рисунки в характерный для Flash векторный формат с дискретными цветовыми зонами. Данное свойство особенно полезно, если вы хотите манипулировать растровым изображением как векторным. В большинстве случаев за счет трассировки растровых изображений и преобразования их в векторную графику уменьшается размер файла.

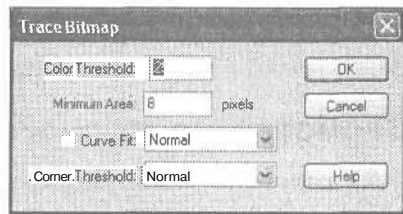
ПРИМЕЧАНИЕ

При трассировке связь между растровым рисунком и символом в библиотеке прерывается. Чтобы поместить растровый рисунок в библиотеку после трассировки, следует конвертировать его в символ. Данный процесс описан в главе 7.

Не следует, однако, думать, что после трассировки растровый рисунок выглядит так же, как и до нее. При таком преобразовании практически невозможно сохранить идентичность. Рассмотрим каждый этап трассировки растрового изображения:

1. С помощью инструмента Aggow выделите растровый рисунок в области действия.
2. Откройте диалоговое окно Trace Bitmap, вызвав команду Trace Bitmap меню Modify.

В этом окне предоставляется доступ к параметрам, которые влияют на ход трассировки. Остановимся на каждом параметре в отдельности.



Цветовой порог

Цветовой порог — это характеристика, определяющая количество цветов в растровом рисунке после трассировки и преобразования в векторное изображение. При трассировке сравниваются цвета смежных пикселей в растровом рисунке. Если разница между значениями RGB-цветов меньше величины, указанной в поле Color Threshold (Цветовой порог), цвета смежных пикселей рассматриваются как одинаковые. Следовательно, описываемая процедура сопровождается тем, что значения цветов растрового рисунка усредняются. Диапазон допустимых значений в поле Color Threshold — от 1 до 500. Чем выше значение порога, тем больше значений цветов будет усреднено и, следовательно, тем меньше цветов будет иметь трассированное изображение. Наоборот, чем меньше значение в поле Color Threshold, тем более многоцветным будет итоговое изображение. На рис. 6.2 показаны результаты трассировки при использовании различных значений цветового порога.



Рис. 6.2. Исходное растровое изображение (слева) и трассированные изображения со значениями параметра Color Threshold 25, 50 и 75

Поле **Minimum Area**

В этом поле указывается величина радиуса (от 1 до 1000), на протяжении которого сравниваются цвета смежных пикселей с использованием параметра **Color Threshold**. Чем выше значение в поле **Minimum Area**, тем больше смежных пикселей будет охвачено при выполнении данной операции.

Аппроксимация кривой

В меню **Curve Fit** (Аппроксимация кривой) определяется способ рисования контуров трассированного изображения. Возможны такие варианты: **Pixels** (На уровне пикселей), **Very Tight** (Очень плотно), **Tight** (Плотно), **Normal** (Обычно), **Smooth** (Сглажено) и **Very Smooth** (Очень сглажено). Если изображение содержит много кривых линий, вид которых нужно сохранить, выберите пункт **Tight** или **Very Tight**, а если вопрос сохранения кривых вас не слишком беспокоит — пункт **Smooth** или **Very Smooth**.

Угловой порог

Регулируя этот параметр, вы задаете, как процесс трассировки будет влиять на углы растрового изображения. При выборе пункта **Many Corners** (Много углов) в меню **Corner Threshold** (Угловой порог) углы сохраняются. Команда **Few Corners** (Мало углов) обеспечивает сглаживание углов изображения.

Свойства растрового изображения

Каждое импортируемое растровое изображение имеет ряд свойств, которыми вы можете манипулировать.

ПРИМЕЧАНИЕ

При установке свойств растровых изображений приходится работать с библиотеками, о которых рассказывается в главе 7.

Приступим к манипуляциям с растровым изображением.

1. Загрузите окно **Library** (Библиотека), выбрав команду **Library** в меню **Window** либо нажав клавишу **F11**.
2. Чтобы открыть диалоговое окно **Bitmap Properties** (Свойства растрового изображения), выделите растровое изображение и выполните одно из следующих действий:
 - ▷ Щелкните на значке **Properties**, который расположен в нижней части окна **Library** (третий слева с буквой **i**).
 - ▷ Щелкните правой кнопкой мыши (**Windows**) или выполните щелчок при нажатой клавише **Ctrl** (**Macintosh**) и активизируйте команду **Properties** контекстного меню.
 - ▷ Активизируйте команду **Properties** меню **Options** в окне **Library**.

Ознакомимся с параметрами, к которым предоставляется доступ в окне **Bitmap Properties**.



Окно Preview

В данном окне осуществляется просмотр выбранного изображения. Если по размеру рисунок больше окна предварительного просмотра, вы сможете перемещать картинку с помощью мыши (указатель в этой области приобретает вид руки) для обзора всех фрагментов.

Поле Name

В поле Name отображается имя растрового изображения. В случае необходимости вы можете изменить его.

Когда растровому изображению присваивается новое имя в диалоговом окне Bitmap Properties (Свойства растрового изображения), название самого файла остается прежним, так как меняется только имя рисунка в библиотеке.

Области Path, Date и Dimensions

Перечисленные области содержат информацию о пути к файлу растрового изображения, дате создания файла и его размере.

Режим Allow Smoothing

В указанном режиме осуществляется размывка изображения.

Меню Compression

Меню Compression включает два пункта.

При выборе пункта Photo (JPEG) изображение сжимается в формате JPEG. Чтобы обеспечить качество сжатия, предусмотренное по умолчанию, активизируйте флажок Use Document Default Quality. Если требуется самостоятельно задать данный параметр, отключите флажок Use Document Default Quality и введите значение от 1 до 100 в поле Quality. Чем больше значение в поле Quality, тем выше качество рисунка. Следует учитывать, что при этом также увеличивается размер файла.

В результате выбора пункта Lossless (GIF/PNG) (Без потери качества) сохраняются все данные об изображении. Такое сжатие не наносит ущерба качеству изображения, однако, к сожалению, размер файла значительно возрастает.

Флажок Use Document Default Quality

Флажок Use Document Default Quality (Использовать качество документа, установленное по умолчанию) применяется для сохранения исходного качества рисунка. Он доступен только в том случае, если вы не нажимали кнопку Test.

Кнопка Test

Кнопка Test предназначена для проверки результатов сжатия. Соответствующая информация, в том числе о размерах файлов до и после сжатия изображения, выводится в нижней части диалогового окна Bitmap Properties (Свойства растрового изображения).

Кнопка Update

С помощью кнопки Update проверяют исходный файл рисунка на предмет выполнения изменений в нем. Если исходный файл был изменен, импортированный файл обновляется, чтобы соответствовать оригиналу.

Сделав все необходимые установки в окне Bitmap Properties (Свойства растрового изображения), нажмите кнопку ОК.

Редактирование растровых изображений

Импортировав растровое изображение в среду Flash, вы можете запустить программу, в которой оно создавалось, для редактирования данного объекта.

ПРИМЕЧАНИЕ

Редактирование растрового изображения в формате PNG возможно при условии, что оно было импортировано как сведенный рисунок (в диалоговом окне Fireworks PNG Import Setting (Параметры импортирования Fireworks PNG) установлен параметр Flatten Image (Свести изображение)), а не как редактируемый объект.

Чтобы отредактировать изображение в формате PNG с помощью приложения Macromedia Fireworks 3 или более поздней версии, выполните следующие действия.

1. Откройте окно Library, активизировав команду Library меню Window или нажав клавишу F11.
2. Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите в контекстном меню пункт Edit with Fireworks 4 (Правка с помощью приложения Macromedia Fireworks).

ПРИМЕЧАНИЕ

Если на вашем компьютере не установлена программа Macromedia Fireworks, команда Edit with Fireworks (Правка с помощью приложения Macromedia Fireworks) окажется недоступной.

3. В диалоговом окне Edit Image (Правка изображения) укажите, какой исходный файл должен быть открыт.
4. После запуска приложения Fireworks сделайте все необходимые изменения.
5. По завершении правки вызовите команду Update меню Select, вследствие чего программа Fireworks обновит файл.

Следующие операции должны быть выполнены, если для редактирования растрового изображения применяется другая программа (не Fireworks):

1. Откройте окно Library.
2. Активизируйте команду Edit with контекстного меню.
3. В диалоговом окне Select External Editor (Выбор внешней программы для правки) укажите, какую программу нужно использовать для правки изображения.
4. После запуска внешней программы редактирования выполните все необходимые изменения и сохраните файл.
5. Обновить изображение в библиотеке Flash можно двумя способами:
 - ▷ Выберите значок растрового изображения в библиотеке и активизируйте команду Update меню Options.
 - ▷ Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) на значке растрового изображения в библиотеке и активизируйте команду Update контекстного меню.

Группирование объектов

Процесс объединения множества объектов в единый блок называется *группированием*. Чтобы продемонстрировать, насколько важна эта операция, рассмотрим простой пример. Предположим, вы хотите создать изображение цветка для использования в фильме. Вы по отдельности создаете элементы цветка — лепестки, стебель, листья — и комбинируете их в области действия, пока не получите наиболее привлекательный и реалистичный вариант. Но что произойдет, если возникнет необходимость переместить *весь* цветок? Безусловно, вы можете выделить каждый элемент с помощью инструмента Arrow и после этого заняться позиционированием. Однако при таком рутинном процессе не исключена возможность, что какая-либо небольшая, но важная деталь будет упущена и останется в исходной позиции после перемещения. Работу придется повторять.

Не следует смеяться: такого рода ошибки часто допускают даже опытные пользователи. В подобной ситуации именно группирование объектов позволит избежать не только столь нелепой ошибки, но и монотонной работы. Если бы все элементы цветка были заранее сгруппированы, программа переместила бы их как единый объект. В этой связи напомним: группируйте обводку и заливку объекта, чтобы избежать их разъединения во время перемещения.

Вот как выполняется группирование.

1. С помощью инструмента Arrow выделите объекты в области действия, которые нужно включить в группу. Помните, что вы можете выбирать все, что находится в области действия: текст, геометрические фигуры, растровые рисунки и т. д.

СОВЕТ

Чтобы выбрать несколько объектов, в процессе выделения удерживайте нажатой клавишу Shift.

- Активизируйте команду Group меню Modify или нажмите клавиши `Cmd/Ctrl+G`. На рис. 6.3 показаны группы выбранных объектов.

Группу объектов окружает тонкий контур, как если бы это был один объект.

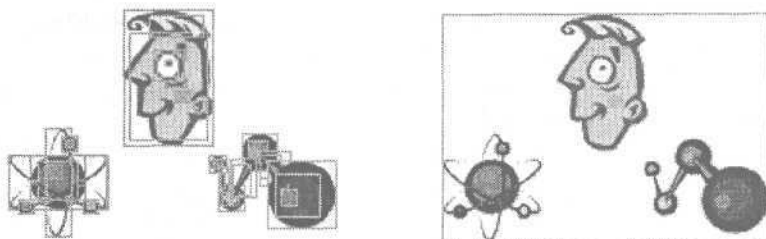


Рис. 6.3. Отдельные выделенные объекты (слева) и группа объектов (справа)

ПРИМЕЧАНИЕ

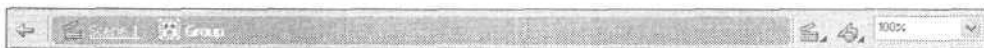
С помощью панели Properties можно идентифицировать, какие объекты были сгруппированы.

Разбить группу несложно.

- Выделите сгруппированные объекты с помощью инструмента Arrow.
- Вызовите команду Ungroup меню Modify или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+Shift+G`.

Особую ценность представляет возможность редактирования элементов группы (например, лепестков цветка) без применения команды Ungroup. Давайте остановимся на этой операции.

- Выберите группу объектов с помощью инструмента Arrow.
- Активизируйте команду Edit Selected (Правка выбранного) меню Edit. Цвет объектов, не являющихся членами группы, немного изменится. Это свидетельствует о том, что они временно недоступны. Обратите внимание на то, что в строке Scene and Symbol, которая расположена вдоль верхнего края области действия, появился значок Group — признак режима редактирования группы.



- Теперь вы можете приступить к обработке каких угодно объектов группы.
- В завершение вызовите команду Edit All меню Edit и, выбрав инструмент Arrow, выполните двойной щелчок в любом месте области действия за пределами группы. Можно также щелкнуть на имени сцены в строке Scene and Symbol.

Позиционирование объектов

При работе с объектами постоянно возникает необходимость перемещать их в пределах области видимости. Существует четыре способа выполнения данной операции: с помощью инструмента **Arrow**, клавиш управления курсором, посредством панели **Properties** или панели **Info**.

Далее описано, как для позиционирования объектов применять инструмент **Arrow**.

1. Выделите объект. Если переместить нужно несколько объектов, при их выделении удерживайте нажатой клавишу **Shift**.
2. Активизируйте инструмент **Arrow**. Перетащите объект или группу в новую позицию. **Flash** позволяет производить перемещение строго по вертикали или горизонтали. Для этого, выполняя операцию, удерживайте нажатой клавишу **Shift**.

СОВЕТ

Если в процессе перемещения удерживать нажатой клавишу **Option/Alt**, **Flash** создаст копию объекта (объектов) и разместит в указанном месте ее, а не оригинал.

Перемещение с помощью клавиш со стрелками производится так же просто:

1. Выделите объект (или группу объектов).
2. Нажимайте ту клавишу, которая соответствует требуемому направлению перемещения. При каждом нажатии объект смещается на 1 пиксел. Чтобы увеличить шаг смещения, удерживайте нажатой клавишу **Shift**. Тогда он составит 8 пикселов.

Теперь рассмотрим, какие возможности в плане перемещения объектов предоставляются на панели **Properties** или на панели **Info**.

1. Загрузите панель **Properties**, вызвав команду **Properties** меню **Window**, или панель **Info**, вызвав команду **Info** меню **Window**.
2. Выделите объект или группу объектов. Обратите внимание на поля **X** и **Y** в левом нижнем углу панели **Properties** и в правой части панели **Info**. В них отображаются текущие координаты выбранного объекта по вертикали (**Y**) и горизонтали (**X**). Следует отметить, что начальная точка системы координат находится в левом верхнем углу области действия.
3. Чтобы изменить положение выбранного объекта, введите требуемые значения в поля координат.

Единицы измерения для системы координат определяются в поле **Ruler Units** (Единицы разметки) диалогового окна **Document Properties** (Свойства документа). Это окно открывается в результате активизации команды **Document** меню **Modify** или нажатия клавиш **Cmd/Ctrl+J**.

Панель Align

Панель Align, которая открывается с помощью команды Align меню Window или комбинации клавиш `Cmd/Ctrl+K`, содержит множество интересных элементов (рис. 6.4).

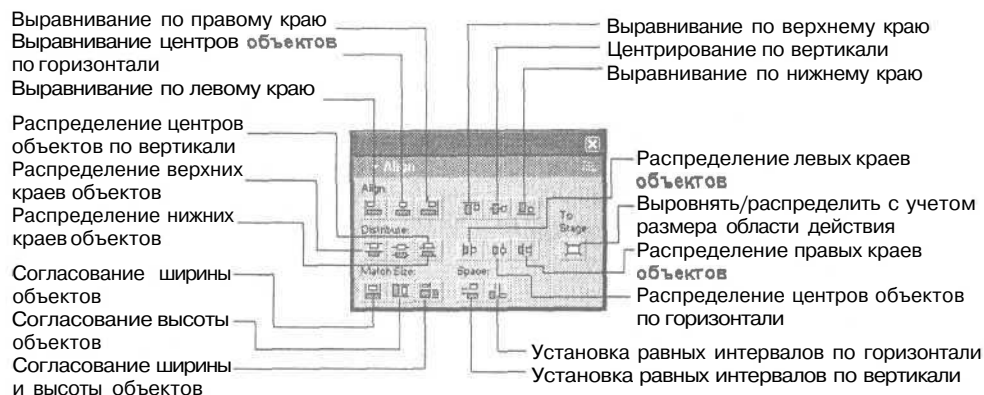


Рис. 6.4. Панель Align состоит из четырех секций: Align (Выравнивание), Distribute (Распределение), Match Size (Соответствие размера) и Space (Расстояние)

Панель Align служит не только для выравнивания объектов. Она позволяет также распределять объекты, согласовывать их размеры и устанавливать одинаковые интервалы между ними. Далее подробно описывается каждая операция.

СОВЕТ

Если вы хотите выровнять объекты относительно области действия, а не других объектов, щелкните на кнопке `Align/Distribute to Stage`, которая находится в правой части панели Align.

Выравнивание объектов

Кнопки выравнивания по горизонтали и вертикали сосредоточены в верхней части панели Align. Перечислим способы выравнивания объектов (слева направо и сверху вниз): по левым краям, горизонтально по их центральным точкам, по правым краям, по верхним краям, вертикально по центральным точкам, по нижним краям.

ПРИМЕЧАНИЕ

При выравнивании объектов Flash использует ограничивающие рамки, а не характеристики объектов.

Выровняем несколько объектов.

1. Откройте панель Align и с помощью инструмента `Arrows` выделите в области действия объекты, которые нужно выровнять. Напомним удобный способ выделения: активизировав соответствующий инструмент, нажмите левую кнопку

мыши и, удерживая ее, переместить указатель так, чтобы нужные объекты оказались внутри рамки выделения. Конечно, можно выполнить щелчок на каждом объекте при нажатой клавише Shift.

2. Выберите способ выравнивания на панели Align. На рис. 6.5 продемонстрирован один из способов *выравнивания*.

При выравнивании по левому краю Flash в качестве ориентира использует левую границу крайнего слева объекта из выбранной группы. Такой же принцип применяется при выравнивании по верхнему, нижнему и правому краям.



Рис. 6.5. Три объекта до выравнивания (слева) и после выравнивания по верхнему краю (справа)

Распределение объектов

Распределение полезно, когда необходимо равномерно расположить множество объектов в рабочей области. Секция Distribute содержит два набора кнопок, каждая из которых задает соответствующий способ распределения: по верхним краям, вертикально по центральным точкам, по нижним краям (первый набор); по левым краям, горизонтально по центральным точкам, по правым краям (второй набор).

Чтобы распределить несколько объектов, выполните следующие действия.

1. Выделите объекты (распределение двух объектов не дает видимого результата, поэтому выбирайте как минимум три объекта).
2. Активизируйте одну из кнопок распределения на панели Align. На рис. 6.6 продемонстрирован один из способов *распределения*.

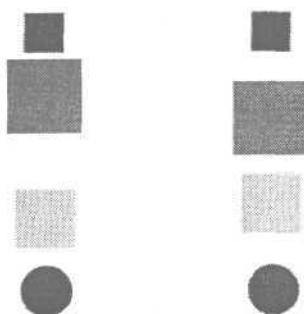


Рис. 6.6. Объекты до распределения (слева) и после распределения по центральным точкам и по вертикали (справа)

Согласование размеров объектов

Суть данной операции состоит в том, что выбранным объектам присваиваются длина и высота (либо один из указанных параметров) наибольшего объекта группы.

Согласование размеров объектов — достаточно простая операция. Вы убедитесь в этом, выполнив следующие действия.

1. Выделите два или более объектов.
2. Щелкните на одной из кнопок в секции Match Size панели Align. На рис. 6.7 продемонстрирован один из способов согласования размеров объектов.



Рис. 6.7. Два объекта до согласования размеров (слева) и после согласования размеров по высоте (справа)

Регулировка интервалов между объектами

Устанавливать равные интервалы по вертикали или по горизонтали между выбранными объектами позволяют кнопки из секции Space (Интервал) панели Align. Иногда трудно определить разницу между результатами распределения и выравнивания интервалов. Она очевидна при работе с объектами разного размера.

При использовании кнопок Space Evenly Horizontally (Установка равных интервалов по горизонтали) и Space Evenly Vertically (Установка равных интервалов по вертикали) между объектами устанавливаются равные интервалы. А в результате распределения объектов различного размера вы обнаружите, что, чем больше объект, тем меньше интервал между ним и следующим объектом. Это объясняется тем, что в процессе распределения в качестве ориентира применяется центральная опорная точка.

Давайте установим равные интервалы между объектами.

1. Выберите несколько объектов в области действия.
2. Щелкните на одной из кнопок секции Space. На рис. 6.8 показан результат использования кнопки Space Evenly Horizontally (Установка равных интервалов по горизонтали).



Рис. 6.8. Изначальное состояние трех объектов (слева) и результаты выравнивания интервалов по горизонтали (справа)

Порядок размещения объектов

Для размещения объектов широко применяются слои (см. главу 8), но, пока мы не знакомы с ними, рассмотрим, как посредством команды **Arrange** упорядочить объекты на одном слое. Какой бы способ вы ни выбрали, некоторые объекты будут располагаться позади, а некоторые — впереди. В чем тогда разница между указанными способами размещения объектов? Как вы вскоре узнаете из главы 8, слои обеспечивают больший контроль над фильмом в целом и делают обработку элементов безопасной. А при работе уже с одним слоем эффективным средством упорядочения является команда **Arrange**.

Подменю **Arrange** находится в меню **Modify**. Его команды позволяют изменить позицию любого выбранного объекта в стеке (сместить его вперед или назад относительно других объектов). Команды подменю **Arrange** просты в применении. Опишем вкратце каждую из них.

Bring to Front (На передний план). Размещение выделенного объекта поверх остальных объектов данного слоя.

Bring Forward (Переместить вперед). Перемещение выделенного объекта или группы на один уровень вперед относительно других объектов.

Send Backward (На задний план). Размещение выделенного объекта позади всех объектов данного слоя.

Send to back (Переместить назад). Перемещение выделенного объекта или группы на один уровень назад относительно других объектов.

Lock (Блокировка). Блокировка положения всех объектов в стеке.

Unlock All (Разблокировка). Отмена блокировки стека текущего слоя.

Трансформация объектов

В предыдущих версиях **Flash** все инструменты, с помощью которых можно было манипулировать формой, размером и ориентацией объектов, выводились в секции **Options** панели **Tools** при выборе инструмента **Arrow**. В **Flash MX** появился независимый инструмент, **Free Transform** (Трансформация), позволяющий манипулировать всеми перечисленными параметрами объекта.

Инструмент **Free Transform** расположен в секции **Tools**. В секции **Options** для него выводится ряд дополнительных средств. Вы можете использовать инструмент **Free Transform** в сочетании с ними или независимо. Рассмотрим второй случай. Убедитесь, что вы активизировали инструмент **Free Transform**, и щелкните на объекте, который подлежит трансформации. Появится рамка выделения и различные маркеры. Перемещайте указатель от одного маркера к другому. Его вид будет меняться в зависимости от того, для каких преобразований предназначен данный маркер.

Средство, близкое по назначению инструменту **Free Transform**, — панель **Transform**, которая открывается вследствие вызова команды **Transform** меню **Window**.

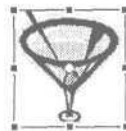
В следующих разделах вы узнаете, как выполнять масштабирование, поворот, отражение, изгиб, искажение и редактирование конверта выделенного объекта.

Масштабирование

Масштабирование заключается в изменении размера объекта по горизонтали, вертикали или во всех направлениях. Для масштабирования можно использовать либо инструмент Free Transform, либо панель Transform.

Выполним масштабирование объекта с помощью инструмента Free Transform.

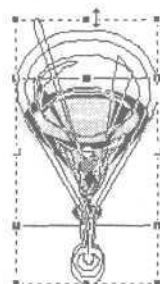
1. Выделите объект посредством инструмента Arrow и активизируйте инструмент Free Transform на панели Tools. Выбрать объект можно и с помощью инструмента Free Transform: для этого достаточно двойного щелчка на объекте.
2. Вокруг объекта появится рамка выделения с маркерами.
3. Щелкните на кнопке Scale (Масштабирование), которая расположена в секции Options (доступна, если активен инструмент Free Transform). Альтернативное средство — команда Modify ▶ Transform ▶ Scale.



ПРИМЕЧАНИЕ

Помните, какими специфическими свойствами обладает инструмент Free Transform? Вы можете пропустить шаг 3 и сразу перейти к шагу 4.

4. Для масштабирования объекта по горизонтали щелкните на одном из маркеров, расположенных на правой или левой стороне рамки выделения и, удерживая кнопку мыши, растяните рамку до требуемого размера.
5. Чтобы отмасштабировать объект по вертикали, выполните шаг 5, но используйте маркеры, находящиеся на верхней или нижней стороне рамки выделения.
6. Если необходимо масштабирование как по вертикали, так и по горизонтали, воспользуйтесь одним из угловых маркеров. Чтобы сохранить пропорции объекта, удерживайте нажатой клавишу Shift.
7. По завершении масштабирования щелкните в любом месте рабочей области за пределами объекта, чтобы скрыть маркеры.



А вот как в целях масштабирования применяется панель Transform:

1. Выделите объект, размеры которого вы хотите изменить.
2. Откройте панель Transform, активизировав команду Transform меню Window.
3. Чтобы отмасштабировать объект по горизонтали, введите значение (в процентах) в поле Width (Ширина).
4. Для масштабирования объекта по вертикали введите значение (в процентах) в поле Height (Высота).

ПРИМЕЧАНИЕ

Диапазон допустимых значений в полях Width (Ширина) и Height (Высота) — от 1 до 1000.

5. Если требуется масштабирование как по вертикали, так и по горизонтали, активизируйте флажок **Constrain** (Сохранять пропорции) и введите значение (в процентах) в любое из полей.

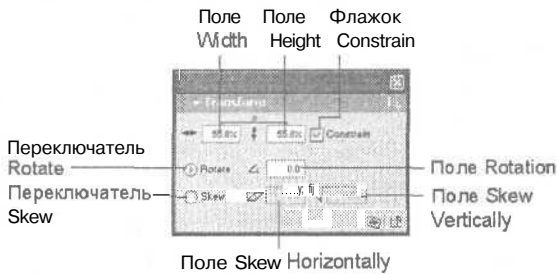


Рис. 6.9. Панель Transform

6. В завершение щелкните в любом месте рабочей области за пределами объекта, чтобы скрыть маркеры.

Вращение

В процессе вращения объект поворачивается вокруг своей точки регистрации. Подробно о точках регистрации рассказывается в разделе «Работа с точками трансформации». На данном этапе достаточно знать, что эти точки действуют как опорные при вращении.

Вращение объекта можно производить с помощью как инструмента **Free Transform**, так и панели **Transform**. Чтобы повернуть объект посредством инструмента **Free Transform**, выполните следующие действия.

1. Выделите объект с помощью инструмента **Aggow** и активизируйте инструмент **Free Transform**. Напомним, что выбрать объект можно и с помощью инструмента **Free Transform**. Для этого достаточно двойного щелчка на объекте.
2. Вокруг объекта появится рамка выделения с маркерами.
3. Щелкните на кнопке **Rotate and Skew** (Вращение и наклон), которая расположена в секции **Options** (доступна, если рабочим является инструмент **Free Transform**). Альтернативное средство — команда **Modify** ▶ **Transform** ▶ **Rotate and Skew**.

ПРИМЕЧАНИЕ

Благодаря специфическим свойствам инструмента **Free Transform** вы можете пропустить шаг 3 и сразу перейти к шагу 4.

4. Установите указатель мыши около одного из *угловых* маркеров, вследствие чего он приобретет вид круговой стрелки.
5. Нажмите левую кнопку мыши и, удерживая ее, переместите указатель так, чтобы объект *повернулся*.
6. В завершение щелкните в любом месте рабочей области за пределами объекта, чтобы скрыть маркеры.



Далее описано, как осуществлять вращение с помощью панели Transform.

1. Выделите объект, который нужно повернуть.
2. Откройте панель Transform (рис. 6.9), вызвав команду Transform меню Window.
3. Активизируйте переключатель Rotate (Вращение).
4. Введите значение (в градусах) в поле Rotation (Вращение).
5. Нажмите клавишу Return/Enter, чтобы повернуть объект на указанный угол.

СОВЕТ

Если шаг вращения должен составлять 90°, используйте команды Modify ▶ Transform ▶ Rotate 90° CW и Modify > Transform ▶ Rotate 90° CCW. Первая выполняет вращение по часовой стрелке, а вторая — против часовой стрелки.

Зеркальное отражение объекта

Flash позволяет отражать объект по горизонтали или вертикали без изменения его относительной позиции в области действия.

1. Выделите объект с помощью инструмента Arrow.
2. Чтобы получить отражение объекта по горизонтали, активизируйте команду Modify ▶ Transform ▶ Flip Horizontal. Если операция должна осуществляться относительно вертикальной оси, вызовите команду Modify ▶ Transform ▶ Flip Vertical. На рис. 6.10 показан результат выполнения команд Flip Horizontal и Flip Vertical.



Рис. 6.10. Исходные объекты (слева), объекты после отражения по горизонтали (в центре) и после отражения по вертикали (справа)

Наклон

Объект может быть наклонен в двух направлениях: по вертикали или горизонтали. Как и в предыдущих случаях, существует два способа выполнения данной операции.

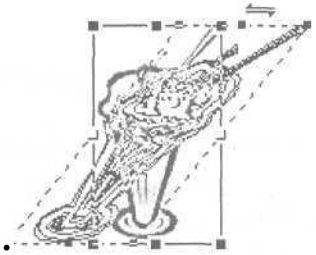
Сначала попробуем наклонить объект с помощью инструмента Free Transform:

1. Выделите объект посредством инструмента Arrow или Free Transform (вы помните, что во втором случае на объекте надлежит выполнить двойной щелчок).
2. Щелкните на кнопке Rotate and Skew (Вращение и наклон), которая расположена в секции Options (данная кнопка доступна, если выбран инструмент Free Transform).

ПРИМЕЧАНИЕ

Еще раз отметим, что благодаря специфическим свойствам инструмента Free Transform вы можете пропустить шаг 2 и сразу перейти к шагу 4.

3. Когда появится рамка выделения с маркерами, установите указатель около одного из *боковых* маркеров. Указатель мыши приобретет вид двух разнонаправленных стрелок, вертикальных или горизонтальных, что зависит от того, на какой стороне рамки выбран маркер.
4. Нажмите левую кнопку манипулятора и, удерживая ее, перемещайте маркер, пока не будет получен требуемый размер.
5. В завершение щелкните в любом месте рабочей области за пределами объекта, чтобы скрыть маркеры.



Наклон объекта можно также выполнить с помощью панели Transform:

1. Выделите объект, который нужно наклонить.
2. Откройте панель Transform, вызвав команду Transform меню Window.
3. Активизируйте переключатель Skew (Наклон).
4. Чтобы наклонить объект по вертикали, введите значение (в градусах) в поле Skew Vertically.
5. Чтобы наклонить объект по горизонтали, введите значение (в градусах) в поле Skew Horizontally.
6. Если объект должен быть наклонен на строго определенный угол, нажмите клавишу Return/Enter.

Искажение

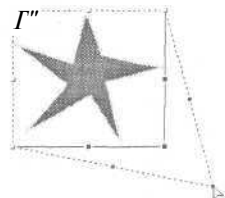
В процессе искажения объекта вы можете изменить позицию любого маркера, углового или центрального. В результате не только изменится *положение* выбранного маркера (а следовательно, и форма объекта), но и *переменятся* близкие к маркеру края рамки.

Исказить можно *несколько* отдельно выделенных объектов, но не объекты, сгруппированные посредством команды Modify ▶ Group.

Чтобы получить коническое искажение, следует удерживать нажатой клавишу Shift.

В отличие от остальных форм трансформации, искажение можно осуществлять только с помощью инструмента Free Transform.

1. Выделите объект с помощью инструмента Arrow.
2. Активизируйте инструмент Free Transform.
3. Щелкните на кнопке Distort (Искажение) в секции Options панели инструментов Tools (данная кнопка доступна, если рабочим является инструмент Free Transform). Альтернативное средство — команда Modify ▶ Transform ▶ Distort.
4. Когда появится рамка выделения, установите указатель мыши рядом с одним из маркеров. В результате вид указателя изменится: он превратится в большую белую стрелку.



5. Нажмите левую кнопку мыши и, удерживая ее, переместите маркер в требуемую позицию. Отпустите кнопку мыши. Объект будет искажен в соответствии с измененной рамкой выделения.

Манипулирование конвертом объекта

МХ *Конверт* — это рамка, которая появляется вокруг выделенного объекта, не входящего в группу. Для редактирования конверта (а следовательно, и трансформации самого объекта) предназначен инструмент Edit Envelope (Правка конверта), который является дополнительным средством инструмента Free Transform.

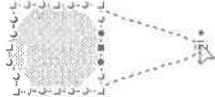
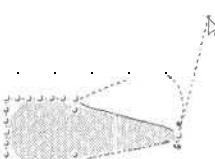
Действие инструмента Edit Envelope не распространяется на сгруппированные объекты, растровые изображения, градиенты и текст. Однако программа позволяет отредактировать конверт нескольких отдельно выделенных объектов.

Рассмотрим, как манипулировать конвертом.

1. С помощью инструмента Arrow выберите объект, конверт которого требует обработки.
2. Активизируйте инструмент Free Transform.
3. Щелкните на кнопке Envelope (Конверт) в секции Options (данная кнопка доступна, если выбран инструмент Free Transform). Альтернативное средство — команда Modify ▶ Transform ▶ Envelope.

ПРИМЕЧАНИЕ

Помните: если объект входит в группу, кнопка Envelope недоступна.

4. Когда появится рамка выделения, установите указатель мыши рядом с одним из маркеров. В результате вид указателя изменится: он превратится в большую белую стрелку. 
5. Нажмите левую кнопку мыши и, удерживая ее, переместите маркер в требуемую позицию. Отпустите кнопку манипулятора. Объект будет искажен в соответствии с новой формой рамки выделения.
6. Обратите внимание на то, что при перемещении одного из маркеров конверта появляются управляющие точки. Как и в случае работы с инструментом Реп (Перо), вы можете манипулировать ими для изменения формы конверта. 
7. Придав объекту подходящую форму, щелкните в любом месте рабочей области.

Восстановление трансформированных объектов

Было бы крайне неудобно, если бы программа не позволяла восстановить исходное состояние трансформированных объектов. К счастью, такая возможность существует.

1. Выделите объект, которому необходимо вернуть исходное состояние.

- Активизируйте команду **Modify** ▶ **Transform** ▶ **Remove Transform** или нажмите кнопку **Reset** (Возврат в исходное положение), расположенную в правом нижнем углу панели **Transform**.

Работа с точками трансформации

МХ *Точка регистрации* — это опорная точка, которую Flash использует при трансформации объекта. Во время вращения объект поворачивается вокруг точки регистрации. При выравнивании или распределении отсчет ведется именно от точек регистрации.

К сожалению, вы не можете непосредственно манипулировать точкой регистрации. Однако у вас есть возможность изменить положение *точки трансформации*, которую следует рассматривать как временную точку регистрации, появляющуюся в процессе трансформации. Благодаря точке трансформации вы можете более эффективно управлять процессами, например повернуть объект относительно не центра, а правого верхнего угла. Для этого достаточно перед вращением переместить точку трансформации из центра в правый верхний угол.

ПРИМЕЧАНИЕ

После перемещения новая позиция точки трансформации является временной. Как только программа преобразует объект и вы выполните щелчок в области действия, она займет свое прежнее положение в центре объекта.

Давайте рассмотрим, как перемещать точку трансформации объекта:

- Выделите объект.
- Задайте любую трансформацию посредством инструмента **Free Transform** либо команды **Modify** ▶ **Transform**. Вокруг объекта появится рамка выделения с маркерами.
- Щелкните на точке трансформации, которая находится в центре объекта и имеет вид белого кружочка. Удерживая кнопку **Shift** мыши, переместите точку в требуемую позицию.
- После перемещения отпустите кнопку манипулятора и трансформируйте объект.



СОВЕТ

Чтобы установить точку трансформации в ее начальное положение в режиме трансформации, выполните на ней двойной щелчок.

Разработки

В качестве примера разработки для данной главы мы выбрали Web-узел компании **Atomic Cartoons** (www.atomiccartoons.com), где наглядно демонстрируются возможности управления объектами.

Канадская компания **Atomic Cartoons** специализируется на создании красочных и веселых Flash-фильмов для распространения в Web. Адресованный широкой

аудитории, Web-узел компании Atomic Cartoons может служить примером прекрасного графического дизайна, в котором ярко проявился талант разработчиков (рис. 6.11).

Прекрасно сконструированные графические элементы свидетельствуют о профессиональном владении всеми функциями и инструментами, предназначенными для манипулирования объектами в среде Flash.

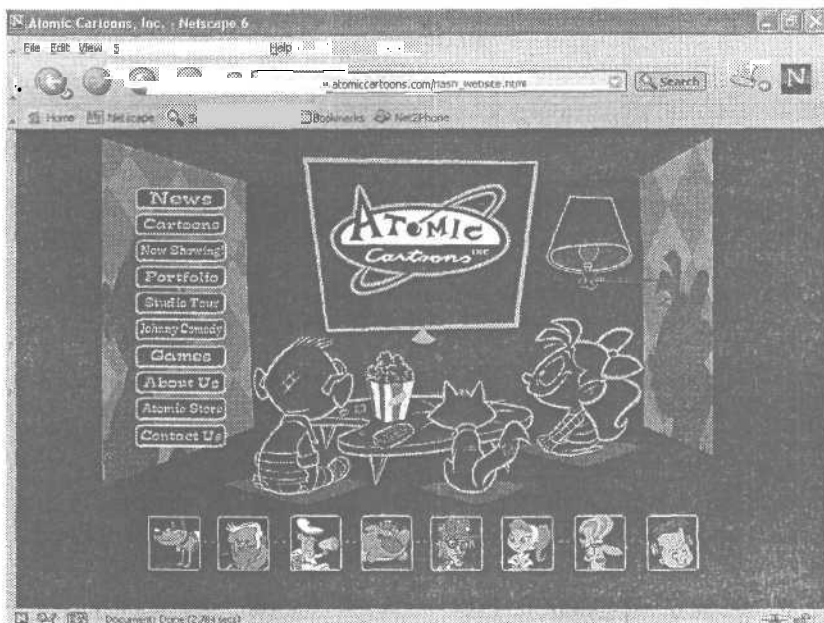


Рис. 6.11. Web-узел компании Atomic Cartoons

Резюме

Глава посвящена вопросам управления внешним видом объектов в среде Flash. Поскольку от умения модифицировать объекты зависит успех Flash-проектов, в главе подробно описаны все способы упорядочения и трансформации объектов: группирование, выравнивание, распределение, масштабирование, вращение, отражение, наклон, искажение и редактирование конверта. Перечисленные процессы не являются необратимыми, программа Flash позволяет быстро восстанавливать первоначальный вид объектов.

Глава 7

Объекты многократного использования

Flash-фильмы часто содержат элементы, имеющие сходные характеристики. Независимая разработка каждого такого элемента связана с неоправданными временными потерями. В подобных случаях целесообразно применять *символы*. Благодаря им вы сможете использовать одно изображение в качестве основы при создании остальных объектов. Ваша работа будет сведена к выполнению специфических изменений в каждом новом элементе. Таким образом, символы позволяют значительно ускорить процесс подготовки изображений.

Цель данной главы — познакомить вас с символами и другими видами объектов многократного использования среды Flash, а также рассказать об инструментах, предназначенных для управления ими.

В этой главе рассматриваются следующие темы:

- ▶ символы и их типы;
- ▶ создание символов;
- ▶ преобразование изображений в символы;
- ▶ импортирование символов;
- ▶ использование библиотек;
- ▶ работа с экземплярами символов;
- ▶ панель Movie Explorer.

Символы

Символы представляют собой графические объекты, которые хранятся в библиотеке. Важнейшее достоинство символов состоит в возможности их многократного применения. Вы можете использовать символы неограниченное число раз как в одном, так и в нескольких Flash-фильмах. После создания символ автоматически добавляется в библиотеку. При извлечении из библиотеки в область действия помещается не символ, а его копия, которую называют *экземпляром*. Вы можете как угодно модифицировать экземпляры, но исходный символ в библиотеке останется неизменным.

ПРИМЕЧАНИЕ

Любая сложная анимация может быть создана только с использованием символов. Следовательно, большинство ваших графических элементов (или даже все) следует преобразовать в символы или изначально создавать как символы. Пусть пока вас не занимают вопросы анимирования символов с помощью временной шкалы — это одна из тем части 3 нашей книги. На данном этапе важно понять различия между основными видами символов.

Символы делятся на две категории: Flash-символы, создаваемые непосредственно в этой среде, и импортированные символы, подготовленные в других программах.

Символы первой категории называют также *символьными характеристиками*. Такое название выбрано потому, что в Flash существуют три разных типа собственных символов, каждый из которых имеет свою отличительную характеристику и применяется для решения определенных задач в фильме.

Символы удобны по целому ряду причин. Во-первых, как уже говорилось, при их использовании значительно ускоряется процесс создания фильма: ведь для получения различных объектов достаточно модифицировать экземпляр символа, хранящегося в библиотеке. Во-вторых, в случае применения символов значительно уменьшается размер файла Flash-фильма. Дело в том, что Flash сохраняет только информацию о различиях между используемыми экземплярами символа. А вот если бы вы создавали для каждого объекта отдельный символ, тогда бы файл содержал все эти символы, что привело бы к увеличению его размера.

Собственные символы Flash

Давайте познакомимся поближе с основными типами Flash-символов.

Графические символы

Графические символы — это статические графические объекты, созданные с помощью различных инструментов рисования и черчения в среде Flash. Они наиболее часто используются в качестве элементов, анимируемых с помощью временной шкалы.

Кнопки

Кнопки представляют собой динамические элементы, которые способны реагировать на щелчок мышью. Они принадлежат к числу наиболее популярных интерактивных элементов в среде Flash. Создавать кнопки вы научитесь в главе 13. На данном этапе достаточно знать, что кнопки формируются из четырех различных статических изображений, называемых *состояниями*. В зависимости от действий пользователя, отображается то или иное состояние. Всего насчитывается четыре состояния. Три из них можно наблюдать на экране: кнопка отжата (состояние up), над кнопкой находится указатель мыши (состояние over), на кнопке выполнен щелчок (состояние down). Четвертое состояние, hit, визуально проследить нельзя, оно необходимо для определения активной части кнопки. Замечательным свойством кнопок является возможность их тесной интеграции с ActionScript, что особенно важно при создании интерактивных фильмов.

Клипы

Клипы — это небольшие фрагменты фильма, доступные для воспроизведения в других фильмах. Один клип может быть вложен в другой. Допускается любое число уровней вложения.

Клипы, элементы с произвольной структурой, функционируют *независимо* от временной шкалы. Их также можно вкладывать в другие символы. Таким образом, вы можете вставить клип в одно из состояний кнопки, тем самым создав анимированную кнопку.

Подробно о создании клипов и манипуляциях с ними рассказывается в главе 10.

Импортированные Flash-символы

Ознакомимся теперь с типами импортируемых символов.

Растровые рисунки

Из предыдущих глав вы уже знаете, как импортировать растровые изображения. Любой импортируемый рисунок автоматически помещается в библиотеку, конвертируется в символ и помещаются в область действия.

Звуковые файлы

Звуковые файлы в процессе импортирования помещаются только в библиотеку. Поскольку обсуждение звуковых эффектов на данном этапе равносильно открытию «ящика Пандоры» (из этого ящика, подаренного Зевсом Пандоре, беды и болезни распространились по Земле), мы перенесем данный разговор на более поздний срок.

Детальная информация о звуковых эффектах приведена в части 6 данной книги.

Цифровое видео

Flash позволяет импортировать и использовать файлы различных цифровых видеоформатов. Если на вашем компьютере установлена программа QuickTime 4 (или ее более поздняя версия), вы можете импортировать файлы форматов QuickTime, Audio Video Interlace (AVI), Digital Video (DV) и Motion Picture Expert Group (MPEG). Разрешается также импортировать только звуковые файлы QuickTime и помещать их в библиотеку как символы.

Если же вы работаете в системе Windows, где инсталлирована программа DirectX 7 (или ее более поздняя версия), у вас есть возможность импортировать файлы форматов AVI, MPEG и Window Media Files (WMF/ASF). Все импортированные видеofайлы автоматически помещаются в библиотеку для использования в качестве символов.

Создание символов

Научившись создавать символы и манипулировать ими, вы откроете для себя мир замечательных возможностей Flash. В этом разделе мы рассмотрим два основных способа образования графических символов: создание новых символов с predeterminedными параметрами и преобразование существующих изображений в символы.

Кнопки и клипы будут обсуждаться в последующих главах. Данный раздел посвящен исключительно созданию графических символов.

Создание графических символов с predeterminedными параметрами

При создании графических символов основные операции выполняются с помощью инструментов рисования и черчения программы Flash.

1. Убедитесь, что в области действия отсутствуют *выделенные* объекты.

- Активизируйте команду **New Symbol** меню **Insert** или нажмите клавиши **Cmd/Ctrl+F8**. Откроется диалоговое окно **Create New Symbol** (Создание нового символа).

ПРИМЕЧАНИЕ

Вы можете переключать режимы отображения основных и дополнительных параметров в окне **Create New Symbol** посредством кнопки **Basic/Advanced**. Параметры, отображаемые в расширенном режиме, используются при создании совместно используемой библиотеки (**Shared Library**). Данный вопрос обсуждается в разделе «Совместно используемые библиотеки».

- Когда появится диалоговое окно **Create New Symbol**, убедитесь, что установлен переключатель **Graphic**, введите имя символа в поле **Name** и щелкните на кнопке **OK**.



СОВЕТ

Если в фильме используется большое количество графических изображений, постарайтесь дать каждому символу уникальное информативное имя, что облегчит операцию поиска. По умолчанию Flash использует имена **Symbol X**, где **X** — порядковый номер символа.

- Символ будет помещен в библиотеку, а Flash перейдет в режим редактирования символа. Имя символа появится в строке **Scene and Symbol** справа от названия сцены, с которой в настоящий момент ведется работа. Обратите внимание на то, что в центре окна редактора **Symbol Editor** (Правка символа) появится точка регистрации символа (перекрестье).

ПРИМЕЧАНИЕ

При работе в режиме редактирования символов внешний вид среды Flash практически не меняется. Чтобы вы не запутались, отметим, что, когда активен этот режим, в строке Scene and Symbol справа от названия текущей сцены отображается имя символа, а в центре области действия появляется точка регистрации символа.

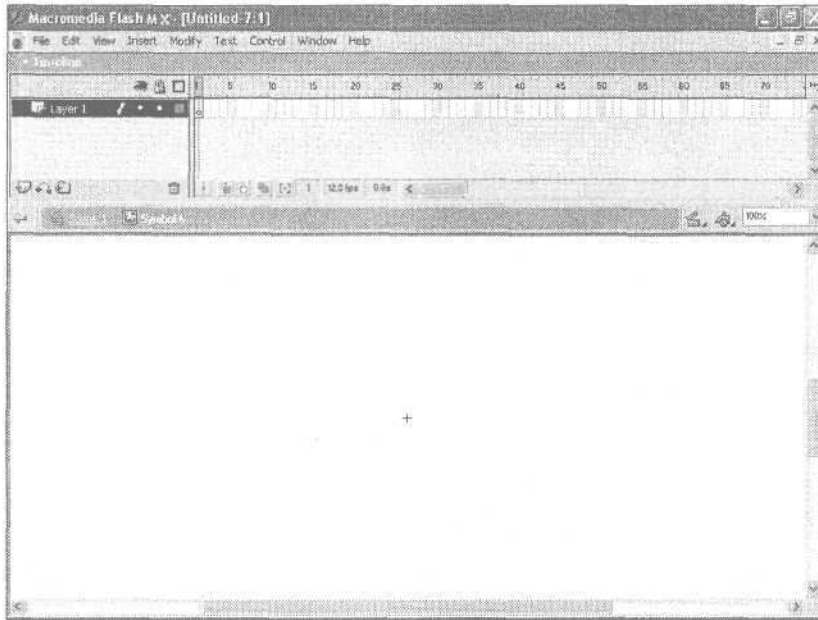


Рис. 7.1. Окно среды Flash в режиме редактирования символов

5. Используйте навыки, полученные при изучении главы 3, для создания графического символа.
6. Если символ готов, выйдите из режима редактирования. Для этого активизируйте команду Edit Document меню Edit либо щелкните на имени сцены в строке Scene and Symbol.

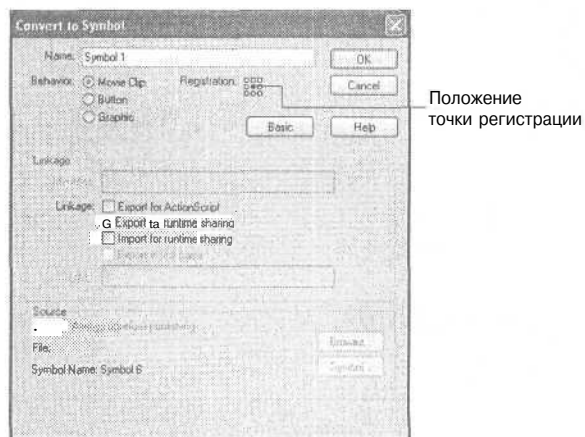
Использование сцен является способом разделения Flash-проекта на управляемые фрагменты. Эти вопросы освещаются в главе 11.

Конвертирование графических изображений в символы

Хотя многие Flash-символы создаются с помощью специального редактора, возможно, вам потребуется конвертировать в символ готовый рисунок.

1. Выберите в области действия объект, который нужно преобразовать в графический символ.
2. Активизируйте команду Convert to Symbol меню Insert.

3. Когда откроется диалоговое окно Convert to Symbol (Конвертирование в символ), убедитесь в том, что установленный переключатель соответствует требуемому типу символа.
4. Введите имя символа в поле Name.
5. Выберите положение точки регистрации. Для этого щелкните на одном из девяти небольших квадратиков соответствующей диаграммы. Каждый квадратик представляет позицию точки регистрации символа в пределах рамки выделения.



6. Нажмите кнопку ОК, вследствие чего символ автоматически будет добавлен в библиотеку.
7. Исходный объект, который вы конвертировали, превратится в экземпляр символа, оригинал которого находится в библиотеке.

Инструментальные средства библиотек

Библиотека, загружаемая посредством команды Library меню Window, — это место хранения всех символов Flash-проекта: графических символов, кнопок, клипов, импортированных символов. Кроме того, с помощью библиотек можно быстро осуществлять предварительный просмотр анимации и звуковых файлов.

Доступ к разделам библиотеки предоставляет команда Common Libraries меню Window. В библиотеке имеется три раздела предварительно созданных символов — Buttons (Кнопки), Learning Interactions (Интерактивные элементы) и Sounds (Звуки). Все эти символы вы можете использовать в своих проектах.

Библиотека снабжена средствами и инструментами, разработанными специально для управления символами (рис. 7.2). Чтобы вы научились в совершенстве оперировать ими, рассмотрим каждый инструмент отдельно.

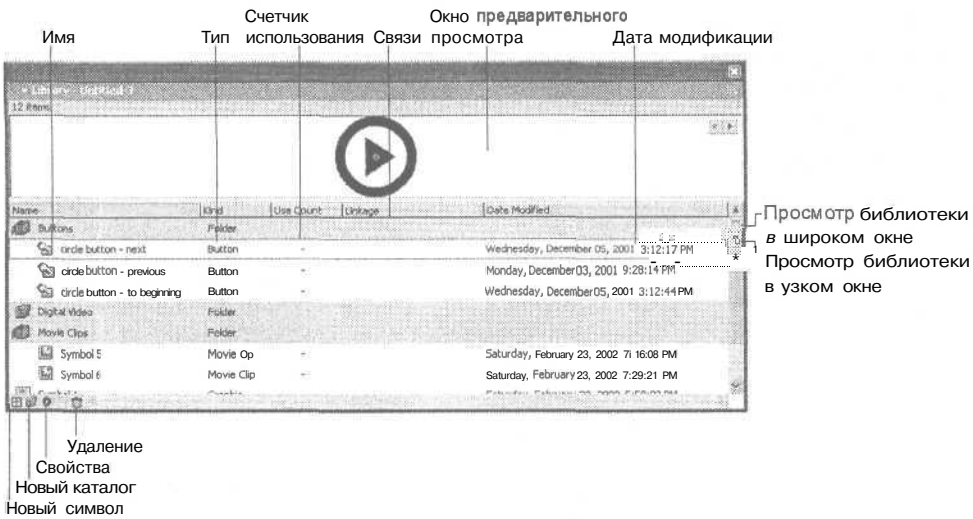


Рис. 7.2. Окно библиотеки

Столбцы в окне библиотеки

В столбцах отображается информация о символах, хранящихся в библиотеке. Чтобы сделать видимыми все столбцы, следует расширить окно библиотеки по горизонтали, перетаскив правую границу или щелкнув на кнопке Wide Library View, которая расположена в правой части панели Library.

В результате выполнения щелчка на любом параметре символа (имени, значке, связи, дате модификации и т. д.) данный символ отображается в окне предварительного просмотра.

Давайте совершим путешествие по столбцам библиотеки.

- ▶ **Name (Имя).** Этот столбец содержит имена, присвоенные символам, а также имена импортированных файлов (например, звуковых файлов или файлов растровых изображений). Сортировка имен в столбце Name производится в алфавитном порядке. Чтобы осуществить сортировку в обратном порядке, щелкните на кнопке Toggle Sorting Order, которая расположена в правой части окна библиотеки.
- ▶ **Kind (Тип).** В данном столбце представлена информация о типах символов, хранящихся в библиотеке: Button (Кнопка), Bitmap (Растровое изображение), Graphic (Графический символ), Movie Clip (Клип) или Sound (Звук). Чтобы отсортировать символы по типу, щелкните на заголовке столбца Kind. Тип символа можно определить также по небольшому значку слева от имени.
- ▶ **Use Count (Счетчик использования).** Значения в этом столбце указывают, сколько раз каждый символ был использован в фильме. Такая информация очень полезна, если вы работаете над большим проектом и хотите владеть точной статистикой употребления того или иного символа в итоговом варианте фильма.

- ▶ **Linkage (Связи).** Здесь содержится информация об использовании символа в других фильмах и о том, является ли символ импортированным из другого фильма.
- ▶ **Date Modified (Дата модификации).** В последнем столбце фиксируется дата последней модификации символа.

Панель кнопок

В нижней части окна библиотеки (рис. 7.2) находится несколько важных кнопок.

- ▶ **New Symbol (Новый символ).** Активизация данной кнопки приводит к открытию диалогового окна **Create New Symbol**, которое предназначено для создания нового символа непосредственно в библиотеке.
- ▶ **New Folder (Новый каталог).** Эта полезная кнопка позволяет создавать каталоги в библиотеке для улучшения организации ее содержимого. Чтобы добавить символ в каталог, щелкните на значке символа и перетащите его в целевой каталог. Если требуется развернуть содержимое каталога, выполните двойной щелчок на его значке.
- ▶ **Properties (Свойства).** Данная кнопка предоставляет доступ к диалоговому окну **Symbol Properties**, с помощью которого вы можете изменить свойства любого выбранного символа.
- ▶ **Delete (Удаление).** Это средство удаления символов из библиотеки. Выберите символ, который подлежит удалению, и щелкните на кнопке **Delete**.

Меню Options

Меню **Options**, кнопка открытия которого расположена в правом верхнем углу панели **Library**, содержит полный набор команд для управления библиотекой. Эти команды продублированы в контекстном меню. Чтобы открыть его, щелкните правой кнопкой мыши (**Windows**) или выполните щелчок при нажатой клавише **Ctrl** (**Macintosh**).

ПРИМЕЧАНИЕ

Команды **New Symbol** и **New Folder** обсуждались в предыдущем разделе.

Ознакомимся с каждой командой.

- ▶ **New Font (Новый шрифт).** Обзор мы начинаем с команды, позволяющей создавать шрифтовые символы типа **Font**, которые хранятся в библиотеке. Данное свойство весьма полезно, поскольку вам больше не понадобится вкладывать шрифт непосредственно в фильм. За информацией о символах типа **Font** обращайтесь к главе 5.
- ▶ **New Video (Новое видео).** В результате активизации команды, указанной в заголовке, в библиотеку добавляется пустой символ типа **Video**. Чтобы заполнить этот символ, выполните двойной щелчок на его значке. Когда откроется диалоговое окно **Embedded Video Symbol (Вложенный видео-символ)**, нажмите кнопку **Import (Импортировать)**, укажите путь к нужному видеофайлу, выберите его и щелкните на кнопке **OK**.

- ▶ **Rename (Переименовать)**. Данная команда позволяет изменить имя символа непосредственно в библиотеке. Выберите символ, который хотите переименовать, откройте меню Options, активизируйте команду Rename и ведите новое имя. Для изменения имени символа можно также выполнить на нем двойной щелчок.
- ▶ **Move to a New Folder (Переместить в новый каталог)**. Эта команда автоматически создает новый каталог и помещает в него выделенный символ.
- ▶ **Duplicate (Дублировать)**. Назначение данной команды — создание копии символа. Выберите символ, который требуется скопировать, откройте меню Options и активизируйте команду Duplicate. Когда появится диалоговое окно Duplicate Symbol (Дублировать символ), установите все необходимые параметры.
- ▶ **Delete (Удалить)**. А эта команда работает так же, как и кнопка Delete, которая находится в нижней части панели Library. Выберите символ, подлежащий удалению, откройте меню Options и активизируйте команду Delete.
- ▶ **Edit (Правка)**. Команда Edit позволяет загрузить графический символ, клип или кнопку в режиме редактирования. Здесь можно произвести любые изменения, в соответствии с которыми будет модифицирован символ в библиотеке.
- ▶ **Edit With (Правка с помощью)**. Благодаря команде Edit With у вас есть возможность открыть растровое изображение или звуковой файл во внешнем приложении. Если на вашем компьютере установлена программа Fireworks, Flash запустит Fireworks для редактирования растрового изображения. Изменения, выполненные в приложении Fireworks, не влияют на исходный файл рисунка, а касаются только символа в библиотеке среды Flash. Когда данная команда активизируется для звукового файла, появляется диалоговое окно, где необходимо указать звуковой редактор.
- ▶ **Properties (Свойства)**. В результате вызова данной команды открывается диалоговое окно Symbol Properties для выделенного объекта. В этом окне вы можете изменять тип символа и его имя, а также манипулировать свойством Linkage.
- ▶ **Linkage (Связь)**. Эта команда позволяет выполнить несколько установок, необходимых для создания совместно используемой библиотеки. Подробное описание этого процесса вы найдете в соответствующем разделе настоящей главы.
- ▶ **Component Definition (Определение компонента)**. Выбрав данную команду, вы сможете управлять настройками компонентов пользовательского интерфейса Flash MX посредством диалогового окна Component Definition (Определение компонента). Детальная информация о компонентах содержится в главе 16.
- ▶ **Select Unused Items (Выбрать неиспользуемые компоненты)**. Благодаря указанной команде обеспечивается возможность выбора символов, которые не используются в текущем проекте.
- ▶ **Update (Обновить)**. Эта полезная команда обновляет растровые изображения и звуковые файлы, которые вы редактировали во внешних программах, что позволяет не импортировать их повторно.

При редактировании изображения с помощью программы Macromedia Fireworks команду Update применять не нужно.

- ▶ Play (Воспроизведение). Назначение команды Play — воспроизведение клипов, звука и символов типа Button (Кнопка). Режим воспроизведения перечисленных элементов активизируется также посредством кнопки Play, которая расположена в верхнем правом углу окна предварительного просмотра.
- ▶ Expand Folder (Развернуть каталог). Настоящая команда служит для отображения на экране содержимого выделенного каталога.
- ▶ Collapse Folder (Свернуть каталог). С помощью команды Collapse Folder автоматически свертывается любой выбранный каталог, чтобы скрыть все его содержимое, — противоположный эффект развертыванию каталога.
- ▶ Expand All Folder (Развернуть все каталоги). Данная команда разворачивает все каталоги текущей библиотеки для просмотра их содержимого.
- ▶ Shared Library Properties (Свойства совместно-используемой библиотеки). Вызов этой команды приводит к загрузке диалогового окна Shared Library Properties (Свойства совместно-используемой библиотеки), где устанавливается связь между текущей библиотекой и URL-адресом.
- ▶ Keep Use Counts Updated (Постоянное обновление счетчиков использования символов). При вызове рассматриваемой команды осуществляется переход в режим постоянного обновления счетчиков применения символов в фильме. Следует учитывать, что работа с командой Keep Use Counts Updated сопровождается интенсивным использованием ресурсов процессора, что может значительно замедлить работу компьютера.
- ▶ Update Use Counts Now (Обновление счетчиков использования символов). Завершает обзор команда, предназначенная для обновления значений счетчиков использования символов в фильме.

Работа с библиотеками символов

В этом разделе рассказывается, как добавлять символы в область действия и манипулировать ими за счет изменения различных настроек библиотеки.

Добавление символа в область действия

Помещать символы в область действия так же легко, как и в библиотеку.

ПРИМЕЧАНИЕ —

Напомним, что в область действия добавляется экземпляр, а не сам символ. Экземпляр — это копия оригинального символа, которую вы можете изменять различными способами. Эти манипуляции никак не отразятся на исходном символе.

Добавим символ в область действия.

1. Откройте библиотеку, активизировав команду Library меню Window либо нажав клавишу F11.
2. С помощью мыши выберите символ и перетащите его в область действия.

Переименование символа

При создании символ получает стандартное имя. Существует несколько способов изменить его.

1. Убедитесь, что окно библиотеки открыто.
2. Выберите символ, который хотите переименовать.
3. Выполните одно из следующих действий:
 - ▷ Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и активизируйте команду Rename. Когда в окне библиотеки будет подсвечено текущее имя символа, введите новое название.
 - ▷ Откройте меню Options и вызовите команду Rename. Когда текущее имя будет подсвечено, введите новое название.
 - ▷ Откройте меню Options и выберите команду Properties (или щелкните на кнопке Properties в нижней части окна). На экране появится диалоговое окно Symbol Properties (Свойства символа). Введите новое имя в поле Name.
 - ▷ Выполните двойной щелчок на имени символа и измените его.

Дублирование символов

Flash позволяет дублировать символы непосредственно в библиотеке. Дублирование осуществляется в тех случаях, когда необходима копия символа, которую можно модифицировать без изменения оригинала. Рассмотрим этапы этого процесса.

1. Откройте окно библиотеки.
 2. Выделите символ, который хотите сдублировать.
 3. Выполните одно из следующих действий:
 - ▷ Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и активизируйте команду Duplicate.
 - > Откройте меню Options и вызовите команду Duplicate.
 4. В открывшемся диалоговом окне Duplicate Symbol укажите имя дубликата символа. Выберите тот переключатель, который соответствует требуемому типу символа.
 5. Щелкните на кнопке ОК. Дубликат появится в библиотеке.
- Теперь вы можете работать с дубликатом символа.

Правка символа

На практике часто приходится вносить существенные коррективы в символ. Далее описываются операции, которые могут быть связаны с этим.

1. Откройте окно библиотеки.
2. Выберите символ, подлежащий редактированию.

3. Выполните одно из следующих действий:
 - ▷ Дважды щелкните на значке символа (слева от имени символа в окне библиотеки).
 - ▷ Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и активизируйте команду Edit.
 - ▷ Откройте меню Options (Опции) и вызовите команду Edit.
 - ▷ Если редактируемый символ является растровым изображением или звуковым файлом, воспользуйтесь командой Edit With, чтобы запустить соответствующую внешнюю программу.
4. Переключившись в режим редактирования символа, внесите необходимые изменения.
5. По окончании правки выйдите из режима редактирования символа. Для этого либо активизируйте команду Edit Document меню Edit, либо нажмите кнопку Scene на строке Scene and Symbol.

Организация библиотеки

По мере приобретения опыта работы с Flash ваши проекты будут становиться все более сложными. Соответственно, возрастет и количество используемых символов. Это чревато возникновением проблем, связанных с поиском необходимых компонентов. Выполнив некоторые действия организационного характера, вы предотвратите многие из них.

Одно из решений — создание в библиотеке нескольких каталогов, каждый из которых содержит символы определенного типа (рис. 7.3). Вы можете также создать каталоги в библиотеке среды Flash, где будут отдельно храниться символы, растровые изображения и звуковые файлы.

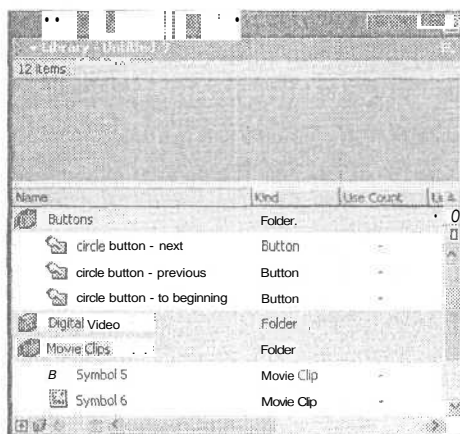


Рис. 7.3. Каталоги, каждый из которых содержит символы определенного типа

Чтобы создать каталог, щелкните на кнопке New Folder, которая находится в левом нижнем углу панели Library или активизируйте команду New Folder меню

Options. Поместить символ в соответствующий каталог несложно, достаточно перетащить его на значок каталога. Существует еще один способ: выберите символ, откройте меню Options и вызовите команду Move to New Folder.

Еще один способ организации, несомненно, меньше связан с изменением общей структуры библиотеки, но, тем не менее, является полезным. Он состоит в использовании столбцов окна библиотеки. Порядок сортировки символов можно изменить, щелкнув на заголовке одного из столбцов, например Date Modified (Дата модификации) или Name (Имя). В результате символы будут упорядочены по дате их последней модификации или в алфавитном порядке.

Доступ к внешним библиотекам

Приложение Flash не ограничивает разработчика ресурсами только той библиотеки, которая используется в текущем проекте. Команда Open as Library предоставляет доступ к библиотекам других фильмов.

ПРИМЕЧАНИЕ —

Во Flash не существует специальных внешних библиотек и соответственно команд для их создания. Библиотеки хранятся в FLA-файлах с фильмами.

Выполните следующие действия.

1. Откройте окно библиотеки.
2. Активизируйте команду Open as Library меню File.
3. В диалоговом окне Open as Library (Открыть как библиотеку) найдите ту библиотеку, к которой необходим доступ, и загрузите ее (рис. 7.4).

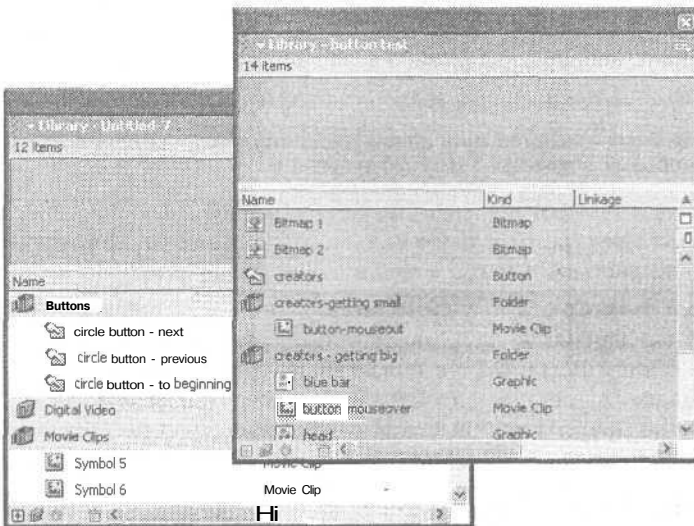


Рис. 7.4. Окно библиотеки текущего проекта (слева) и окно внешней библиотеки (справа)

ВНИМАНИЕ

Если вы не перемещали окно библиотеки вашего проекта, окно *новой* библиотеки откроется поверх него. Разместите окна так, чтобы видеть содержимое обеих библиотек.

- Щелкните на значке символа и перетащите его из окна *внешней* библиотеки в окно библиотеки текущего проекта. При этом автоматически будет создана копия символа.

Совместно используемые библиотеки

Команды разработчиков Flash-фильмов с каждым днем все многочисленнее, а проекты все сложнее. Поэтому вопросы совместного использования ресурсов важны как никогда ранее.

Имея возможность обращаться к совместно используемой (общей) библиотеке символов, или, другими словами, к библиотечному фонду, вы сможете оперировать в своих фильмах многочисленными графическими изображениями, кнопками, клипами, звуковыми файлами и другими элементами. Для получения доступа к этому *ценному* источнику *необходимо* установить связь с библиотекой центрального SWF-файла (который может храниться, например, на сервере). В результате формирование *отдельной* библиотеки для Flash-фильма окажется обязательным.

Одно из главных достоинств совместно используемой библиотеки состоит в том, что команда разработчиков сможет применять набор стандартных символов во всех своих фильмах. Кроме того, в библиотечном фонде разрешается производить окончательную модификацию символа. После публикации общей библиотеки все фильмы, в которых задействованы ее элементы, автоматически обновляются. Важно также то, что компоненты совместно используемой библиотеки не вкладываются в фильм, следовательно, размер файла фильма остается небольшим.

СОВЕТ

Мы рекомендуем использовать общие библиотеки исключительно для хранения больших (по объему занимаемой памяти) элементов.

Прежде чем мы приступим к изучению методов создания совместно используемых библиотек, рассмотрим две модели таких библиотек.

Первая модель — *runtime*. В случае использования такой модели символы загружаются в фильм-адресат из фильма-источника при воспроизведении фильма-адресата. Хотя фильм-источник не обязательно должен быть доступен во время создания фильма-адресата, его необходимо разместить на Web-узле для извлечения символов во время воспроизведения фильма-адресата. Такая модель общей библиотеки применяется при распространении Flash-фильмов в Web.

Вторая модель — *author-time*. В случае ее использования вы можете заменить любой символ в фильме, с которым работаете в данный момент, символом из другого фильма. Хотя имя и свойства символа в фильме-адресате сохраняются, его содержимое заменяется. Применение модели *author-time* целесообразно, например, если требуется интегрировать в фильм символы, созданные другим разработчиком.

Основное различие между моделями runtime и author-time состоит в том, что во втором случае фильм-источник должен быть доступен в вашей локальной сети (либо на вашем жестком диске). Поскольку вы извлекаете символы из совместно используемой библиотеки во время создания фильма, модель author-time может использоваться для публикации фильмов как в Web, так и на других носителях.

Создание общей библиотеки модели runtime

Формирование общей библиотеки модели runtime осуществляется в два этапа. На первом этапе необходимо создать фильм-источник (где будет находиться совместно используемая библиотека) и символы с уникальными именами-идентификаторами. Присвоение уникальных имен символам в совместно используемой библиотеке — очень важный процесс, поскольку эти имена применяются в фильме-адресате при загрузке символов во время воспроизведения фильма.

СОВЕТ

Рекомендуем в качестве идентификаторов использовать фактические имена символов — это поможет избежать конфликтов имен в больших проектах.

На втором этапе, когда создается фильм-адресат, программе Flash предоставляется информация о том, что используемые в фильме символы будут загружаться из фильма-источника во время воспроизведения.

Давайте рассмотрим каждый шаг описываемой процедуры.

1. Создайте новый Flash-файл, вызвав команду New меню File или нажав клавиши `Cmd/Ctrl+N`.
2. Создайте или импортируйте все элементы, которые хотите включить в совместно используемую библиотеку модели runtime, и добавьте их в библиотеку текущего проекта.
3. Если вы желаете включить в общую библиотеку шрифт, откройте меню Options и активизируйте команду New Font. В диалоговом окне Font Symbol Properties (Свойства шрифтового символа) укажите добавляемый шрифт в поле Font и, при необходимости, стиль.

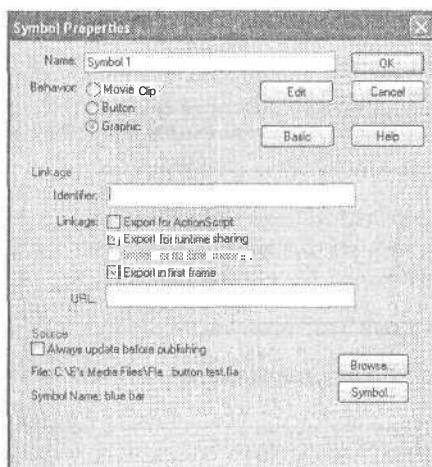
Итак, на данном этапе в нашем распоряжении имеются все символы, которые будут храниться в конечном варианте совместно используемой библиотеки. Теперь необходимо присвоить им уникальные имена. Это позволит избежать конфликтов при использовании символов в фильме-адресате.

1. Выберите любой символ в библиотеке.
2. Выполните одно из следующих действий:
 - > Активизируйте команду Linkage меню Options либо контекстного меню.
 - ▷ Активизируйте команду Properties меню Options либо контекстного меню. Когда появится диалоговое окно Symbol Properties (Свойства символа), щелкните на кнопке Advanced для получения доступа к параметрам совместно используемой библиотеки.

ПРИМЕЧАНИЕ

Если выделенный элемент не является Flash-символом (кнопкой, клипом или графическим символом), команда Properties недоступна.

3. В диалоговом окне Linkage Properties (Свойства связи) установите флажок Export for Runtime Sharing.
4. Введите уникальное имя (без пробелов) в поле Identifier.
5. В поле URL укажите адрес фильма-источника.
6. Повторите описанные операции для всех символов, которые должны быть добавлены в общую библиотеку.
7. Щелкните на кнопке ОК.

**СОВЕТ**

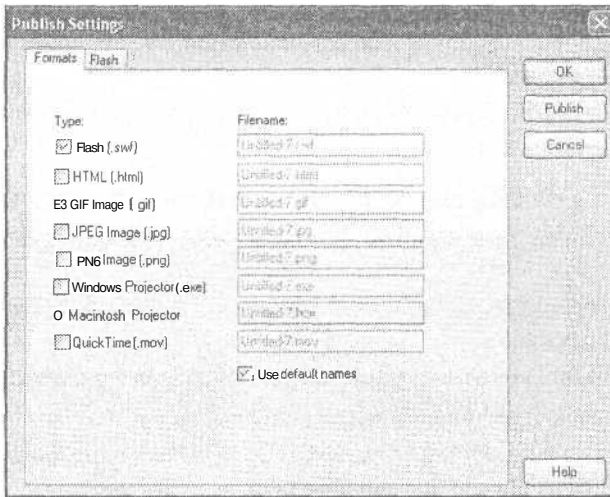
В специальное поле вводить нужно полный URL-адрес (<http://someURL.com/movies/my-movie.swf>), а не `/movies/mymovie.swf`). Если по указанному адресу вы не разместите фильм-источник, символы, необходимые для фильма-адресата, не будут обнаружены и загружены.

Процесс присвоения уникального идентификатора каждому символу довольно утомителен и в то же время крайне необходим. Символ без идентификатора не сможет функционировать корректно в конечном варианте совместно используемой библиотеки.

Все свойства фильма-источника, который будет действовать как совместно используемая библиотека модели runtime, установлены. Сохраните и опубликуйте его, выполнив следующие действия.

1. Сохраните текущий Flash-файл (включающий библиотеку, с которой вы работали) посредством команды Save меню File (ее клавиатурный аналог — `Cmd/Ctrl+S`). Надлежит присвоить файлу информативное имя, которое бы подчеркивало, что данный файл содержит совместно используемую библиотеку модели runtime.

- Опубликуйте FLA-файл как SWF-файл (Web-формат для фильмов Flash). Для этого активизируйте команду Publish Settings меню File. Убедитесь, что на вкладке Formats выбран формат Flash. Никаких изменений более не требуется.
- Щелкните на кнопке Publish и вызовите команду Publish меню File или нажмите клавиши Shift+F2. SWF-файл будет сохранен в том каталоге, где вы разместили исходный файл с расширением FLA.
- URL-адрес файла с расширением SWF, содержащего символы с идентификаторами, должен соответствовать введенному вами в поле URL.



Процесс публикации Flash-фильмов подробно рассматривается в главе 27.

Связь с общей библиотекой модели runtime

Элементы созданной общей библиотеки символов модели runtime теперь *можно* использовать в различных Flash-фильмах.

Связь с общей библиотекой модели runtime устанавливается практически так же, как и с внешней Flash-библиотекой. Основное различие состоит в том, что при *изменении* символа в совместно используемой библиотеке модели runtime (в фильме-источнике) изменяются все связанные с ним символы в фильмах-адресатах.

ПРИМЕЧАНИЕ-

В библиотеках фильмов-адресатов надлежит создавать копии символов, поступивших из общей библиотеки модели runtime. Только при соблюдении этого условия вы сможете установить связь с библиотекой. Какой же смысл в этой операции? Дело в том, что символ, который связан с совместно используемой библиотекой модели runtime, фактически играет роль заполнителя в вашем фильме.

Связь с библиотекой устанавливается следующим образом.

- Создайте *новый* Flash-фильм или загрузите уже существующий. Это -- фильм-адресат.

2. Откройте окно библиотеки фильма посредством команды Library меню Window.
3. Выберите графический символ, клип или кнопку и активизируйте команду Properties меню Options, чтобы отобразить диалоговое окно Symbol Properties (Свойства символа). Расширьте это окно, щелкнув на кнопке Advanced.
4. Установите флажок Import for Runtime Sharing.
5. В поле Identifier укажите идентификатор символа, извлекаемого из совместно используемой библиотеки модели runtime (в фильме-источнике). Будьте внимательны при вводе идентификаторов. Если вы допустите ошибку, требуемый символ не сможет быть найден и загружен в фильме-адресате.
6. В поле URL введите точный адрес, на котором расположена общая библиотека модели runtime. Важно правильно ввести URL-адрес, иначе фильм-источник не будет найден и фильм-адресат не получит требуемый символ.
7. Щелкните на кнопке ОК.

Разрыв связи с общей библиотекой модели runtime

Чтобы разорвать связь с совместно используемой библиотекой модели runtime, выполните такие действия.

1. Загрузите фильм-адресат, содержащий символ из той библиотеки, с которой нужно разорвать связь.
2. Откройте окно библиотеки фильма посредством команды Library меню Window.
3. Выберите символ: графику, клип или кнопку.
4. Активизируйте команду Properties меню Options, чтобы отобразить диалоговое окно Symbol Properties.
5. Отключите флажок Import for Runtime Sharing.
6. Щелкните на кнопке ОК.

Связь с общей библиотекой модели author-time

В случае использования общей библиотеки модели author-time вы можете заменить любой символ в текущем фильме символом из другого фильма.

ПРИМЕЧАНИЕ

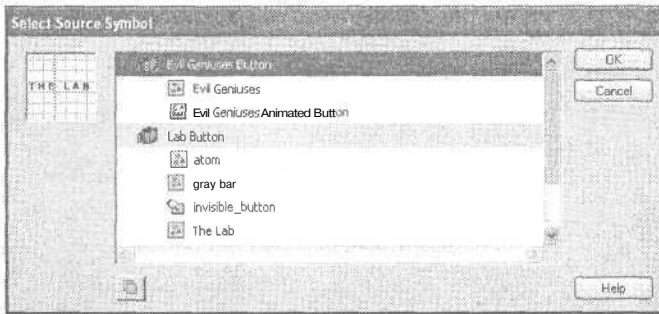
При работе с совместно используемой библиотекой модели author-time фильм-источник должен находиться на жестком диске вашего компьютера или быть доступным в локальной сети.

Помните, что хотя в фильме-адресате имя и свойства символа сохраняются, его содержимое полностью заменяется содержимым символа из фильма-источника, с которым установлена связь. Любой компонент, с которым связан символ в фильме-источнике (например, звуковой файл в клипе), также копируется в фильм-адресат.

Рассмотрим пошагово, как заменить символ в фильме-адресате символом из общей библиотеки модели author-time.

1. Загрузите фильм, в котором вы хотите заменить символ.
2. Откройте окно библиотеки фильма с помощью команды Library меню Window.

3. Выберите символ, подлежащий замене.
4. Активизируйте команду Properties меню Options, вследствие чего отобразится диалоговое окно Symbol Properties.
5. Щелкните на кнопке Browse (Просмотреть), которая расположена в нижней части окна свойств символа. В результате на экран будет выведено еще одно окно — Locate Macromedia Flash Document File (Адрес файла Macromedia Flash).
6. Укажите файл с расширением FLA, содержащий символ, который вы хотите использовать в качестве заменителя, и щелкните на кнопке Open (Открыть).
7. Перед вами появится диалоговое окно Select Source Symbol (Выбор символа из источника), где представлен полный перечень символов, содержащихся в указанном FLA-файле. Выберите символ, который вы хотите использовать в качестве заменителя.



8. Щелкните на кнопке ОК.
9. Снова нажмите кнопку ОК, но на этот раз — в диалоговом окне Symbol Properties. Обратите внимание на то, что содержимое символа заменено.

ПРИМЕЧАНИЕ

Чтобы заменить один символ другим в одном и том же фильме, щелкните на кнопке Symbol в области Source диалогового окна Symbol Properties. В результате откроется диалоговое окно Select Source Symbol, где вы сможете выбрать символ, который будет использоваться в качестве заменителя.

Работа с экземплярами символов

Мы уже неоднократно акцентировали внимание на том, что в область действия из библиотеки добавляется не символ, а его копия, называемая экземпляром. Изменения экземпляра никак не влияют на вид оригинала, хранящегося в библиотеке. Таким образом, в результате обработки экземпляр может утратить малейшее сходство с оригинальным символом.

ПРИМЕЧАНИЕ

При редактировании символа в библиотеке все его экземпляры обновляются, а при редактировании экземпляра изменяется только сам экземпляр.

Далее мы познакомимся с различными характеристиками экземпляров символов и изучим приемы управления ими.

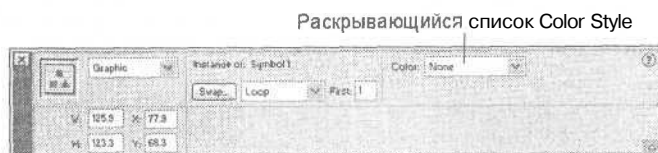
Обработка экземпляров с помощью панели Properties

Этот раздел посвящен вопросам управления основными параметрами экземпляров, определяющими их внешний вид.

Некоторые свойства экземпляра (прозрачность, яркость, цвет и т. п.) имеют особое значение при создании анимации, поскольку, изменяя их, можно добиться эффекта преобразования экземпляра с течением времени. Например, варьируя значения прозрачности, можно симитировать появление и исчезновение объекта. Подробно о таких операциях рассказывается в главе 9.

При изменении внешнего вида экземпляра в редких случаях не используется раскрывающийся список Color Style (Цветовой стиль).

1. Выделите экземпляр символа.
2. Загрузите панель Properties с помощью команды Properties меню Window.



3. Раскройте список Color Style (Цветовой стиль).

Давайте остановимся на каждом пункте этого списка.

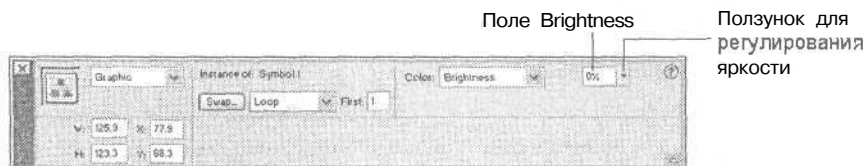
None

Благодаря этой команде у вас есть возможность восстановить изначальное состояние экземпляра после изменений.

Brightness (Яркость)

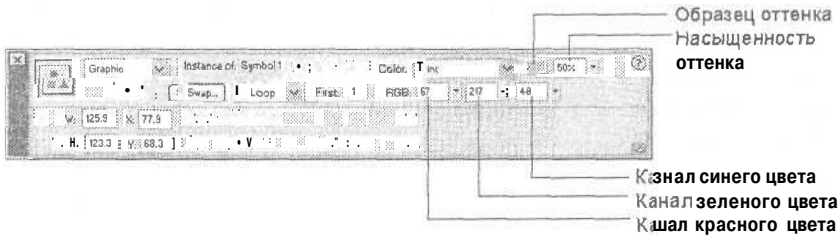
Выбрав команду Brightness, вы сможете изменить яркость экземпляра путем ввода значения (в процентах) в поле Brightness или за счет смещения ползунка. Диапазон допустимых значений — от 100 процентов белого цвета до -100 процентов черного.

При указании значения 0 в поле Brightness яркость экземпляра не меняется.



Tint (Цветовой тон)

Данная команда служит для изменения цветового оттенка экземпляра. Существует несколько способов выполнения этой операции. Самый простой — щелчок в окне с образцом оттенка и выбор требуемого цвета в появившейся палитре. Хорошей традицией является ввод точных значений компонентов цвета в поля R, G и B. Это позволит добиться полного соответствия оттенков в ваших работах. Насыщенность цветового оттенка регулируется в поле Tint Amount.

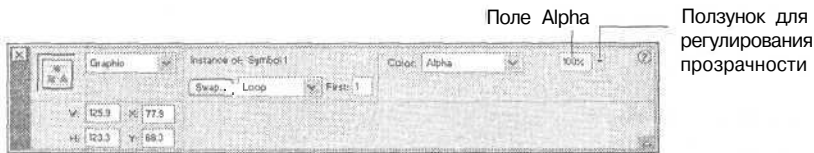


ПРИМЕЧАНИЕ

Значения цветовых компонентов и насыщенности можно установить с помощью ползунков, которые появляются при щелчке на небольшой стрелке *справа* от каждого поля.

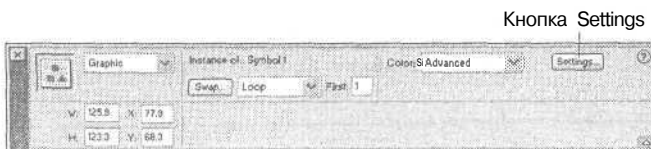
Alpha

Команда Alpha предназначена для управления прозрачностью выбранного экземпляра. Чтобы изменить прозрачность, введите новое значение в поле Alpha с клавиатуры или посредством *ползунка*. Допускается использование значений от 0 (полная прозрачность) до 100 (непрозрачность).



Advanced (Дополнительно)

В результате активизации команды Advanced раскрывающегося списка Color Style или выполнения щелчка на кнопке Settings (Установки), которая расположена на панели Properties, открывается диалоговое окно Advanced Effect (Дополнительный эффект). В этом окне предоставляется возможность одновременного изменения цвета и прозрачности.

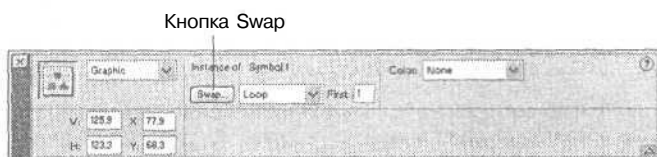


Диалоговое окно Advanced Effect содержит два набора элементов управления. Четыре поля в правой части — xR, xG, xB и xA) — предназначены для ввода абсолютных значений цвета и прозрачности выбранного экземпляра, а четыре поля в левой части — Red, Green, Blue и Alpha — служат для задания относительных значений указанных параметров.

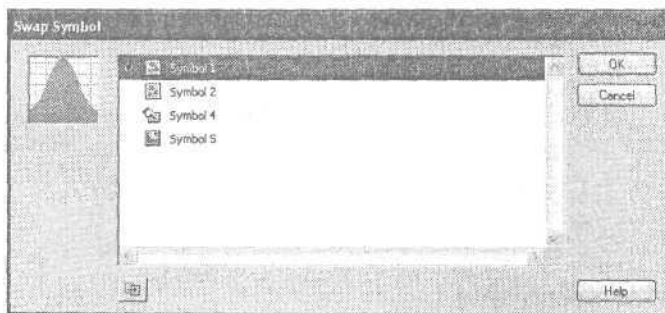
Замена экземпляров

Возможны ситуации, когда необходимо заменить один экземпляр символа другим. При этом новый экземпляр должен занять точно то же место, в котором располагался его предшественник. Вот как решается данная задача.

1. Выделите экземпляр, подлежащий замене.
2. Загрузите панель Properties с помощью команды Properties меню Window.
3. Щелкните на кнопке Swap (Замена).



4. В открывшемся диалоговом окне Swap Symbol (Замена символа) выберите символ, которым вы хотите заменить выделенный экземпляр, и нажмите кнопку OK.



Редактирование символов и экземпляров

В процессе редактирования экземпляров символов программа Flash позволяет использовать все свои стандартные средства рисования, черчения и трансформации. Не забудьте предварительно выделить символ.

СОВЕТ

Для получения полного контроля над экземпляром символа следует произвести его разбивку. С этой целью выделите символ и в меню Modify активизируйте команду Break Apart.

Если редактированию подлежит оригинал символа, последовательность действий будет следующей. Откройте окно библиотеки. Выберите символ и активизируйте команду Edit меню Options (в качестве альтернативы можно выполнить двойной щелчок на значке символа в окне библиотеки). В результате вы попадете в режим редактирования, где и осуществляется модификация самих символов. Помните, что при изменении символа автоматически обновляются все его экземпляры.

Панель Movie Explorer

В ранних версиях Flash было крайне трудно следить за структурой проектов, особенно при использовании сценариев ActionScript, вложенных символов и сложных клипов. К радости разработчиков Flash-фильмов, современные версии программы содержат прекрасное средство для анализа структуры проекта — панель Movie Explorer.

На панели Movie Explorer отображается иерархия компонентов фильма (рис. 7.5). Здесь осуществляется поиск символов и экземпляров в пределах фильма, а также замена текстовых блоков и шрифтов. Содержимое панели Movie Explorer можно скопировать в буфер и вывести на печать. Рассматриваемая панель загружается в результате вызова команды Movie Explorer меню Window либо нажатия клавиш Option/Alt+F3.

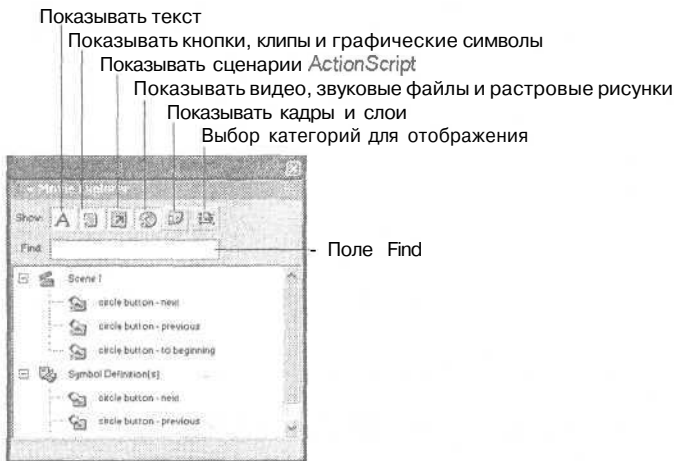


Рис. 7.5. Панель Movie Explorer

В следующих разделах мы расскажем о различных командах, предоставляемых в распоряжение разработчика на панели Movie Explorer. Вы также узнаете, как отображать и скрывать категории, осуществлять поиск элементов с помощью поля Find, управлять свойствами символа и выполнять замену шрифтов.

Команды панели

Панель Movie Explorer (как и большинство других панелей Flash) содержит меню Options, для открытия которого следует щелкнуть на значке в правом верхнем углу. Какие же команды доступны в этом меню?

- ▶ Go to Location (Переход). Осуществляет переход к нужному компоненту фильма: слою, сцене или кадру. Еще один способ — двойной щелчок на элементе в окне панели Movie Explorer.
- ▶ Go to Symbol Definition (Переход к определению символа). Выделяет все файлы, связанные с выбранным символом. Рассматриваемая команда работает только тогда, когда нажата кнопка Show Buttons, Movie Clips and Graphics (Показывать кнопки, клипы и графические символы) и установлен режим Movie Elements.
- ▶ Select Symbol Instances (Выбрать экземпляры символа). Осуществляет переход к сцене, которая содержит выбранный экземпляр символа. Рассматриваемая команда работает только тогда, когда нажата кнопка Show Buttons, Movie Clips and Graphics (Показывать кнопки, клипы и графические символы) и установлен режим Movie Elements.
- ▶ Find in Library (Поиск в библиотеке). Открывает окно библиотеки и находит выделенный символ.
- ▶ Rename (Переименовать). Позволяет переименовать выбранный символ.
- ▶ Edit in Place (Правка на месте). Предоставляет возможность модифицировать символ, выбранный в области действия, без переключения в режим редактирования.
- ▶ Edit in New Window (Правка в новом окне). Открывает новое окно для модификации выделенного символа.
- ▶ Show Movie Elements (Показать элементы фильма). Выводит все элементы фильма, упорядоченные по сценам, в окне Movie Explorer.
- ▶ Show Symbol Definitions (Показывать определения символов). Выводит все файлы, связанные с выбранным элементом, в окне Movie Explorer.
- ▶ Show All Scenes (Показывать все сцены). Выводит все сцены фильма в окне Movie Explorer.
- ▶ Copy All Text to Clipboard (Копировать весь текст в буфер обмена). Копирует в буфер обмена весь текст, который впоследствии может быть вставлен в любой документ.
- ▶ Cut (Вырезать). Вырезает выбранный текст.
- ▶ Copy (Копировать). Копирует выбранный текст (но не помещает его в буфер обмена, как команда Copy All Text to Clipboard).
- ▶ Paste (Вставить). Вставляет текст, который ранее был скопирован либо в программе Flash, либо в другом приложении.
- ▶ Clear (Очистить). Удаляет выделенный текст.
- ▶ Expand Branch (Развернуть ветвь). Разворачивает выбранную ветвь иерархического дерева. Тот же результат можно получить, щелкнув на знаке «плюс» слева от названия раздела.

- ▶ Collapse Branch (Свернуть ветвь). Сворачивает выбранную ветвь иерархического дерева. Тот же результат можно получить, щелкнув на знаке «минус» слева от названия раздела.
- ▶ Collapse Others (Свернуть остальные ветви). Автоматически сворачивает все разделы иерархической структуры, за исключением выбранной.
- ▶ Print (Печать). Выводит на печать все содержимое панели Movie Explorer (в том числе все развернутые разделы иерархии). Это прекрасный способ создать копию структуры Flash-фильма.

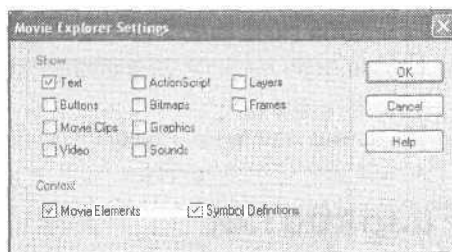
Использование панели

Теперь, когда вам знакома каждая команда панели Movie Explorer, давайте рассмотрим некоторые методы организации содержимого этой панели и способы манипулирования им.

Выбор категорий для отображения

Панель Movie Explorer содержит несколько кнопок, позволяющих определять, какие категории элементов следует отображать. Вот эти кнопки (слева направо): Show Text (Показывать текст), Show Buttons, Movie Clips and Graphics (Показывать кнопки, клипы и графические символы), Show ActionScripts (показывать сценарии ActionScript), Show Video, Sounds and Bitmaps (Показывать видео, звуковые файлы и растровые рисунки), Show Frames and Layers (Показывать кадры и слои).

Завершает цепочку кнопка Customize Which Items to Show (Ручная настройка выводимых категорий), предоставляющая доступ к окну Movie Explorer Settings (Параметры панели Movie Explorer). Здесь вы самостоятельно можете задать, какие группы элементов должны быть видимыми в окне панели Movie Explorer.



Поиск элементов

Когда работа ведется над масштабным проектом, бывает трудно отыскать нужный символ или экземпляр. В таких ситуациях на помощь приходят поисковые средства панели Movie Explorer. Введите имя символа, экземпляра, шрифта либо сценария ActionScript или же номер кадра в поле Find, и программа моментально переведет вас в то место иерархического дерева, где находится данный элемент.

Редактирование символов

В некоторых случаях требуется модифицировать экземпляр символа, не покидая панель Movie Explorer. Вот что для этого нужно сделать:

1. Найдите экземпляр символа, который подлежит редактированию. Воспользуйтесь с этой целью инструментом поиска панели Movie Explorer.
2. В меню Options панели Movie Explorer или в контекстном меню активизируйте команду Edit in Place или Edit in New Window.

ПРИМЕЧАНИЕ

Переход в режим редактирования символа осуществляется также при выполнении двойного щелчка на значке экземпляра в окне Movie Explorer.

3. По завершении правки экземпляра вернуться к работе над фильмом.

Замена шрифтов

Средства панели Movie Explorer позволяют осуществлять поиск и производить замену любого шрифта, используемого в фильме. Данное свойство особенно полезно, если шрифт имеет множество экземпляров. Важно, что вам не придется производить многократную замену: изменение шрифта приведет к автоматическому обновлению всех его экземпляров.

1. Убедитесь, что панель Movie Explorer открыта.
2. Введите имя шрифта в поле Find.
3. В иерархической структуре найдите шрифт и выделите его.

СОВЕТ

Если в фильме шрифт используется несколько раз, на панели Movie Explorer фиксируются все случаи его применения. Чтобы при выделении охватить каждый из этих случаев, выполняйте щелчки при нажатой клавише Ctrl (Windows) или Cmd (Macintosh).

4. Загрузите панель Properties.
5. С помощью элементов панели Properties произведите все необходимые изменения.
6. Когда все изменения будут внесены, ваш фильм автоматически обновится.

Разработки

На практике существует множество прекрасных примеров удачной реализации объектов многократного использования (векторных и растровых изображений, звуковых файлов и т. д.), обеспечивающих унификацию дизайна. К их числу принадлежит и Web-узел компании Doug Chiang Studio (www.dchlang.com).

Основанная Доугом Чиангом, ответственным дизайнером сериала «Звездные войны», компания Doug Chiang Studio в настоящий момент работает над проектом фильма/книги *Robota: Reign of Machines*, который должен увидеть свет в этом году. На 160 страницах рассказывается об «отношениях» между технологией и живой природой в фантастическом обществе. Web-узел адресован поклонникам таланта Доуга Чианга, желающим познакомиться с его работами и текущими проектами студии.

На Web-узле компании Red Industries (www.redindustries.com), разработанном с помощью таких приложений, как Dreamweaver, Fireworks и Flash, неповторимые иллюстрации Доуга Чианга и звук сочетаются с прекрасно выполненной навигационной схемой, что в комплексе производит прекрасное впечатление (рис. 7.6).

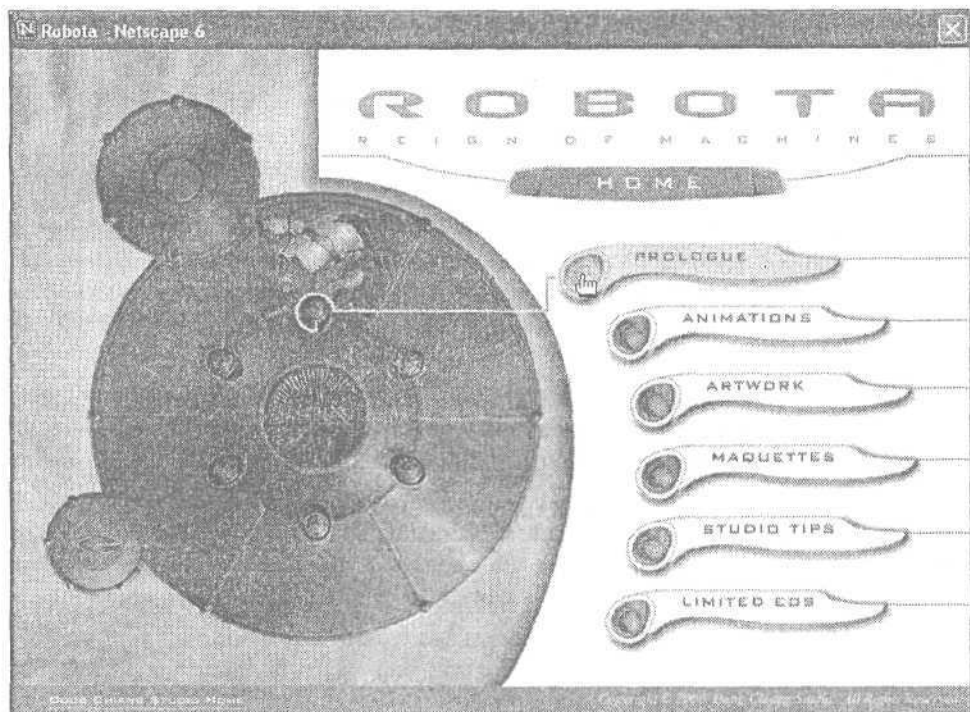


Рис. 7.6. Web-узел компании Doug Chiang Studio

Резюме

Любая сложная анимация может быть создана только с использованием *символов*. Большинство графических элементов (или даже все) следует преобразовать в символы или изначально создавать как таковые. Глава содержит детальную информацию о символах, а также о средствах и приемах работы с ними. Вам был представлен полный спектр операций по управлению символами: создание, импортирование и вставка символов, конвертирование графических изображений в символы, редактирование экземпляров символов в области действия и оригинальных символов в библиотеке.

Значительное внимание в главе уделено библиотекам — хранилищам всех символов Flash-проекта. Подробно описаны методы организации библиотек, а также приемы работы с внешними и общими библиотеками. Благодаря возможности обращения к *внешним* библиотекам разработчик не ограничен ресурсами библиотеки текущего проекта и имеет доступ к компонентам библиотек других фильмов. А *общие* библиотеки вообще делают необязательным формирование отдельных библиотек для Flash-фильмов.

Завершает главу обзор элементов двух важнейших панелей — панели Properties, предназначенной для изменения свойств экземпляров символов, и панели Movie Explorer, мощного средства управления элементами фильма.

Глава 8

Слои

До сих пор мы создавали объекты на одном слое. Настало время перейти в третье измерение! Нет, речь не идет о разработке трехмерных изображений. К изучению этой сложной и увлекательной темы мы перейдем лишь в главе 26. А пока наша задача — научиться составлять двухмерные изображения из разных слоев.

Как и многие графические приложения, Flash позволяет создавать любое число слоев. Слой можно сравнить с прозрачной пленкой, на которой изображен объект. Накладывая слой один на другой, мы получаем изображение.

Слои удобны тем, что при работе со сложным изображением позволяют выборочно отображать его составляющие и сосредоточиться на обработке той или иной детали. Сложно переоценить роль слоев и в процессе подготовки анимации. Вы убедитесь в этом, прочитав главу 9.

В настоящей главе предлагаем ознакомиться с такими темами:

- ▶ принципы использования слоев;
- ▶ создание слоя;
- ▶ каталоги слоев;
- ▶ добавление объектов на слой;
- ▶ настройка слоев;
- ▶ редактирование слоев;
- ▶ распределение объектов по слоям;
- ▶ создание направляющих слоев;
- ▶ создание слоев-масок.

Принципы использования слоев

Программа Flash позволяет создавать любое количество слоев и располагать на каждом из них произвольное число объектов. Слои не имеют обозначений в области действия, но каждый из них представлен в виде строки на временной шкале. Именно отсюда осуществляется доступ к слоям.

Позиция слоя относительно других имеет большое значение. Поскольку слои образуют «стопку», верхний слой находится на переднем плане, а нижний — на заднем. Рассмотрим пример. Предположим, у вас есть два изображения на двух разных слоях, занимающих одинаковое пространство в области действия. Изображение, размещенное на верхнем слое временной шкалы, отобразится перед изображением нижнего слоя.



Слои бывают различных видов, каждый из которых играет определенную роль в фильме Flash. С их особенностями мы познакомимся в разделах «Создание направляющих слоев» и «Создание слоев-масок». Каждый слой имеет набор специфических параметров, доступ к которым предоставляется на временной шкале. Обзор параметров слоев дан в разделе «Настройка слоев».

Выбор слоя осуществляется путем выполнения щелчка на его имени в левой части временной шкалы. Имя активного слоя выводится белыми буквами на черном фоне и снабжается значком в виде карандаша. Следует помнить о том, что новые объекты добавляются на текущий слой.

Создание слоя

После запуска приложения Flash создается один слой, о чем свидетельствует запись па временной шкале. Конечно, вы не обязаны добавлять новые слои, но использование одного слоя в процессе подготовки крупномасштабных Flash-проектов не только усложняет работу, но и приводит к возникновению ряда проблем функционирования. Подкрепим сказанное примером. Из главы 12 вы узнаете, что существует возможность создания сценариев, которые активизируются, когда головка воспроизведения проходит через определенный кадр. Эти сценарии называются *действиями кадра*. Если в фильме имеется только один слой, действие кадра влияет на все содержимое, что часто вовсе не требуется. При размещении объектов на разных слоях действие кадра оказывает влияние только на объекты заданного слоя. Рекомендуется отводить отдельный слой для каждого объекта сложной анимации. Это обеспечит повышенный контроль над каждым объектом в отдельности и, следовательно, над фильмом в целом. Число слоев, которые вы можете создать, ограничено только объемом памяти вашего компьютера.

ПРИМЕЧАНИЕ

На заключительном этапе все слои Flash-фильма объединяются в один, поэтому их количество не повлияет на общий объем файла.

Существует несколько способов создания слоя:

- ▶ Щелкните на кнопке Insert Layer, расположенной в нижнем левом углу временной шкалы.

- ▶ Активизируйте команду Layer меню Insert.
- ▶ Выберите команду Insert Layer в контекстном меню, для открытия которого щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Shift (Macintosh)

Новые слои можно создавать на любом этапе работы над проектом. Flash производит вставку нового слоя поверх текущего.

Добавление объектов на слой

В пустых слоях нет никакого смысла, поэтому поговорим о наполнении слоев. Добавить изображение на слой несложно. Все, что требуется, — это выбрать слой и создать некоторое изображение в области действия (используя инструменты рисования и черчения) либо вставить экземпляр символа из библиотеки. Напомним, что все изображения, создаваемые в области действия, автоматически помещаются на активный слой.

Когда слой не содержит объектов, его первый кадр, *ключевой кадр*, также является пустым и обозначается полым кружком. При добавлении объекта на слой появляется черный кружок — признак того, что ключевой кадр имеет содержимое. О кадрах подробно рассказывается в главе 9.

Настройка слоев

Теперь, когда вам известно, как создать слой и добавить на него объекты, поговорим о настройке слоев. Каждый слой обладает набором специфических параметров, доступ к которому можно получить, выполнив двойной щелчок на значке слева от имени слоя.

Далее вы научитесь скрывать и снова отображать слои, работать в режиме контурного представления, изменять цвет контуров объектов, а также блокировать слои и снимать их блокировку.

Управление отображением слоев

В процессе работы над Flash-фильмом у вас есть возможность отображать нужные слои и скрывать те, которые на данном этапе не требуют редактирования. Содержимое скрытого слоя становится невидимым, вследствие чего уменьшается число объектов в области действия, и вы можете сосредоточить внимание только на тех элементах изображения, с которыми в настоящий момент работаете. В любой момент слой можно вновь отобразить на экране.

Далее описаны приемы, позволяющие скрыть или отобразить нужное количество слоев. Сразу отметим, что управление отображением слоев осуществляется в столбце, над которым изображен глаз.

- ▶ Чтобы скрыть отдельный слой, перейдите в его строку и щелкните на точке в столбце, в заголовке которого изображен глаз. Точка превратится в красный крестик — признак того, что слой скрыт. Отображение слоя возобновляется при выполнении щелчка на красном крестике.

- ▶ Чтобы скрыть все слои одновременно, щелкните на пиктограмме в виде глаза. Отображение всех слоев возобновляется при повторном щелчке на указанной пиктограмме. Тот же результат можно получить, выбрав команду Show All в контекстном меню, для открытия которого следует щелкнуть правой кнопкой мыши (Windows) или выполнить щелчок при нажатой клавише Shift (Macintosh).
- ▶ Чтобы скрыть все слои, кроме одного, нажмите клавишу Option/Alt и, удерживая ее, щелкните на точке в строке того слоя, который вы хотите оставить видимым. Альтернативное средство — команда Hide Others контекстного меню.

Контурное представление объектов слоя

В режиме контурного представления все объекты, расположенные на заданном слое, отображаются в виде цветных контуров. Данный режим полезен при редактировании и тестировании анимации.

Активизировать режим контурного представления объектов можно несколькими способами:

- ▶ Щелкните на цветном квадратике в строке слоя. В результате значок станет полым, указывая, что объекты данного слоя представлены в виде контуров. Для отключения контурного режима щелкните на полном значке. На рис. 8.1 продемонстрированы оба состояния значка контурного режима.
- ▶ Чтобы установить режим контурного представления для объектов всех слоев, щелкните на значке Show All Layers as Outlines в заголовочной части области слоев. Для отмены данного режима повторите щелчок на указанном значке.
- ▶ Существует возможность активизировать контурный режим для всех слоев, за исключением одного. Для этого, удерживая клавишу Option/Alt, щелкните на соответствующем значке в строке того слоя, объекты которого должны отображаться в обычном режиме.

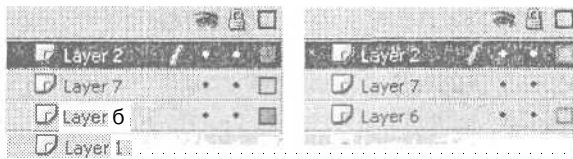
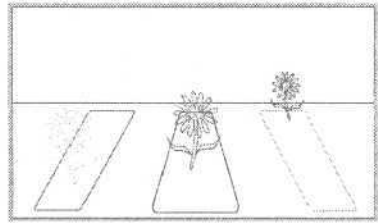


Рис. 8.1. Режим контурного представления объектов слоя включен (справа) и отключен (слева)

Изменение цвета контуров объектов слоя

Наверное, вы обратили внимание на то, что каждый значок контурного режима имеет свой цвет. В таком цвете выводятся контуры объектов слоя при переходе в данный режим. Таким образом, если квадратик окрашен в красный цвет, то и контуры объектов данного слоя в соответствующем режиме имеют красный

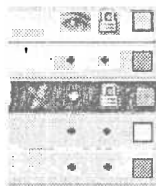
цвет. Вы можете изменить цвета, используемые по умолчанию. Как это сделать, рассказывается ниже.

1. Откройте диалоговое окно Layer Properties (Свойства слоя) одним из следующих способов:
 - > Выполните двойной щелчок на значке слоя (слева от имени слоя).
 - > Активизируйте команду Properties контекстного меню.
 - > Выберите слой, для которого хотите поменять цвет в контурном режиме, и вызовите команду Layer меню Modify.
2. В диалоговом окне Layer Properties щелкните на образце цвета Outline Color и выберите подходящий цвет в палитре.
3. Нажмите кнопку ОК.

Блокировка слоев

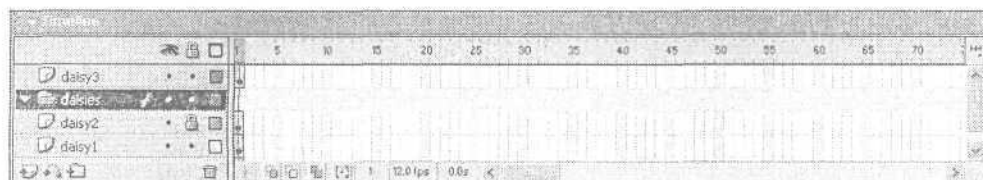
По умолчанию разрешается редактирование содержимого создаваемых слоев. Поскольку Flash позволяет манипулировать содержимым слоя, даже если он не выбран, вы можете случайно изменить объекты одного слоя при работе с другим. Порой такая ошибка чревата потерей результатов многочасовой работы. Предотвратить подобные ситуации позволяет блокировка слоев. Существует несколько способов блокировки/разблокировки:

- ▶ Щелкните на точке в столбце блокировки, который обозначен пиктограммой в виде замка. Вместо точки отобразится замок, указывающий, что слой в настоящий момент заблокирован. Блокировка отменяется в результате выполнения щелчка на изображении замка в строке слоя.
- ▶ Чтобы заблокировать все слои, щелкните на пиктограмме с изображением замка в заголовочной части области слоев. Для отмены данного режима повторите щелчок на указанной пиктограмме.
- ▶ Существует возможность заблокировать все слои, за исключением одного. Для этого щелкните на точке в столбце блокировки рядом с именем того слоя, на который не должна распространяться операция блокировки. Альтернативное средство — команда Lock Others контекстного меню.



Каталоги слоев

Flash позволяет создавать для слоев специальные каталоги, называемые *каталогами слоев*. Доступ к ним осуществляется с помощью временной шкалы.



Предположим, вы используете в фильме несколько мультипликационных персонажей. Каждая деталь любого персонажа (рука, голова, нога, уши и т. д.) занимает отдельный слой. Чтобы не перепутать части разных персонажей, необходимо распределить изображения по каталогам, каждый из которых отвечает отдельному персонажу.

Вы можете сворачивать и разворачивать каталоги, тем самым скрывая связанные с ними слои на временной шкале, что не повлияет на изображение в области действия. Многие свойства каталогов слоев присущи обычным слоям. Например, существует возможность заблокировать слои каталога, сделать их невидимыми, а также представить в контурном режиме.

Следующие разделы содержат детальные сведения о характеристиках каталогов слоев. Вы научитесь создавать, редактировать, сворачивать и разворачивать каталоги: слоев, а также добавлять в них новые слои.

ПРИМЕЧАНИЕ-

ПОЗИЦИЯ каталога слоев изменяется тем же способом, что и позиция слоя, но об этом чуть позже.

Создание каталога слоев

Создать каталог слоев несложно. Выполните одно из следующих действий:

- ▶ Щелкните на кнопке Layer Folder в левой нижней части временной шкалы.
- ▶ Выберите команду Layer Folder в меню Insert.
- ▶ Активизируйте команду Insert Folder контекстного меню, для открытия которого следует щелкнуть правой кнопкой мыши (Windows) или выполнить щелчок при нажатой клавише Shift (Macintosh).

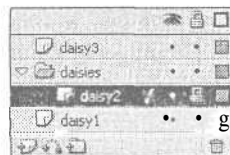
При выполнении любой из указанных процедур Flash разместит новый каталог слоев над активным слоем.

Программа предоставляет возможность преобразовать выбранный слой в каталог слоев. Для этого с помощью команды Layer меню Modify необходимо открыть диалоговое окно Layer Properties и изменить в нем тип слоя с Normal на Folder.

Добавление слоя в каталог слоев

Рассмотрим, как добавить слой в каталог.

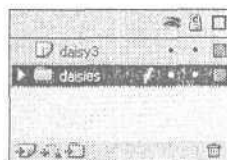
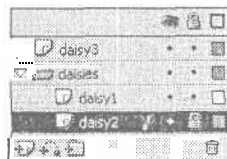
1. Создайте новый каталог одним из описанных способов.
2. Установите указатель мыши на имени слоя и, не отпуская кнопку манипулятора, перетащите его к значку каталога. (Обратите внимание на то, что при перемещении указателя мыши над значком каталога имя каталога подсвечивается.)
3. Отпустите кнопку мыши.
4. Вложенный слой отобразится на временной шкале ниже своего каталога с небольшим отступом, который служит признаком принадлежности слоя каталогу.



Развертывание и свертывание каталога слоев

Каталог слоев можно развернуть, чтобы просмотреть содержимое, и свернуть, чтобы скрыть его. Обе операции осуществляются с помощью небольшой стрелки, расположенной слева от имени каталога. Если стрелка указывает вниз, значит, каталог развернут, а входящие в него слои видны на временной шкале.

Если же стрелка указывает вправо, то каталог свернут, а его содержимое скрыто.



Редактирование слоев

Вы уже умеете создавать слои, добавлять на них объекты и манипулировать свойствами слоя. В этом разделе речь пойдет о перемещении, копировании, удалении и переименовании слоев, а также об изменении позиции слоя на временной шкале.

Перемещение слоя

Как вы уже знаете, позиция слоя в стеке определяет положение его объектов относительно объектов других слоев. Следовательно, перемещая слой в стеке, вы можете расположить его объекты впереди или позади других элементов изображения. Для перемещения слоя выполните следующие действия:

1. Выберите слой, который хотите переместить.
2. Перетащите слой в требуемую позицию при нажатой кнопке мыши. Обратите внимание на то, что в процессе перемещения появляется черная полоса, указывающая возможное положение слоя.
3. Выбрав подходящую позицию, отпустите кнопку манипулятора.

Копирование слоя

На практике нередки ситуации, когда требуется создать точную копию слоя. Этот несколько утомительный процесс включает копирование содержимого слоя кадр за кадром и вставку его на новый слой. Вот как все происходит:

1. Выберите слой, который нужно скопировать, щелкнув на его имени. При этом будет выбрано и все содержимое слоя.
2. Активируйте команду Copy Frames меню Edit или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+Option/Alt+C`. В результате будет скопировано содержимое каждого кадра слоя.
3. Создайте новый слой.
4. Перейдите на только что созданный слой, щелкнув на его имени.
5. Вызовите команду Paste Frames меню Edit или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+Option/Alt+V`. Скопированное ранее содержимое появится на новом слое.

Удаление слоя

Теперь рассмотрим процедуру удаления слоя:

1. Перейдите на слой, подлежащий удалению.
2. Выполните одно из следующих действий:
 - > Нажмите кнопку Delete.
 - > Перетащите слой в область кнопки Delete.
 - ▷ Активизируйте команду Delete Layer контекстного меню, для открытия которого следует щелкнуть правой кнопкой мыши (Windows) или выполнить щелчок при нажатой клавише Shift (Macintosh).

СОВЕТ -

При удалении слоя удаляются и все расположенные на нем объекты. Следовательно, прежде чем удалять слой, убедитесь, не содержит ли он нужных объектов, поскольку Flash не выдает предупреждающих сообщений.

Переименование слоя

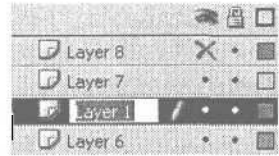
При создании слоям автоматически присваиваются невыразительные имена наподобие Layer 1. К счастью, в Flash предусмотрена возможность изменения имен слоев.

СОВЕТ

Как и файлам, слоям рекомендуется назначать имена, по которым легко определить их содержимое. Это позволит быстро находить требуемый слой при увеличении объема фильма.

Существуют различные способы переименования слоя:

- ▶ Выполните двойной щелчок на имени слоя (не на значке) и введите новое название в появившееся поле.
- ▶ Выполните двойной щелчок на значке слоя, который расположен слева от его имени. Откроется диалоговое окно Layer Properties (Свойства слоя). Введите новое имя в поле Name. Диалоговое окно Layer Properties загружается и с помощью команды Layer меню Modify.

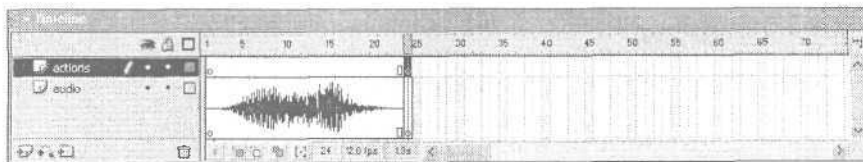


СОВЕТ

Если имя слоя не отображается полностью, перетащите вправо полосу, разделяющую левую и правую части временной шкалы.

Изменение высоты строки слоя

Каждый слой отображается в правой части временной шкалы в виде строки кадров определенной высоты. В некоторых ситуациях эту высоту требуется изменить. Например, если увеличить высоту слоя на временной шкале при работе со звуковыми эффектами, то можно увидеть амплитуду звукового сигнала.



В этом разделе речь идет только об изменении высоты слоя на временной шкале. Об операциях с кадрами рассказывается в главе 9.

Чтобы увеличить высоту слоя на временной шкале, выполните следующие действия.

1. Откройте диалоговое окно Layer Properties.
2. Выберите одно из значений в списке Layer Height: 100%, 200%, 300% и т. д.
3. Щелкните на кнопке ОК.

СОВЕТ

Вы, вероятно, заметили, что число слоев, отображающихся на временной шкале, ограничено размерами этой шкалы. Для просмотра всех слоев можно использовать полосу прокрутки, а можно увеличить высоту временной шкалы, сдвинув вниз линию, отделяющую ее от области действия.

Распределение объектов по слоям

Часто при импортировании векторных графических файлов (например, из приложений Adobe Illustrator и Macromedia Freehand) изображение содержит множество негруппированных объектов.

К счастью, Flash MX располагает командой, которая позволяет распределить выбранные объекты таким образом, чтобы каждый из них занимал отдельный слой.

Создаваемым слоям Flash присваивает уникальные имена. Так, если распределяются символы надписи, которая была разбита на части (см. главу 5), каждому слою в качестве имени присваивается определенная буква (например, F, L, A, S и H). Слою, содержащему библиотечный объект (например, символ, растровое изображение или клип), Flash MX назначает то имя, которое имеет сам объект. Слой с экземпляром символа получает имя, как у данного экземпляра. Наконец, слою, на котором расположен графический объект, созданный с помощью инструментов рисования и черчения в среде Flash, присваивается стандартное имя Layer с порядковым номером.

Вот как осуществляется распределение объектов по слоям:

1. Выделите в области действия объекты, которые нужно распределить. Помните, что выбирать можно только объекты, расположенные на одном слое.
2. Активизируйте команду Distribute to Layers меню Modify.

Создание направляющих слоев

Существуют различные виды слоев. Одна из разновидностей — *направляющие слои*, которые, в свою очередь, также делятся на несколько видов: *обычные* и

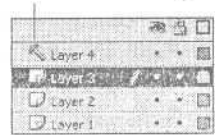
направляющие движения. На обычный направляющий слой добавляют объекты, играющие вспомогательную роль во время позиционирования элементов изображения (к их числу относятся, например, линии). Направляющая движения применяется для создания пути перемещения анимированного объекта.

В данном разделе вы научитесь преобразовывать обычный слой в направляющий, а также создавать направляющую движения. Остальные вопросы анимации освещены в главе 9.

Все направляющие слои, как обычные, так и направляющие движения, не включаются в окончательный вариант фильма Flash.

Чтобы создать направляющий слой, выполните следующие действия.

1. Выберите слой, который вы хотите преобразовать в направляющий.
2. Для получения обычного направляющего слоя активизируйте команду Guide контекстного меню (в упражнениях неоднократно говорилось о способах его открытия). В результате слева от имени слоя появится значок направляющего слоя Guide Layer.



3. Направляющую движения можно создать тремя способами:

- > Щелкните на значке Add Motion Guide, который расположен в нижней части временной шкалы.
- > Вызовите команду Motion Guide меню Insert.
- > Активизируйте команду Add Motion Guide контекстного меню.

Значок Motion Guide Layer



Вследствие выполнения любого из указанных действий создается дополнительный слой, который размещается над текущим слоем. Признак слоя Motion Guide — символ в виде дуги с кружком на конце.

Направляющий слой, как и направляющую движения, можно создавать с помощью диалогового окна Layer Properties (Свойства слоя) путем выбора соответствующего переключателя.

Создание слоев-масок

Слой-маски используются для создания пустот, через которые видны объекты нижнего слоя. Представьте, что вы смотрите сквозь замочную скважину на сложную сцену. Поскольку замочная скважина имеет определенную форму и размер, вы можете видеть только ограниченную часть сцены. Именно так и работают слой-маски. На рис. 8.2 показан пример использования слоя-маски.

Замечательное свойство слоев-масок состоит в том, что их можно группировать для получения сложных эффектов. Интересны анимированные маски, которые перемещаются по изображению.

Возможности масок безграничны, а создаваемые с их помощью эффекты разнообразны и интересны. Один из примеров — просмотр сцены через телескоп.

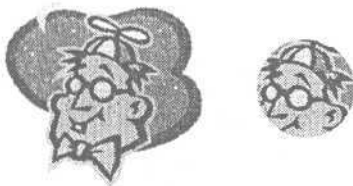


Рис. 8.2. Изображение без маски (слева) и после наложения маски (справа)

ПРИМЕЧАНИЕ

К сожалению, нельзя применять слои-маски внутри символов типа Button.

Поскольку процесс анимирования слоев-масок будет рассмотрен в главе 9, в этом разделе мы расскажем только о создании статических слоев-масок.

1. Выберите слой, содержимое которого будет просматриваться сквозь маску. Помните, что маска действует только на объекты слоя, который расположен непосредственно под ней.
1. Вставьте слой, активизировав команду Layer меню Insert или щелкнув на кнопке Insert Layer, которая находится в нижней части временной шкалы.
3. На новом слое создайте область произвольной формы, которая обязательно *должна быть* заполнена. После преобразования слоя в маску области, которые были заполнены, становятся прозрачными, а области без сплошной заливки — непрозрачными.
4. Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Shift (Macintosh), чтобы открыть контекстное меню. Активизируйте в нем команду Mask.
5. Слой превратится в маску. Обратите внимание на то, что значки слоя-маски и маскированного слоя изменились.



Разработки

В любом заслуживающем внимания проекте удачное сочетание дизайна, интерактивности и анимации достигается исключительно благодаря слоям. Примером такого проекта может служить Web-узел компании Blitz Digital Studios (www.blitzds.com).

Компания Blitz Digital Studios, офис которой находится в городе Валенсия, штат Калифорния, специализируется на производстве интерактивных информационных и развлекательных продуктов, предназначенных для коммерческого распространения на телевидении, а также на новых рынках сетевых технологий и систем беспроводной связи.

Web-узел компании Blitz Digital Studios выполнен в изящном стиле, здесь великолепный дизайн сочетается с хорошо продуманной схемой расположения интерактивных элементов (рис. 8.3).

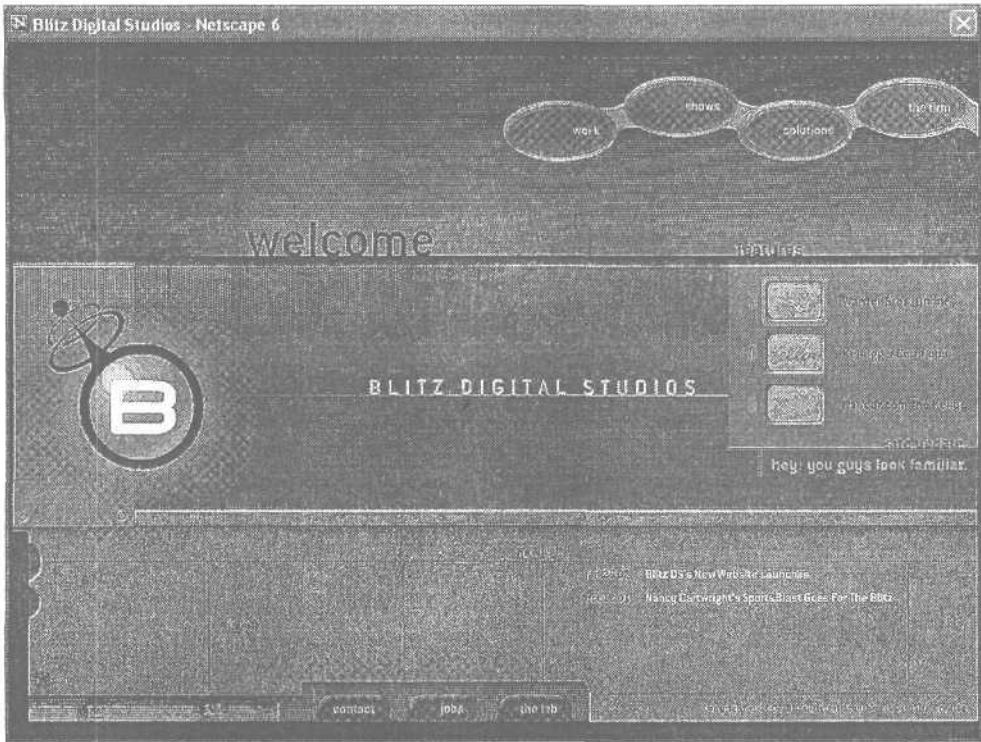


Рис. 8.3. Web-узел компании Blitz Digital Studios

Резюме

Правильно организованная анимация состоит из множества слоев. Рекомендуется для каждого объекта изображения отводить отдельный слой. Flash позволяет создавать любое число слоев.

Прочитав настоящую главу, вы научились управлять слоями: создавать новые слои, добавлять на них объекты, объединять слои в каталоги и, наконец, редактировать отдельные слои.

Особый интерес представляет материал о масках и направляющих слоях, позволяющих получать сложные эффекты.

Практикум 2

Создание изображения сада с использованием объектов и слоев

В части 2 вы познакомились с инструментами Flash, предназначенными для рисования, черчения и манипулирования объектами. Теперь вам предстоит использовать все полученные знания и навыки для выполнения задания — создания изображения сада.



Все необходимые файлы вы найдете на прилагаемом компакт-диске в каталоге Hands On 2. Файл `garden fla` содержит завершенную версию изображения сада. Хотя для решения данной задачи можно использовать готовые символы из библиотеки, мы рекомендуем создать все объекты самостоятельно. Ценность предлагаемого упражнения в том и состоит, чтобы своими силами выполнить все действия от начала до конца.

Фон: небо

Прежде всего необходимо создать фон рисунка.

1. Создайте новый документ, выбрав команду **New** в меню **File** или нажав клавиши **Cmd/Ctrl+N**.
2. Откройте диалоговое окно **Document Properties**. Для этого активизируйте команду **Document** меню **Modify** или нажмите клавиши **Cmd/Ctrl+J**.
3. Щелкните на элементе **Background Color** и в появившейся палитре выберите голубой цвет. Данный цвет в нашем рисунке будет символизировать небо.
4. Щелкните на кнопке **ОК**.

Почва

Теперь подготовим лужайку, на которой будут расти виртуальные цветы.

1. Откройте диалоговое окно **Layer Properties (Свойства слоя)**, дважды щелкнув на значке существующего слоя (слева от его имени).
2. В поле **Name** введите имя **Ground**.
3. Щелкните на кнопке **ОК**.
4. Активизируйте инструмент **Rectangle**. Щелкните на элементе **Fill Color**, который расположен в секции **Colors** панели **Tools**, и в появившейся палитре выберите зеленый цвет.
5. Поскольку создаваемый объект, почва, не должен иметь контуров, отмените обводку. С этой целью щелкните на образце цвета обводки **Stroke Color** (панель **Tools**). На экране отобразится палитра. Выполните щелчок на кнопке **No Color** в правом верхнем углу.

6. Все готово для создания почвы. С помощью инструмента Rectangle начертите прямоугольник, занимающий нижнюю половину области действия, как показано на рисунке.

ПРИМЕЧАНИЕ

С помощью инструмента Rectangle можно создать фигуру, выступающую за пределы области действия. Однако все, что находится на рабочей области (серая зона вокруг области действия), автоматически исключается из сцены при публикации.

7. Преобразуйте прямоугольник в символ. Для этого выделите прямоугольник и активизируйте команду Convert to Symbol меню Insert или нажмите клавишу F8.
8. В открывшемся диалоговом окне Convert to Symbol (Конвертировать в символ) введите имя Ground в поле Name. Убедитесь, что установлен переключатель Graphic.
9. Щелкните на кнопке ОК.

После выполнения описанных действий сцена должна выглядеть так, как показано на рисунке.

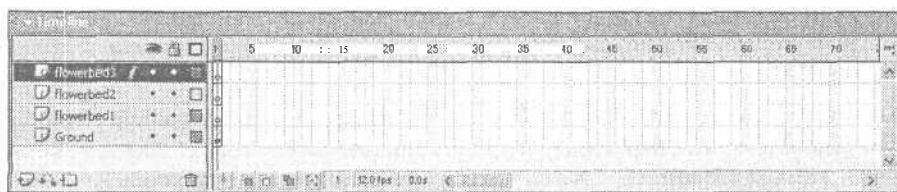


Подготовка к созданию клумб

Основной фон, небо и земля, готов. Приступим к подготовке клумб.

1. Создайте несколько слоев, на которых вы будете размещать клумбы. Для вставки слоя нажмите кнопку Insert Layer, которая расположена в левом нижнем углу временной шкалы, или вызовите команду Layer меню Insert.
2. Выполните двойной щелчок на значке слоя и введите имя flowerbed1 в поле Name.
3. Повторите процесс дважды, присвоив всем слоям выразительные имена (например, flowerbed2, flowerbed3).

Ниже показано, как временная шкала должна выглядеть на данном этапе.



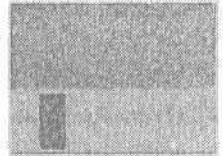
Создание клумб

Займемся непосредственно клумбами:

1. Активизируйте инструмент Rectangle.
2. Чтобы симитировать землю, выберите коричневый цвет в палитре Fill Color (секция Colors панели Tools).
3. Поскольку создаваемые объекты, клумбы, не должны иметь контуров, отмените обводку. Для этого щелкните на элементе Stroke Color (панель Tools). На

экране отобразится палитра. Выполните щелчок на кнопке No Color в правом верхнем углу.

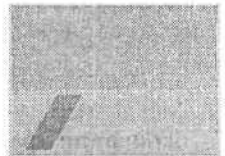
4. Чтобы придать клумбе естественный вид, нажмите кнопку Round Rectangle Radius (Радиус закругления углов прямоугольника). Данная кнопка доступна в секции Options панели Tools, если рабочим является инструмент Rectangle. В открывшемся диалоговом окне Rectangle Settings (Параметры прямоугольника) введите значение 5 в поле Corner Radius,
5. Выберите слой `flowerbed1` и создайте вытянутый по вертикали прямоугольник в зеленой области. Сцена должна выглядеть так, как показано на рисунке.
6. Выделите созданный прямоугольник и конвертируйте его в символ посредством команды Convert to Symbol меню Insert или клавиши F8. В открывшемся диалоговом окне Convert to Symbol введите имя `flowerbed1` в поле Name. Убедитесь, что установлен переключатель Graphic.



Перспектива

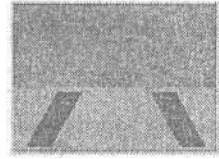
Эффект трехмерного пространства мы создадим за счет трансформации прямоугольника.

1. Выделите прямоугольник, активизируйте инструмент Free Transform и нажмите кнопку Rotate and Skew (Вращение и наклон) в секции Options.
2. Когда появится рамка выделения с маркерами, установите указатель между угловым и боковым маркерами. Указатель мыши приобретет вид двух разнонаправленных стрелок, вертикальных или горизонтальных, что зависит от того, на какой стороне рамки выбран маркер. Нажмите левую кнопку манипулятора и, удерживая ее, перемещайте маркер, пока не получите требуемый результат.
3. Приступая к созданию второй клумбы, выберите слой `flowerbed2`.
4. Откройте окно библиотеки, вызвав команду Library меню Window или нажав клавишу F11.
5. Выберите символ `flowerbed1`. Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Shift (Macintosh) и активизируйте команду Duplicate (Дублировать) контекстного меню.
6. В диалоговом окне Duplicate Symbol введите имя `flowerbed2` в поле Name.
7. Щелкните на кнопке ОК.
8. Убедитесь, что слой `flowerbed2` по-прежнему является текущим. Перетащите символ `flowerbed2` из библиотеки в область действия.
9. Наклоните символ, помещенный на слой `flowerbed2`, таким образом, чтобы он представлял собой зеркальное отражение символа `flowerbed1`. Для этого активизируйте инструмент Free Transform и щелкните на кнопке Rotate and Skew в секции Options. Когда появится рамка выделения с маркерами, установите



указатель мыши между угловым и боковым маркерами. Нажмите левую кнопку мыши и, удерживая ее, перемещайте указатель до получения требуемого результата.

10. После выполнения наклона объекта влево переместите его в левый угол сцены.



Завершение работы над клумбами

Мы разбили клумбы по краям сцены. Теперь добавим еще одну в середину. Чтобы представить рисунок в перспективе, придадим средней клумбе особую форму.

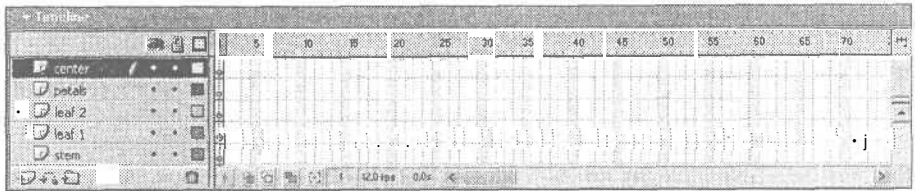
1. Откройте окно библиотеки, вызвав команду Library меню Window или нажав клавишу F11.
2. Выберите символ `flowerbed1`. Активизируйте команду Duplicate контекстного меню.
3. В диалоговом окне Duplicate Symbol введите имя `flowerbed3` в поле Name.
4. Щелкните на кнопке ОК.
5. Оставьте окно библиотеки открытым и выполните двойной щелчок на значке символа `flowerbed3`, чтобы переключиться в режим редактирования.
6. В окне редактора Symbol Editor выделите символ `flowerbed3` с помощью инструмента Arrow.
7. Активизируйте инструмент Free Transform и щелкните на кнопке Distort в секции Options.
8. Удерживая клавишу Shift, установите указатель мыши около маркера, расположенного в правом или левом верхнем углу рамки выделения. Нажмите левую кнопку мыши и, удерживая ее, переместите маркер к центру объекта. По окончании трансформации отпустите кнопку мыши.
9. Продолжайте удерживать клавишу Shift и установите указатель возле маркера, находящегося в правом или левом нижнем углу рамки выделения. Нажмите левую кнопку мыши и, удерживая ее, переместите маркер наружу, чтобы нижняя сторона фигуры стала шире верхней. Отпустите кнопку мыши.
10. На данном этапе символ `flowerbed3` должен выглядеть так, как показано на рисунке.
11. Щелкните на ссылке Scene 1 в строке Scene and Symbol, чтобы вернуться в область действия.
12. Перейдите на слой `flowerbed3`.
13. Перетащите символ `flowerbed3` из библиотеки в область действия и выровняйте его относительно двух других символов. Сцена должна выглядеть так, как показано на рисунке.



Элементы цветов

Клумбы для выращивания цветов готовы, поэтому поработаем над изображениями растений. Мы создадим один цветок (ромашку), скопируем его и трансформируем, чтобы все цветы выглядели по-разному.

1. Перейдите в режим Symbol Editing. Активизируйте команду New Symbol меню Insert или нажмите клавиши Cmd/Ctrl+F8. В диалоговом окне New Symbol введите имя daisy (ромашка) в поле Name. Убедитесь, что установлен переключатель Graphic.
2. Присвойте существующему слою имя stem (стебель).
3. Вставьте дополнительные слои с именами leaf 1, leaf 2, petals и center.
4. На данном этапе временная шкала в режиме редактирования символа должна выглядеть так, как показано на рисунке.



5. Активизируйте инструмент Brush (Кисть) и щелкните на элементе Fill Color в секции Colors. Задайте темно-зеленый цвет. В раскрывающемся списке Brush Size выберите одну из кистей с небольшим диаметром. Данный список доступен в секции Options, когда рабочим является инструмент Brush.

ПРИМЕЧАНИЕ

При использовании графических планшетов (таких как Wacom Intuos 2) в секции Options панели Tools появляется дополнительная кнопка Use Pressure (Нажим), с помощью которой вы можете установить различные стили рисования кистью для графического планшета.

6. Нарисуйте вертикальную линию длиной 2,5 см (1 дюйм).
7. Выберите инструмент Pencil.
8. Перейдите на слой leaf 1 и нарисуйте листок. Убедитесь, что контур замкнут. Укажите зеленый цвет в палитре Fill Color и заполните фигуру с помощью инструмента Paint Bucket.



СОВЕТ

Если вы хотите, чтобы края листка были сглаженными, при создании контура установите режим Smooth в меню Pencil Mode или в секции Options панели Tools (во втором случае предварительно следует активизировать инструмент Pencil).

9. Выделите весь листок (заливку и обводку) и вызовите команду Group меню Modify.

10. Перейдем к созданию второго листка. Выберите существующий листок и скопируйте его посредством команды Copy меню Edit.
11. Сделайте текущим слой leaf 2 и вставьте копию листка с помощью команды Paste меню Edit.
12. Выделите только что вставленный листок и выполните команду Modify ▶ Transform ▶ Flip Horizontal (Модификация/Трансформация/Отобразить по горизонтали).
13. Позиционируйте листки на сцене так, чтобы они казались растущими из стебля, как показано на рисунке. ~\-
14. От стебля с листьями перейдем непосредственно к цветку. Выберите слой center. С помощью инструмента Oval начертите небольшой кружок и переместите его к верхней части стебля. Закрасьте его любым понравившимся цветом. ~\-

ПРИМЕЧАНИЕ

Помните, что все объекты содержат как обводку, так и заливку. Чтобы избежать проблем при перемещении объекта, группируйте указанные элементы посредством команды Group меню Modify.

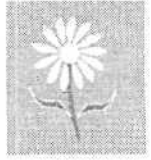
15. Нарисуем лепестки. Выберите слой petals (лепестки) на временной шкале.
16. С помощью инструмента Pencil нарисуйте один лепесток. Убедитесь, что контур замкнут.

СОВЕТ

Чтобы избежать создания лепестков со ступенчатыми краями, установите режим Smooth в меню Pencil Mode или в секции Options панели Tools (во втором случае предварительно следует активизировать инструмент Pencil).

17. Закрасьте лепесток, используя инструмент Paint Bucket. Сцена должна выглядеть так, как показано на рисунке.
18. Нет смысла рисовать каждый лепесток отдельно, поскольку можно «клонировать» уже созданный столько раз, сколько необходимо. Выберите лепесток, а затем скопируйте его с помощью команды Copy меню Edit. Затем вставьте лепесток на тот же слой посредством команды Paste меню Edit. Расположите его справа от первого лепестка таким образом, чтобы один край был прикреплен к центру цветка. +
20. Выберите лепесток с помощью инструмента Arrow, активизируйте инструмент Free Transform и нажмите кнопку Rotate and Skew (Вращение и изгиб) в секции Options.
21. Установите указатель мыши возле одного из угловых маркеров, вследствие чего он приобретет вид круговой стрелки. Нажмите левую кнопку мыши и, удерживая ее, переместите маркер так, чтобы воображаемая ось проходила через центр цветка. +

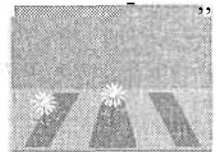
22. Повторите шаги 18–21, пока серединка цветка не будет окружена лепестками.



Посадка цветов

В нашем распоряжении имеются все объекты, необходимые для того, чтобы засадить подготовленные клумбы цветами.

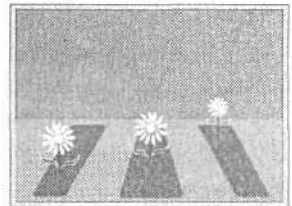
- Щелкните на кнопке Scene 1 в строке Scene and Symbol, чтобы вернуться к самой первой сцене.
- Создайте дополнительный слой и назовите его daisy 1.
- Выберите добавленный слой daisy 1 и перетащите символ daisy из библиотеки на одну из клумб.
- Создайте еще один слой и назовите его daisy 2.
- Перейдите на слой daisy 2 и перетащите символ daisy из библиотеки в область действия.
- Убедитесь, что экземпляр символа выбран, и выполните команду Modify ▶ Transform ▶ Flip Horizontal. Зеркальное отражение объекта поместите па одну из клумб.



В завершение поместим один из цветков на задний план, чтобы придать рисунку ощущение пространства.

- Создайте дополнительный слой и назовите его daisy 3.
- Убедившись, что слой daisy 3 активен, перетащите символ daisy из библиотеки в область действия.
- Выделите добавленный символ с помощью инструмента Arrow.
- Активизируйте инструмент Free Transform и нажмите в секции Option кнопку Scale (Масштабирование).
- Удерживая клавишу Shift, установите указатель около одного из угловых маркеров. При нажатой левой кнопке мыши переместите маркер к центру объекта так, чтобы он уменьшился приблизительно вдвое. По окончании трансформации отпустите кнопку мыши.
- Переместите уменьшенный цветок на одну из клумб, как показано на рисунке.

Вы создали незатейливый садик с тремя цветками. С помощью описанных в данном разделе приемов вы можете сделать изображение более сложным. Например, можно добавить солнышко и другие цветы. Попробуйте, вы получите прекрасный результат.



ЧАСТЬ 3

Создание анимации в среде Flash

Хотя возможности Flash MX позволяют создавать полный спектр превосходных мультимедийных продуктов, в основе своей Flash остается программой для создания анимации. Преимущества векторной графики, на которой базируется Flash, особенно ярко проявляются при создании фильмов. Поэтому рекомендуем вам хорошо изучить все техники, применяемые для «оживления» статических изображений.

Данная часть книги посвящена основам анимации во Flash. Сначала вы научитесь, используя временную шкалу, создавать простые анимационные эффекты. Далее познакомитесь с одними из основных компонентов анимации в среде Flash — клипами. Знания этой темы особенно пригодятся при дальнейшем изучении языка ActionScript. Наконец, освоите наиболее важное средство — сцены, с помощью которых сможете контролировать анимацию, управлять и манипулировать ее.

Глава 9. Анимация с использованием временной шкалы.....	168
Глава 10. Работа с клипами.....	190
Глава 11. Использование сцен при работе с анимационными фильмами.....	201

Глава 9

Анимация с использованием временной шкалы

Когда компания Macromedia купила у компании Future Wave приложение Future-Splash (которое и стало базовым для Flash), оно представляло собой несложную анимационную программу на основе векторной графики. И хотя с развитием Flash добавились сотни возможностей, разработано множество новых инструментов, по своей сути Flash все равно осталась анимационной программой, основная мощь которой проявляется именно при создании фильмов с содержанием, изменяющимся во времени.

До настоящей главы мы не создавали никаких движущихся объектов. И если вам необходимы только статические объекты, то вполне можно обойтись такими программными продуктами, как Adobe Photoshop или Macromedia Fireworks. Но поскольку вы приступили к изучению этой книги, вам требуются объекты, которые смогут двигаться, прыгать и скакать! В данной главе вы познакомитесь с основным инструментом, который используется для создания анимации и манипулирования ее в среде Flash MX, — временной шкалой.

В главе рассматриваются следующие темы:

- ▶ понятие анимации;
- ▶ покадровая анимация;
- ▶ анимация с построением промежуточных изображений;
- ▶ анимация с изменением формы объекта;
- ▶ использование направляющей движения для перемещения объектов вдоль пути;
- ▶ анимирование маски;
- ▶ создание специальных эффектов с использованием панели Properties.

Что такое анимация

Анимация — это процесс изменения размера, положения, цвета или формы объекта во времени. Во Flash существуют два метода создания анимации — *frame by frame* (покадровый метод) и *tweening* (метод автоматического построения промежуточных кадров), каждый из которых имеет свои преимущества. Прежде чем приступить к изучению этих методов, необходимо познакомиться с основными понятиями кадра, ключевого кадра и частотой смены кадров, используемыми при создании фильма.

Кадры и ключевые кадры

Все фильмы, будь это современные голливудские проекты, первые мультфильмы студии Уолта Диснея или фильмы Flash, состоят из *кадров*.

Каждый кадр содержит одно статическое изображение. При последовательном просмотре таких изображений создается иллюзия движения. Во Flash кадры представлены в виде небольших прямоугольников, которые называются ячейками. Они горизонтально располагаются на временной шкале. Используя команды меню Insert, любую ячейку можно преобразовать в кадр, ключевой кадр или пустой ключевой кадр.

При прохождении кадра головкой воспроизведения его содержимое выводится на сцене, таким образом последовательно выводятся изображения всех кадров. Во Flash кадр, содержащий информацию любого типа, созданную непосредственно разработчиком (в отличие от автоматически создаваемого содержимого промежуточных кадров), отмечен на временной шкале небольшим черным кружком и называется *ключевым кадром*. Каждый раз, когда необходимо сделать изменения в анимации (добавить содержимое, удалить часть содержимого, начать движение объекта и так далее), используется ключевой кадр.

Таким образом, различие между кадром и ключевым кадром состоит в том, что даже в случае идентичности их содержимого непосредственно манипулировать вы можете только содержимым ключевого кадра. Для изменения содержимого кадра необходимо его преобразовать в ключевой.

Кроме двух названных типов кадров существует еще *пустой ключевой кадр* — ключевой кадр, не заполненный содержимым. На временной шкале он представлен в виде маленького полого кружка.

ПРИМЕЧАНИЕ

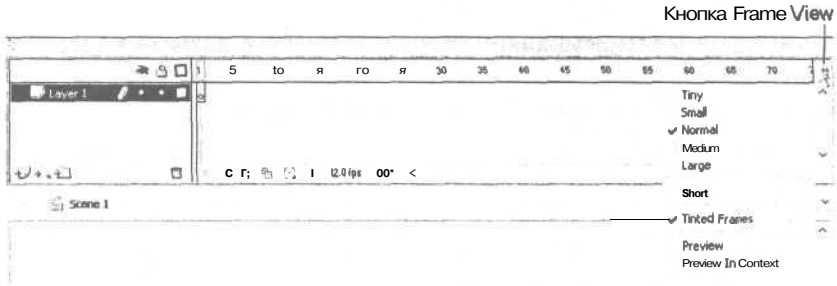
По умолчанию первый кадр нового Flash-фильма создается как пустой ключевой кадр. Все остальные кадры любого заданного слоя представляют собой заполнители для кадров и ключевых кадров, создаваемых в процессе работы.



Изменения внешнего вида кадров на временной шкале

Пока вы знакомы только с одним, рассмотренным в главе 8, способом модификации внешнего вида кадров на временной шкале — посредством изменения высоты отображения всего слоя. При этом фактически меняется внешний вид самого слоя, а не кадров. Кроме того, как вы знаете, существует всего три варианта выбора высоты отображения слоя.

Но в среде Flash MX существует возможность изменять внешний вид и самих кадров. Это осуществляется с помощью кнопки Frame View, расположенной в правом верхнем углу временной шкалы. При щелчке на данной кнопке открывается меню, которое содержит команды для модификации внешнего вида кадров.

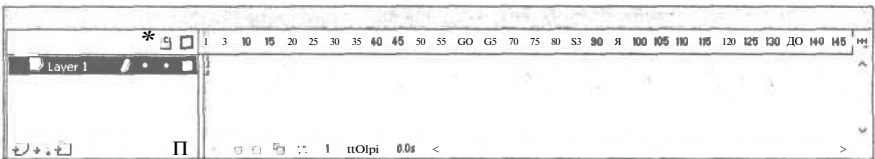


ПРИМЕЧАНИЕ

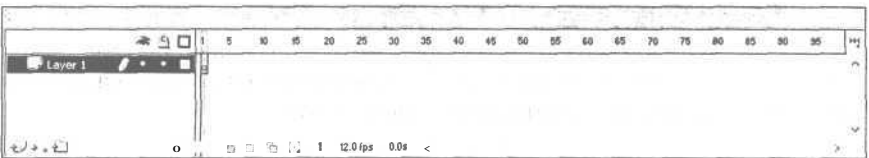
К сожалению, при выборе одной из команд меню кнопки Frame View внешний вид устанавливается для кадров сразу всех слоев временной шкалы. Внешний вид для кадров отдельного слоя с помощью этой кнопки установить нельзя.

Ниже перечислены команды меню Frame View и соответствующие им действия.

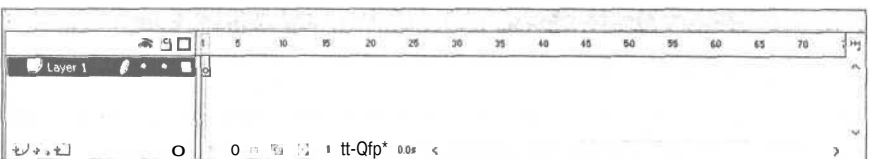
- ▶ Tiny — устанавливает минимальный размер кадров, что позволяет одновременно просматривать большее количество кадров на временной шкале.



- ▶ Small — устанавливает средний размер кадров между наименьшим и размером по умолчанию.



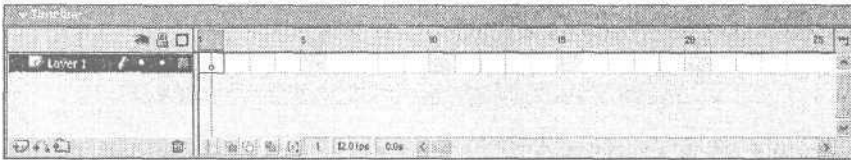
- ▶ Normal — размер кадров, устанавливаемый по умолчанию.



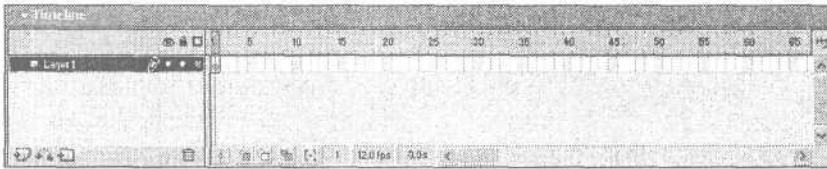
- ▶ **Medium** — устанавливает размер кадров, немного **большой**, чем по умолчанию.



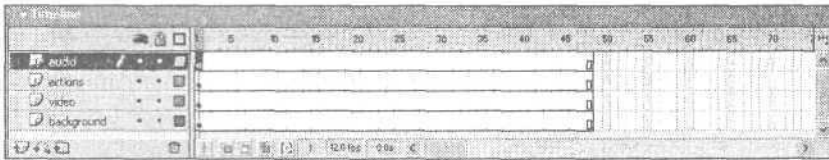
- ▶ **Large** — устанавливает **максимальный** размер кадров.



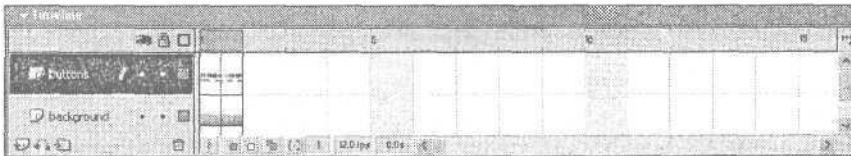
- ▶ **Short** — уменьшает **высоту** кадров на временной шкале.



- ▶ **Tinted Frames** — создает **закрашивание** серым цветом последовательных кадров фильма. При выключении команды последовательные кадры **вновь** становятся белыми.



- ▶ **Preview in Context** — создает **уменьшенное** изображение каждого **ключевого** кадра, **прямо пропорционально** отображающее изображение на области действия, **включая** все **незаполненное** пространство.



- ▶ **Preview** — создает **уменьшенное** изображение каждого **ключевого** кадра, **включающее** только **часть** области действия с **изображением** объектов. **Незаполненное** пространство в **уменьшенное** изображение **не** включается.

ПРИМЕЧАНИЕ-

Поскольку при выборе команды Preview in Context или Preview на временной шкале выводится дополнительная информация, скорость работы компьютера может снижаться.

Представление анимации на временной шкале

Различные типы анимации представлены на временной шкале по-разному. Чтобы не допускать ошибок при работе над реальной проектом, вы должны научиться определять тип анимации по ее внешнему виду.

- ▶ Анимация типа *frame by frame* (покадровая) обычно представлена в виде слоя с серией последовательных ключевых кадров.



- ▶ Анимация типа *motion tweening* (с автоматическим построением промежуточных кадров при перемещении объекта в пределах сцены или при изменении характеристик объекта) представлена в виде ключевых кадров в начале и конце временной шкалы, между которыми расположена черная стрелка, представляющая промежуточные кадры. Кроме того, промежуточные кадры имеют светло-голубую окраску.



- ▶ Анимация типа *shape tweening* (с автоматическим построением промежуточных кадров при преобразовании одного объекта в другой), как и предыдущий тип анимации, представлена в виде ключевых кадров в начале и конце временной шкалы, между которыми расположена черная стрелка, представляющая промежуточные кадры. Но в этом случае промежуточные кадры имеют светло-серую окраску.
- ▶ Если за ключевым кадром следует пунктирная линия, это означает, что анимация с построением промежуточных кадров является незавершенной, поскольку отсутствует конечный ключевой кадр.



- ▶ Если за ключевым кадром следует серия серых кадров, а в конце располагается пустой прямоугольник, это означает, что все кадры после ключевого имеют одинаковое содержимое.



- ▶ Наличие небольшой строчной буквы *a* в кадре или ключевом кадре говорит о том, что в данной точке фильма добавлено действие. (Более подробно о действиях и языке *ActionScript* будет рассказано в частях 4 и 5 данной книги.)

Кадр действия



- Красный флажок в кадре или в ключевом кадре является меткой кадра.



СОВЕТ

Метки кадров особенно полезны при действиях, требующих идентификации и указания определенных кадров (то есть вместо использования номеров кадров). Метки также можно применять, чтобы добавить комментарии к нужным кадрам в процессе работы над проектом. Для установки метки выберите кадр, откройте панель Properties и введите имя метки в поле Frame Label.

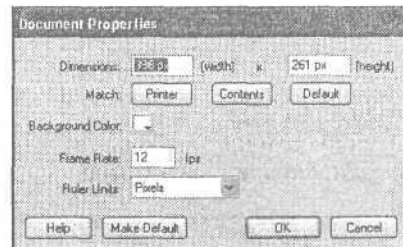
Частота смены кадров

Частота смены кадров определяет скорость воспроизведения фильма. Данная величина измеряется в количестве кадров в секунду (*fps*) (от англ. frames per second — кадров в секунду) и обуславливает как скорость воспроизведения фильма, так и его качество. Чем выше это значение, тем больше кадров воспроизводится за одну секунду, следовательно, тем лучше качество фильма.

На первый взгляд кажется, что следует всегда задавать высокое значение частоты смены кадров, поскольку это гарантирует хорошее качество фильма. Но следует учитывать, что чем выше значение частоты смены кадров, тем больше ресурсов памяти необходимо и тем шире частотная полоса при передаче такого фильма в сети. Когда установлено высокое значение частоты смены кадров, ваш компьютер вынужден выводить больше информации за одну секунду. Если же скорость передачи данных в сети низкая или компьютер не обладает необходимыми возможностями, данные просто могут быть потеряны, а это приведет к зависанию, пропускам или полной остановке воспроизведения фильма. Поэтому ваша задача заключается в нахождении оптимального соотношения между скоростью и качеством воспроизведения.

Чтобы установить значение частоты смены кадров, выполните следующие действия.

1. Откройте диалоговое окно Document Properties (Свойства документа), для чего выберите команду Document меню Modify или воспользуйтесь комбинацией клавиш **Cmd/Ctrl+J**.
2. Введите значение частоты смены кадров в поле Frame Rate.



ПРИМЕЧАНИЕ

По умолчанию для фильмов Flash установлено значение частоты смены кадров 12 fps. Для улучшения качества фильма вы можете увеличить данное значение. Однако не рекомендуется устанавливать его выше 20 fps, кроме тех случаев, когда вы полностью уверены, что компьютер пользователя корректно воспроизведет данный фильм.

Создание покадровой анимации

До изобретения компьютера художникам-мультипликаторам приходилось рисовать каждый кадр вручную, при этом каждый **новый** кадр должен был несколько отличаться от предыдущего, чтобы при последовательном просмотре создавалась иллюзия движения.

Во Flash работа наиболее простой покадровой анимации основывается на том же принципе. По существу каждый кадр в этом случае является **ключевым**, поскольку его содержимое изменяет анимацию. При прохождении каждого отдельного кадра **головкой воспроизведения** его содержимое выводится на сцене, что создает иллюзию изменения во времени. Покадровая анимация применяется в тех случаях, когда вам необходим полный контроль над изображением каждой детали вашего фильма.

Однако создание покадровой анимации требует больших временных затрат, поскольку каждый кадр должен быть «заполнен» уникальным содержимым. Например, если фильм содержит 300 кадров, вам потребуется создать как минимум 300 статических изображений, а это достаточно трудоемкий процесс. Конечно, Flash позволяет использовать библиотеки, символы и экземпляры символов, что значительно облегчает труд, но в любом случае такая работа отнимет у вас много времени.

Для освоения основных приемов создания покадровой анимации рассмотрим простой пример, в котором сфера будет перемещаться в пределах области действия.

1. Создайте новый файл, для чего выберите команду New меню File или используйте комбинацию клавиш **Cmd/Ctrl+N**.
2. Выберите команду New Symbol меню Insert. В открывшемся диалоговом окне Create New Symbol (Создание нового символа), введите имя sphere в поле Name. Убедитесь, что переключатель типа символа установлен в положение Graphic (Графический), и щелкните на кнопке ОК. После перехода в режим редактирования символа создайте сферу диаметром около 3 сантиметров. Затем возвратитесь к сцене, щелкнув мышью на надписи Scene 1, расположенной в строке Scene and Symbol.
3. Откройте окно библиотеки, которая содержит символ с изображением сферы, выбрав команду Library меню Window. Щелкните на первом кадре слоя, который по умолчанию создается как пустой ключевой кадр, и переместите символ из библиотеки в левую часть области действия. Обратите внимание, что первый кадр автоматически превратился в ключевой кадр (обозначен **небольшим черным кружком**).

ПРИМЕЧАНИЕ —

Помните, что Flash автоматически создает первый кадр как пустой ключевой кадр. Но все остальные необходимые ключевые кадры вам придется **вставлять** вручную.

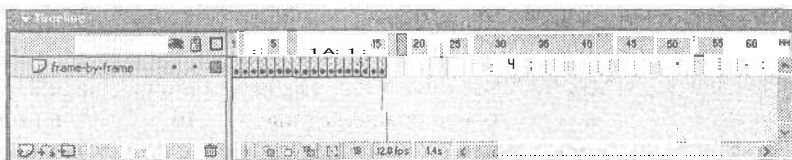
4. Теперь, когда первый ключевой кадр содержит изображение, можно создавать оставшуюся часть последовательности. Щелкните на втором кадре временной шкалы (справа от только что созданного ключевого кадра) и выберите команду

Keyframe меню Insert или нажмите клавишу F6. При этом будет вставлен второй ключевой кадр. Обратите внимание, что Flash автоматически поместит содержимое первого ключевого кадра во второй ключевой кадр. Вы можете продолжать последовательно вставлять ключевые кадры, куда Flash будет автоматически помещать содержимое первого ключевого кадра.

ПРИМЕЧАНИЕ

Если по каким-либо причинам вы не хотите, чтобы содержимое первого ключевого кадра помещалось в создаваемые ключевые кадры, произведите вставку пустых ключевых кадров посредством выбора команды Blank Keyframe меню Insert. Для добавления содержимого в ключевой кадр выберите его и создайте какой-либо объект в рабочей области. При этом пустой ключевой кадр автоматически превратится в ключевой.

5. Убедитесь, что выбран второй ключевой кадр. С помощью инструмента Arrow выберите сферу и немного переместите ее вправо.
6. Вставьте третий ключевой кадр.
7. Убедитесь, что выбран третий ключевой кадр. С помощью инструмента Arrow выберите сферу и вновь немного переместите ее вправо.
8. Продолжайте последовательно вставлять ключевые кадры, перемещая сферу вправо в каждом последующем из них до тех пор, пока она не достигнет правого края области действия. Количество созданных кадров зависит от того, на какое расстояние вы перемещали сферу в каждом новом кадре. Чем меньше это расстояние, тем больше кадров будет содержаться на временной шкале.



Поздравляем! Вы создали свою первую покадровую анимацию. Для ее тестирования **выполните** одно из следующих действий:

- ▶ с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
- ▶ выберите команду Play меню Control, либо нажмите клавишу Return/Enter;
- ▶ выберите команду Test Scene или Test Movie меню Control. При этом откроется новое окно, в котором ваш фильм будет воспроизводиться циклически.

Работа в режиме onion skinning

В процессе создания покадровой анимации, описанном ранее, вы перемещали сферу, не зная ее точного положения в предыдущем кадре и, следовательно, не могли позиционировать сферу так, чтобы она двигалась точно по прямой линии. Для решения данной проблемы существует режим onion skinning (полупрозрачное представление содержимого кадров).

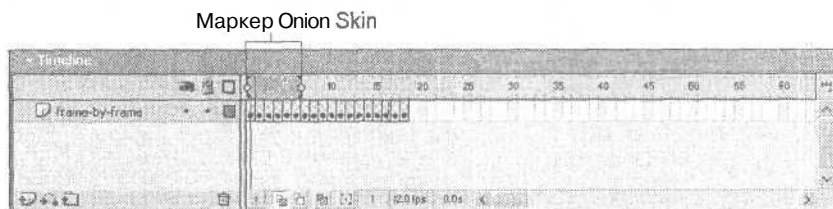
Этот режим позволяет видеть содержимое кадров, расположенных до и после выделенного кадра. Такая техника напоминает использование прозрачной кальки для просмотра последовательных кадров анимации. Режим onion skinning позволяет освободиться от кропотливой работы, когда необходимо возвращаться к предыдущему кадру, запоминать расположение объектов, а затем, перейдя к текущему кадру, перемещать объекты в требуемое положение.

Существует несколько типов режима полупрозрачного представления содержимого кадров.

Включение режима полупрозрачного представления содержимого кадров

Чтобы включить режим полупрозрачного представления содержимого кадров, щелкните на кнопке Onion Skin, которая расположена в нижней части временной шкалы.

При этом происходит следующее. Во-первых, на линейке временной шкалы появляется маркер Onion Skin, который выводит содержимое кадров, включенных в режим полупрозрачного представления. Увеличить или уменьшить число включенных кадров позволяют метки-манипуляторы, расположенные с обеих сторон этого маркера. Установите указатель мыши около одной из меток-манипуляторов, расположенной справа или слева от маркера Onion Skin. Нажмите левую кнопку мыши и, удерживая ее, перемещайте маркер до получения требуемого результата.



Во-вторых, содержимое кадров, включенных маркером Onion Skin, выводится как полупрозрачное. Содержимое выбранного в настоящий момент кадра выводится в обычном режиме.



ПРИМЕЧАНИЕ

Если в вашем фильме имеются заблокированные слои, их содержимое в режиме onion skinning выводиться не будет.

Режим onion skinning outlines

Режим полупрозрачного контурного представления содержимого кадров можно включить, щелкнув на кнопке Onion Skinning Outlines, которая расположена в нижней части временной шкалы. Обратите внимание, что при включении этого режима содержимое выбранного в настоящий момент кадра выводится в обычном

режиме. Цвет объектов при включении режима `onion skinning outlines` зависит от цвета контурного представления объектов данного слоя.



Редактирование множества кадров

При включенном режиме полупрозрачного представления объектов вы не можете редактировать объекты кадра до тех пор, пока данный кадр не будет выбран на временной шкале. Это несколько затрудняет работу, когда вам нужен быстрый доступ к объектам различных кадров.

Кнопка `Edit Multiple Frames`, расположенная справа от кнопки `Onion Skinning Outlines`, позволяет редактировать содержимое всех кадров без необходимости перехода от одного кадра к другому. Следует отметить, что в режиме редактирования содержимого множества кадров все объекты выводятся как непрозрачные.



Меню `Modify Onion Markers`

Чтобы открыть меню `Modify Onion Markers`, нажмите кнопку `Modify Onion Markers`, расположенную справа от кнопки `Edit Multiple Frames`. С помощью данного меню можно установить положение маркеров `Onion Skin`, а следовательно, определить число кадров, выводимых в режиме полупрозрачного представления их содержимого.

Далее перечислены команды данного меню и соответствующие им действия.

- ▶ При выборе команды `Always Show Markers` (Всегда показывать маркеры) маркеры `Onion Skin` будут видны на линейке временной шкалы независимо от того, включен режим `onion skin` или нет.
- ▶ Число представленных на сцене кадров в режиме `onion skin` зависит от текущего положения головки воспроизведения и положения маркеров `Onion Skin`. Выбрав команду `Anchor Onion`, вы можете зафиксировать текущее положение маркеров `Onion Skin` на линейке временной шкалы.
- ▶ При выборе команды `Onion 2` будет выведено содержимое двух кадров с каждой стороны от выбранного в настоящий момент кадра.
- ▶ При выборе команды `Onion 5` будет выведено содержимое пяти кадров с каждой стороны от выбранного в настоящий момент кадра.
- ▶ При выборе команды `Onion All` будет выведено содержимое всех кадров с каждой стороны от выбранного в настоящий момент кадра.

Автоматическое построение промежуточных кадров

Теперь, когда вы научились создавать покадровую анимацию, рассмотрим более эффективный процесс создания фильма — автоматическое построение промежуточных кадров. В его основе лежит создание ключевых кадров, которые в

дальнейшем служат опорными точками анимации. При этом все промежуточные кадры Flash формирует автоматически.

Существуют два типа анимации с автоматическим построением промежуточных кадров: *motion tweening* и *shape tweening*. Оба типа будут рассмотрены в следующих разделах данной главы. Затем мы расскажем о способе определения пути следования объекта при создании анимации типа *motion tweening*.

Создание анимации типа *motion tweening*

Из предыдущего раздела вы узнали, как создавать покадровую анимацию посредством «заполнения» каждого кадра содержимым. Данный процесс очень трудоемок, поскольку каждый кадр делается вручную. Значительно ускорить процесс создания анимации позволяет технология *motion tweening*. При работе с анимацией этого типа достаточно создать первый и последний ключевые кадры, а все промежуточные будут созданы автоматически. Данная техника имеет еще одно существенное преимущество. Поскольку Flash сохраняет только данные о первом и последнем ключевом кадре (а также некоторые числовые значения, касающиеся способа преобразования объекта), размер конечного файла будет заметно меньшим по сравнению с покадровой анимацией.

В основном анимация типа *motion tweening* используется для создания эффекта движения объекта, однако с ее помощью можно изменять также размер, цвет и ориентацию объекта на сцене. Обобщенно можно сказать, что любая модификация объекта может быть выполнена с использованием техники *tweening*.

ПРИМЕЧАНИЕ

Тип *motion tweening* может применяться для анимации текста, а также группы объектов.

В этом разделе мы рассмотрим использование техники *tweening* для создания несложной анимации, которую уже умеете создавать покадровым способом. Вы научитесь перемещать квадрат вдоль области действия.

1. Создайте новый файл, выбрав команду *New* меню *File*, или используйте комбинацию клавиш *Cmd/Ctrl+N*.
2. Выберите команду *New Symbol* меню *Insert*, после чего откроется диалоговое окно *Create New Symbol*. Укажите имя *square* в поле *Name*. Убедитесь, что переключатель типа символа установлен в положение *Graphic*, и щелкните на кнопке *ОК*. После перехода в режим редактирования символа создайте квадрат с размером сторон около 3 сантиметров. Закончив, возвратитесь к сцене, щелкнув мышью на надписи *Scene 1*, расположенной в строке *Scene and Symbol*.
3. Откройте окно библиотеки, которая уже содержит символ с изображением квадрата, выбрав команду *Library* меню *Window*. Щелкните на первом кадре слоя и переместите символ из библиотеки в левую часть области действия. Обратите внимание, что первый кадр, который по умолчанию создается как пустой ключевой кадр, автоматически превратился в ключевой кадр (обозначен небольшим черным кружком).

4. Теперь, когда первый ключевой кадр заполнен содержимым, можно приступить к созданию последнего ключевого кадра. Щелкните на кадре с номером 25 на временной шкале и произведите вставку ключевого кадра, выбрав команду Keyframe меню Insert (или нажмите клавишу F6). На данном этапе временная шкала должна выглядеть так, как показано на следующем рисунке.



ПРИМЕЧАНИЕ —

Кадры, автоматически создаваемые между ключевыми кадрами, являются простыми. Они служат только для хранения содержимого, автоматически генерируемого Flash. Если вы хотите отредактировать содержимое такого кадра, преобразуйте его в ключевой кадр.

Как и в случае с покадровой анимацией, Flash автоматически вставляет содержимое первого ключевого кадра в создаваемый второй ключевой кадр. Таким образом, вам необходимо выбрать вновь созданный ключевой кадр и перетащить квадрат в то место области действия, куда он должен переместиться в конце фильма.

1. Выберите последний ключевой кадр.
2. Используя инструмент Arrow, выберите квадрат и переместите его в правую часть области действия, таким образом указав Flash, что в первом кадре квадрат должен находиться в левой части области действия, а в последнем кадре — в правой. Теперь все готово для построения промежуточных кадров.
3. Выберите первый ключевой кадр.
4. Щелкните на команде Create Motion Tween меню Insert. В качестве альтернативы можно выбрать первый ключевой кадр, открыть панель Properties (посредством команды Properties меню Window) в раскрывающемся списке Tween и щелкнуть на пункте Motion.
5. Flash автоматически заполнит содержимым все промежуточные кадры. Обратите внимание, что цвет кадров на временной шкале изменился, а также появилась черная стрелка между первым и последним ключевым кадром, указывающая на наличие автоматически созданных промежуточных кадров.



Поздравляем! Теперь вы создали первую анимацию с автоматическим построением промежуточных кадров. Чтобы проверить ее, выполните одно из следующих действий:

- ▶ с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
- ▶ выберите команду Play меню Control либо нажмите клавишу Return/Enter;
- ▶ выберите команду Test Scene или Test Movie меню Control. При этом откроется новое окно, в котором ваш фильм будет воспроизводиться циклически.

Процесс создания анимации с автоматическим построением промежуточных кадров достаточно прост, но не думайте, что таким способом можно делать только несложную анимацию. Напротив, использование этой техники позволяет создавать действительно сложные эффекты движения. В качестве эксперимента попробуйте вставить ключевой кадр в середине только что созданного фильма и переместите объект в новое положение. При воспроизведении объект изменит траекторию движения, пройдет положение, указанное в среднем ключевом кадре, а затем переместится в конечную точку.

Теперь с помощью панели Transform или инструмента Free Transform попробуйте в последнем ключевом кадре изменить размер объекта. В результате этот объект будет не только двигаться, но и постепенно изменится в размерах.

Используя технику создания анимации с автоматическим построением промежуточных кадров, можно не только перемещать объект в пределах области действия, но также менять его размер и другие характеристики. Эти возможности будут подробно рассмотрены в разделе «Создание эффектов анимации с помощью панели Properties» данной главы.

Использование панели Properties для правки анимации с автоматическим построением промежуточных изображений

При создании анимации с автоматическим построением промежуточных изображений многими характеристиками анимации можно управлять с помощью панели Properties.

Некоторые установки на этой панели предназначены для других типов анимации, таких как shape tweening и motion path tweening. Данные типы анимации будут рассмотрены в разделах «Создание анимации типа shape tweening» и «Создание анимации типа motion path tweening» этой главы. Но сначала рассмотрим установки, предназначенные для анимации типа motion tweening.



Раскрывающийся список Rotate Поле Rotate Times

Замедление движения объекта при автоматическом создании промежуточных кадров

Вероятно, вы заметили, что в предыдущем примере квадрат двигался вдоль области действия с постоянной скоростью. Используя установки панели Properties, можно замедлить ход объекта в начале или в конце движения. То есть объект может начинать движение медленно, а потом увеличивать скорость к концу анимации, либо быстро перемещаться в начале, а затем замедлять движение.

Чтобы при автоматическом создании промежуточных кадров замедлить ход объекта, выполните следующие действия.

1. Выберите первый кадр анимации.
2. Откройте панель Properties, выбрав команду Properties меню Window, либо воспользуйтесь комбинацией клавиш `Cmd/Ctrl+F3`.
3. Введите значение в поле Ease (Замедление хода). Положительное значение замедляет ход объекта в конце движения, а отрицательное — в начале. Чем выше значение (независимо от знака), тем заметнее эффект. Для установки значения Ease можно также воспользоваться бегунком. Допустимые значения Ease выбираются из диапазона от -100 до 100.

Вращение объекта при автоматическом создании промежуточных кадров

Чтобы задать вращение объекта при перемещении, выполните следующие действия.

1. Выберите первый кадр анимации.
2. Откройте панель Properties, выбрав команду Properties меню Window, либо воспользуйтесь комбинацией клавиш `Cmd/Ctrl+F3`.
3. Откройте раскрывающийся список Rotate (Вращение) и щелкните на одном из пунктов:
 - ▷ None. При выборе данной команды объект не вращается;
 - ▷ Auto. Объект поворачивается один раз в направлении наименьшего перемещения;
 - ▷ CW. Вращение объекта по часовой стрелке. Число вращений определяется значением, которое введено в поле Times (справа от раскрывающегося списка Rotate);
 - ▷ CCW. Вращение объекта против часовой стрелки. Число вращений определяется значением, которое введено в поле Times.

Создание анимации типа shape-tweening

Если анимация типа *motion tweening* используется для создания эффекта движения объекта из одной точки в другую или для изменения характеристик объекта (размера, ориентации и т. д.), то анимация типа *shape tweening* применяется для преобразования одного объекта в другой. Таким способом можно, создавать удивительные эффекты.

ПРИМЕЧАНИЕ

Необходимо запомнить, что с помощью анимации типа *shape-tweening* можно преобразовывать только те объекты, которые созданы на области действия с помощью инструментов рисования и черчения. Используя данную технику, нельзя преобразовывать экземпляры символов, растровые рисунки, текст или группы объектов. Все эти элементы необходимо сначала разбить на части посредством команды Break Apart меню Modify, а уже затем использовать анимацию типа *shape-tweening*.

Для создания анимации типа *shape-tweening* выполните следующие действия.

1. Создайте новый файл, выбрав команду **New** меню **File**, или воспользуйтесь комбинацией клавиш **Cmd/Ctrl+N**.
2. Выберите первый кадр, который по умолчанию является пустым ключевым кадром.
3. С помощью инструментов для рисования и черчения создайте изображение начального объекта. Обратите внимание, что первый кадр автоматически превратился в ключевой кадр (обозначен небольшим черным кружком).
4. Выберите кадр, в котором преобразование объекта должно завершиться, и произведите вставку ключевого кадра. Как обычно, Flash поместит содержимое первого ключевого кадра в создаваемый ключевой кадр. Затем с помощью инструментов трансформации измените форму объекта. В качестве альтернативы можно удалить изображение и создать совершенно новую форму.

ПРИМЕЧАНИЕ

Если изменить положение объекта в последнем кадре, Flash автоматически создаст промежуточные кадры движения объекта. Это значит, что объект будет не только преобразовываться из одной формы в другую, но и перемещаться в пределах области действия.

5. Выберите первый кадр и откройте панель **Properties** посредством команды **Properties** меню **Window**.
6. Щелкните на раскрывающемся списке **Tween** и выберите пункт **Shape**.
7. На данном этапе вы можете определить некоторые характеристики преобразования объекта:
 - > введите значение в поле **Ease**. Помните, что при положительном значении процесс преобразования объекта происходит быстрее в начале, а при отрицательном — в конце. Кроме того, для установки значения **Ease** можно использовать бегунок;
 - > в раскрывающемся списке **Blend** установите характеристику преобразования объекта. Если выбрать пункт **Distribute**, все края промежуточных форм будут сглажены, а при выборе пункта **Angular** все острые углы объекта в промежуточных кадрах будут сохранены.



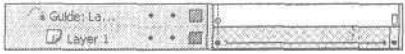


8. После завершения протестируйте анимацию, выполнив одно из следующих действий:
 - > с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
 - > выберите команду **Play** меню **Control** либо нажмите клавишу **Return/Enter**;
 - > щелкните на команде **Test Scene** или **Test Movie** меню **Control**.

Создание анимации типа motion path tweening

При создании анимации типа motion tweening объект движется по прямой линии от одной точки области действия к другой. Добавляя промежуточные ключевые кадры, можно заставить объект двигаться зигзагообразно. Если же необходимо, чтобы объект двигался по кругу, используется тип анимации motion path tweening, в котором можно создавать специальные пути для перемещения объекта.

Из главы 8 вы уже знаете, как создавать специальный слой, называемый направляющей движения (motion guide). Задача, стоящая перед вами в этом разделе, — научиться совместно использовать навыки создания направляющей движения и анимации типа motion tweening, чтобы создавать еще один тип анимации, в которой объект перемещается по сложному заранее определенному пути (траектории).

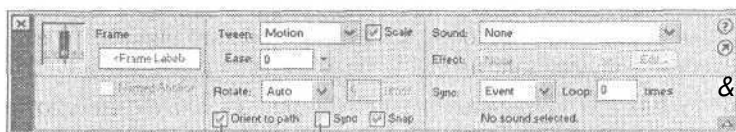
Для создания такой анимации выполните следующие действия.

1. Создайте простую анимацию типа motion tweening с двумя ключевыми кадрами, в начале и в конце. (Можно использовать ранее созданную анимацию с перемещением квадрата.) Если вы забыли, как это делается, вернитесь к разделу «Создание анимации типа motion tweening».
2. Выберите слой, содержащий кадры с анимацией типа motion tweening, а затем добавьте направляющую движения, выбрав команду Motion Guide меню Insert. В качестве альтернативы щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите пункт Add Motion Guide в раскрывшемся контекстном меню. Направляющую движения также можно создать посредством щелчка на кнопке Add Motion Guide, расположенной в левом нижнем углу временной шкалы. При этом будет создан слой направляющей движения, который разместится выше слоя, содержащего кадры анимации.
 
3. Выберите первый кадр созданного слоя направляющей движения и с помощью инструментов рисования создайте траекторию перемещения, по которой должен следовать объект. Можно использовать инструменты Pencil, Oval, Rectangle, Brush, Pen или Line.
 
4. Теперь свяжите объект с созданной траекторией движения. Выберите первый кадр анимации, откройте панель Properties, используя команду Properties меню Window, и убедитесь, что флажок Snap (Привязка) отмечен. Это необходимо для привязки точки регистрации объекта к траектории движения.
5. С помощью инструмента Arrow выберите объект и переместите его таким образом, чтобы точка регистрации объекта совпала с начальной точкой траектории движения.
 
6. Выберите последний ключевой кадр анимации. С помощью инструмента Arrow выберите объект и переместите его таким образом, чтобы точка регистрации объекта совпала с конечной точкой траектории движения. Если при перемещении объекта точка регистрации не появилась, нажмите левую кнопку мыши и, прежде чем переместить объект к точке окончания пути движения, удерживайте ее несколько секунд.

СОВЕТ

Если при выборе объекта точка регистрации не появляется, отпустите кнопку мыши и переместите указатель к центру объекта. Затем вновь нажмите левую кнопку мыши и удерживайте ее несколько секунд, потом переместите объект к точке окончания пути движения.

7. Если вы хотите, чтобы объект при перемещении менял ориентацию в соответствии с траекторией движения, отметьте флажок **Orient to Path** на панели **Properties**.



Флажок **Orient to path** Флажок **Sync**


8. По завершении работы протестируете анимацию, выполнив одно из следующих действий:
 - ▷ с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
 - ▷ выберите команду **Play** меню **Control** либо нажмите клавишу **Return/Enter**;
 - ▷ выберите команду **Test Scene** или **Test Movie** меню **Control**.

СОВЕТ

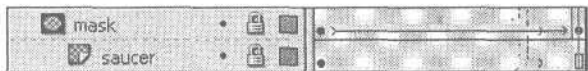
Чтобы отменить связь слоя с направляющей движения, выберите слой и перетащите его с помощью мыши выше слоя направляющей движения на временной шкале. В качестве альтернативы выберите слой и посредством щелчка мыши на команде **Layer** меню **Modify** откройте диалоговое окно **Layer Properties**. В разделе **Type** этого окна выберите пункт **Normal**.

Анимирование маски

Как вы знаете из материала главы 8, слои-маски применяются для управления видимостью содержимого одного из слоев, расположенных ниже, чем он. Слои-маски можно анимировать, что позволяет создавать ряд интересных эффектов, имитирующих, например, просмотр изображения в телескоп или через замочную скважину. Изучив этот раздел, вы научитесь создавать эффект луча прожектора, который перемещается вдоль области действия и освещает неподвижное изображение объекта.

1. Создайте новый документ.
2.  Выберите команду **Import** меню **File**. Когда откроется диалоговое окно **Import**, найдите каталог **Chapter 9** на прилагаемом CD. Откройте список **Files of Type** и выберите пункт **All Files (Все файлы)**. Выберите файл **flying_saucer.swf** и нажмите кнопку **OK**.

3. Далее щелчком на кадре с номером 25 выберите команду Frame меню Insert. При этом содержимое первого ключевого кадра будет автоматически вставлено в кадры с номерами от 2 до 25.
 4. На данном этапе вам необходимо создать объект, который будет использоваться в качестве маски. Выберите команду New Symbol меню Insert. Когда появится диалоговое окно Create New Symbol (Создать новый символ), введите имя `circle` в поле Name. Убедитесь также, что переключатель типа символа установлен в положение Graphic.
 5. После переключения в режим редактирования символа создайте окружность диаметром около 3 сантиметров. Помните, для того чтобы маска работала, создаваемый на слое-маске объект должен иметь заливку. Цвет заливки в данном случае значения не имеет.
 6. Завершите работу в окне редактора символов, щелкнув мышью на надписи Scene 1 в строке Scene and Symbol.
 7. Создайте новый слой, щелкнув на кнопке Insert Layer (Вставить слой), расположенной на временной шкале. В качестве альтернативы щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите команду Insert Layer в раскрывшемся контекстном меню либо команду Layer меню Insert. Созданный слой в дальнейшем будет преобразован в слой-маску.
 8. Выберите первый пустой ключевой кадр созданного слоя и с помощью указателя мыши перетащите символ из окна библиотеки на область действия ближе к левому краю.
 9. Щелкните на кадре с номером 25 созданного слоя и выберите команду Keyframe меню Insert. Перетащите изображение окружности ближе к правому краю области действия.
 10. Выберите первый ключевой кадр. Откройте меню Insert и выберите команду Create Motion Tween. Будет создана анимация, в которой окружность перемещается вдоль области действия и проходит над изображением летающей тарелки, импортированным ранее.
- И. На данном этапе необходимо преобразовать изображение окружности в маску. Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) на имени слоя, содержащем изображение окружности, и выберите команду Mask в контекстном меню. При этом слой с изображением окружности будет преобразован в слой-маску. Обратите внимание, что внешний вид обоих слоев на временной шкале изменился. Они стали взаимосвязанными и, кроме того, заблокированными и должны оставаться таковыми для нормальной работы анимации.



ПРИМЕЧАНИЕ

Пусть вас не беспокоит, что на области действия исчезло изображение летающей тарелки, — так и должно было произойти при блокировке слоев. Если вам необходимо видеть изображение обоих слоев, их необходимо разблокировать.

12. После завершения протестируете анимацию, выполнив одно из следующих действий:
- > с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
 - > выберите команду Play меню Control либо нажмите клавишу Return/Enter;
 - > выберите команду Test Scene или Test Movie меню Control.

Остановка движения объекта во время анимации

До этого момента все эффекты анимации вы создавали с помощью временной шкалы. Однако существуют и другие способы их создания, хотя и простые, но достаточно эффективные. Например, можно вставить в некоторые кадры статическое изображение, таким образом, в течение определенного промежутка времени будет наблюдаться задержка движения на области действия. Вот как это сделать.

1. Создайте новый документ.
2. Выберите первый пустой ключевой кадр.
3. Поместите на область действия какое-либо изображение.
4. Щелкните на последнем кадре.
5. Выберите команду Frame меню Insert или нажмите клавишу F5. В качестве альтернативы щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите команду Insert Frame в раскрывшемся контекстном меню.
6. Обратите внимание, что все промежуточные кадры имеют серый цвет, а конечный кадр содержит небольшой полый прямоугольник. Это означает, что содержимое первого ключевого кадра перенесено во все кадры.

Создание эффектов анимации с помощью панели Properties

Техника автоматического создания промежуточных кадров позволяет не только создавать эффекты движения объекта, изменения его размера и формы, но и другие, более сложные визуальные эффекты. С помощью панели Properties можно создавать эффект появления или исчезновения объекта, а также эффект изменения его цвета. Вы уже знаете, как изменять визуальные свойства объекта (об этом речь шла в главе 6). В следующем разделе мы расскажем о создании анимации с изменением таких свойств объекта, как прозрачность, яркость и др.

Данную технику можно использовать только с символами. Создать анимацию с изменением свойств объекта или группы объектов, созданных с помощью инструментов Flash непосредственно на области действия, нельзя.

Создание анимации с изменением прозрачности объекта

Для создания анимации с изменением прозрачности объекта выполните следующие действия.

1. Создайте новый документ.
2. Создайте анимацию с автоматическим построением промежуточных кадров, используя символы.
3. Щелкните на первом ключевом кадре и с помощью инструмента Arrow выберите объект, который должен постепенно появляться на области действия.
4. Откройте панель Properties посредством команды Properties меню Window.
5. Откройте раскрывающийся список Color и выберите пункт Alpha.
6. В поле Alpha Value введите значение 0. При этом объект на первом ключевом кадре станет прозрачным.
7. Щелкните на последнем ключевом кадре анимации и с помощью инструмента Arrow выберите на области действия объект, с которым вы работаете.
8. На панели Properties откройте список Color и, выбрав пункт Alpha, введите в поле Alpha Value значение прозрачности 100. При этом объект в последнем ключевом кадре станет полностью *непрозрачным*.
9. Протестируйте анимацию, выполнив одно из следующих действий:
 - > с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
 - > выберите команду Play меню Control либо нажмите клавишу Return/Enter;
 - > выберите команду Test Scene или Test Movie меню Control.

СОВЕТ

Чтобы создать эффект исчезновения объекта, необходимо для первого ключевого кадра ввести в поле Alpha Value значение прозрачности 100, а для последнего — значение 0.

Создание анимации с изменением яркости объекта

Данный процесс практически идентичен ранее описанному процессу создания анимации с изменением прозрачности объекта.

1. Создайте новый документ.
2. Создайте анимацию с автоматическим построением промежуточных кадров, используя символы.
3. Щелкните на первом ключевом кадре и с помощью инструмента Arrow выберите объект, яркость которого должна постепенно меняться.
4. Откройте панель Properties посредством команды Properties меню Window.
5. Откройте список Color и выберите пункт Brightness (Яркость).

6. В поле Brightness Value введите значение яркости или используйте для этой цели бегунок.
7. Щелкните на последнем ключевом кадре анимации и с помощью инструмента **Aggow** выберите на области действия объект, с которым вы работаете.
8. На панели **Properties** откройте список **Color** и, выбрав пункт **Brightness**, введите в поле **Brightness Value** значение яркости, отличное от предыдущего.
9. Для тестирования анимации выполните одно из следующих действий:
 - ▷ с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
 - ▷ выберите команду **Play** меню **Control** либо нажмите клавишу **Return/Enter**;
 - ▷ выберите команду **Test Scene** или **Test Movie** меню **Control**.

ПРИМЕЧАНИЕ

Создание совершенно реального эффекта изменения яркости объекта невозможно, поскольку в природе это все равно выглядит иначе. Поэтому, говоря об изменении яркости объекта, мы фактически имеем в виду эффект изменения интенсивности его цвета.

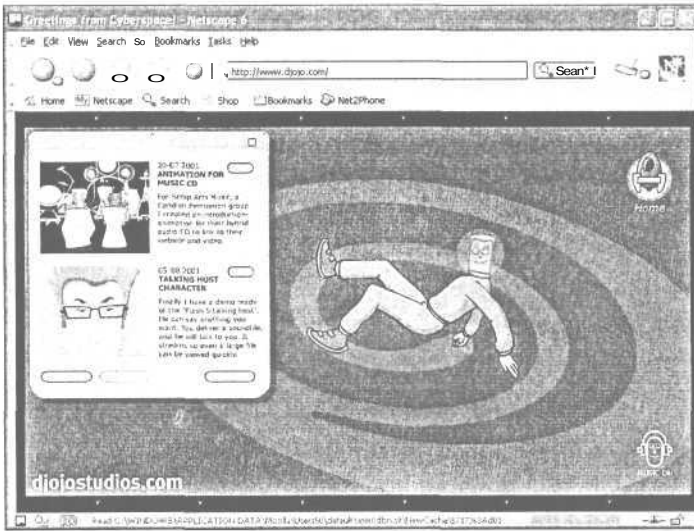
Создание анимации с изменением цвета объекта

Данный процесс аналогичен двум ранее описанным.

1. Создайте новый документ.
2. Создайте анимацию с автоматическим построением промежуточных кадров, используя символы.
3. Щелкните на первом ключевом кадре и с помощью инструмента **Aggow** выберите объект, цвет которого должен постепенно меняться.
4. Откройте панель **Properties** посредством команды **Properties** меню **Window**.
5. Откройте список **Color** и выберите пункт **Tint** (Цветовой оттенок).
6. Щелкните на образце цвета, при этом откроется палитра, в которой следует выбрать один из цветов. В качестве альтернативы можно самостоятельно создать **RGB**-цвет посредством ввода значений в поля **R**, **G** и **B**.
7. Щелкните на последнем ключевом кадре анимации и с помощью инструмента **Aggow** выберите на области действия объект, с которым вы работаете.
8. Используя панель **Properties**, выберите другой цвет.
9. Для тестирования анимации выполните одно из следующих действий:
 - ▷ с помощью мыши переместите головку воспроизведения вдоль временной шкалы;
 - ▷ выберите команду **Play** меню **Control** либо нажмите клавишу **Return/Enter**;
 - ▷ выберите команду **Test Scene** или **Test Movie** меню **Control**.

Разработки

Хотя возможности современной версии Flash MX позволяют создавать полный спектр превосходных мультимедийных продуктов, основная ее мощь проявляется при создании анимации. С помощью Flash можно разрабатывать как самую простую анимацию, так и профессиональные проекты для Web, содержащие большое число визуальных и звуковых эффектов. В этом разделе мы хотим познакомить вас с проектом, в котором прекрасно реализованы возможности Flash по созданию анимации. Это Web-узел компании Djojo Studios www.djojostudios.com.



Компанию основал энтузиаст из Нидерландов Джуст ван Шейк, чтобы создавать различные интерактивные мультимедийные продукты для Web. Web-узел компании являет собой пример отличного дизайна, в котором гармонично сочетаются прекрасный стиль и профессионально выполненная анимация. Один из самых ярких визуальных эффектов этого Web-узла — изображение вращающегося по спирали космонавта, выполненное с использованием 2D и 3D-технологий.

Резюме

В этой главе рассмотрены вопросы, связанные с созданием анимации в среде Flash. Сначала вы познакомились с основными понятиями анимации, такими как кадры, ключевые кадры и частота смены кадров. Далее научились создавать простейшую покадровую анимацию, а затем изучили *tweening* — технику анимации с автоматическим созданием промежуточных кадров. В конце главы были рассмотрены вопросы создания анимированных масок, а также специальных эффектов с помощью панели Properties.

Глава 10

Работа с клипами

Возможно, одним из самых важных элементов Flash-фильма являются клипы. В главе 7, говоря о типах символов, мы уже упоминали клипы, но не рассматривали их подробно. Клипы — это самостоятельные фильмы, которые можно помещать на область действия и воспроизводить независимо от главной временной шкалы. Кроме того, сценарии языка `ActionScript` позволяют управлять клипами, что делает их исключительно мощным средством создания анимационных эффектов.

Техника создания анимации с помощью временной шкалы вам уже знакома, поэтому данную главу мы начнем с описания основных характеристик клипов. Далее вы научитесь создавать клипы, а также преобразовывать в клипы уже созданную анимацию. В конце главы будет рассказано, как помещать клипы в кадры главной временной шкалы.

В этой главе мы рассмотрим следующие темы:

- что такое клип;
- создание новых клипов;
- преобразование анимации в клипы;
- вставка клипов в кадры временной шкалы;
- редактирование клипов с помощью панели `Properties`;
- пример практического создания клипа.

Что такое клип

Не будет преувеличением сказать, что клипы являются одними из самых эффективных и мощных объектов `Flash`. Однако в отличие от кнопок и графических символов клипы лучше рассматривать и изучать в одном контексте с анимацией. Клипы по своей сути являются самостоятельными фильмами, которые вкладываются в основной фильм, поэтому их временная шкала работает независимо от главной временной шкалы. Другими словами, клип можно рассматривать как фильм внутри другого фильма.

На области действия клипы могут использоваться как независимо, так и совместно с другими символами. Например, для анимации символа типа `Button` (Кнопка) вы можете связать клип с одним из состояний кнопки (этот процесс будет рассмотрен в главе 13). Клипы имеют еще одно существенное преимущество. В отличие от обычной анимации, для работы которой необходимо большое число кадров и ключевых кадров на временной шкале, клип функционирует, если создан только один ключевой кадр.

Поскольку клипы воспроизводятся независимо от основной временной шкалы, их часто создают не связанными с основным фильмом. Например, представьте,

что в вашей анимации несколько птиц, каждая из которых создается как отдельный клип, летают в небе. Обычно клипы воспроизводятся либо один раз, либо как непрерывный цикл. В обоих случаях возможности фильма достаточно ограничены. Но если использовать сценарии, созданные с помощью языка ActionScript, клипами можно управлять. Фактически можно создавать сценарий для управления отдельным клипом. Вернемся к примеру со «стаей» птиц: можно написать сценарий, где ни одна птица не будет сталкиваться в небе с другой. При соответствующем опыте и подготовке можно создать сценарий, который позволит имитировать поведение птиц в стае. (Более подробно об управлении клипами с помощью сценариев ActionScript будет рассказано в главе 16.)

Создание нового клипа

Для создания нового клипа вам понадобятся знания и навыки, полученные при изучении глав 7 и 9.

1. Откройте новый документ, выбрав команду New меню File.
2. Выберите команду New Symbol меню Insert.
3. После открытия диалогового окна Create New Symbol (Создание нового символа) убедитесь, что переключатель типа символа установлен в положение Movie Clip, введите имя в поле Name и щелкните на кнопке ОК. (Процедура создания нового символа описана в главе 7.)
4. На данном этапе Flash перейдет в режим редактирования символа. Для создания анимации используйте временную шкалу. Помните, что клип ничем не отличается от обычной анимации, созданной с помощью временной шкалы, то есть допустимо применение любой известной вам техники (tweening, кадрового способа, специальных эффектов и т. д.). Возможно также использование нескольких слоев для анимации множества различных объектов.

ПРИМЕЧАНИЕ

Поскольку клипы содержат сложную анимацию, для их создания могут потребоваться графические символы. Создать графические символы вы можете непосредственно в пределах клипа, после чего они автоматически помещаются в библиотеку, что позволяет использовать их в дальнейшем.

5. Завершив создания анимации в клине, возвратитесь к основной временной шкале, щелкнув на имени сцены в строке Scene and Symbol.
6. Для просмотра созданного клипа откройте окно библиотеки, выберите клип и щелкните на кнопке Play, расположенной в правом верхнем углу окна предварительного просмотра символов.

СОВЕТ

Чтобы протестировать клип непосредственно в режиме редактирования символа, нажмите клавишу Return/Enter.

Преобразование существующей анимации в клипы

В Flash-фильмах большинство клипов изначально создаются как символы типа Movie Clip. Однако существует возможность преобразовать уже созданную анимацию в клип. Для этого выполните следующие действия.

1. На главной временной шкале выберите на всех слоях все кадры, которые хотите превратить в клип.
2. Скопируйте нужные кадры, выбрав команду Copy Frames меню Edit. В качестве альтернативы щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и активизируйте команду Copy Frames в раскрывшемся контекстном меню.
3. Выберите команду New Symbol меню Insert или воспользуйтесь комбинацией клавиш Cmd/Ctrl+F8.

СОВЕТ

Прежде чем производить вставку нового символа, убедитесь, что на области действия нет выбранных объектов.

4. После появления диалогового окна Create New Symbol убедитесь, что переключатель типа символа установлен в положение Movie Clip, введите имя в поле Name и щелкните на кнопке ОК.
5. Переключившись в режим редактирования символа, выберите первый кадр на временной шкале и вставьте кадры командой Paste Frames меню Edit. В качестве альтернативы щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите команду Paste Frames в раскрывшемся контекстном меню. При этом все скопированные кадры будут вставлены в клип.
6. Для возвращения к главной временной шкале выберите команду Edit Document меню Edit или щелкните на имени сцены в строке Scene and Symbol.
7. Теперь, когда вы возвратились к основному фильму, можно удалить все кадры, которые использовались при создании клипа. Для этого выберите все кадры, которые необходимо удалить, откройте меню Edit и выберите команду Cut Frames. В качестве альтернативы щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите команду Cut Frames в раскрывшемся контекстном меню.

Вставка клипа в кадр на главной временной шкале

Итак, вы научились создавать новые клипы и преобразовывать существующую анимацию в клипы. Теперь расскажем, как вставить их в главный фильм.

1. Откройте окно библиотеки, выбрав команду Library меню Window.

2. Выберите на временной шкале кадр, в который хотите вставить клип. Помните, что клипы можно вставлять только в ключевые кадры.
3. С помощью мыши перетащите клип из окна библиотеки на область действия. Хотя сам клип может содержать сотни кадров, на основной шкале он будет занимать только один ключевой кадр.
4. Чтобы увидеть, как клип будет воспроизводиться в фильме, выберите команду Test Movie или Test Scene меню Control.

Манипулирование клипами

После того как клип вставлен в кадр на временной шкале, им можно манипулировать, как обычным символом, используя панель Properties. Некоторые важные операции можно также выполнять, используя библиотеку. В следующих разделах будут рассмотрены различные действия, которые можно производить с клипами.

Присвоение имени экземпляру символа

Вам уже известно, что при помещении символа из библиотеки на область действия создается экземпляр данного символа — копия родительского символа, хранящегося в библиотеке. Это справедливо и для клипов, поскольку они также являются символами. С помощью панели Properties можно присвоить созданному экземпляру символа уникальное имя. При работе со сценариями языка ActionScript такое имя необходимо для управления клипами, помещенными в кадры на главной временной шкале. Чтобы точно указать в сценарии требуемый клип, необходимо идентифицировать каждый экземпляр символа с помощью уникального имени. Таким образом, присвоение уникальных имен экземплярам символов абсолютно необходимо при создании сложных Flash-фильмов с большим числом клипов:

Чтобы присвоить имя экземпляру символа типа Movie Clip, выполните следующие действия.

1. С помощью мыши перетащите клип из окна библиотеки на область действия.
2. Используя инструмент Arrow, выберите только что перемещенный экземпляр символа.
3. Откройте панель Properties, выбрав команду Properties меню Window.
4. Введите имя для экземпляра символа в поле Name Instance.

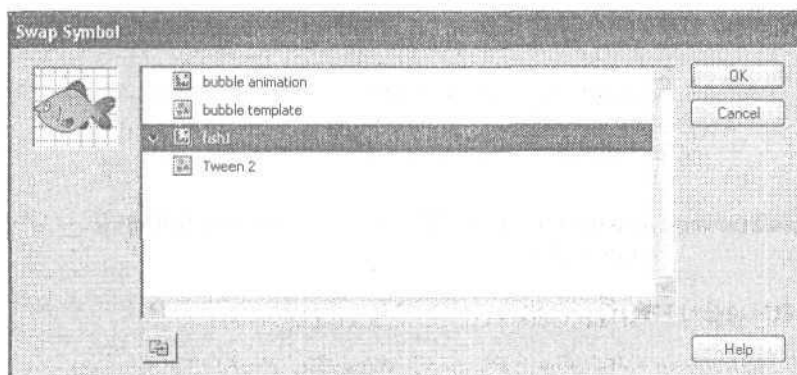
СОВЕТ

При вводе имени в поле Instance Name недопустимо использование пробелов. Если вы хотите задать имя из двух слов, разделите их с помощью дефиса (-) или символа подчеркивания (_).

Замена одного клипа другим

Иногда возникают ситуации, когда необходимо заменить один экземпляр символа другим. Для выполнения этой задачи используйте кнопку Swap панели Properties.

1. На главной временной шкале выберите клип, который хотите заменить.
2. Откройте панель Properties, выбрав команду Properties меню Window.
3. Щелкните на кнопке Swap.
4. Когда откроется диалоговое окно Swap Symbol (Замена символа), выберите заменяющий символ и щелкните на кнопке ОК.



Редактирование клипа

Если вы хотите отредактировать родительский символ клипа, откройте окно библиотеки, выберите клип, затем откройте меню Options и активизируйте команду Edit. В качестве альтернативы можно выполнить двойной щелчок мышью на экземпляре клипа в области действия и, переключившись в режим редактирования символа, произвести все необходимые изменения. Для выхода из режима редактирования символа щелкните на имени сцены в строке Scene and Symbol.

Создание дубликата символа

Flash позволяет создавать дубликаты символов библиотеки. Такой дубликат может понадобиться при замене символа его модифицированной копией. Создав дубликат символа, вы можете отредактировать его с помощью команды Edit меню Options, а затем использовать для замены экземпляра символа на области действия. Рассмотрим процесс создания дубликата символа.


1. Откройте Flash-фильм, содержащий клип, который вы хотите дублировать.
2. Откройте окно библиотеки командой Library меню Window.
3. Выберите клип, который хотите дублировать.
4. Откройте меню Options и выберите команду Duplicate.

5. Когда появится диалоговое окно Duplicate Symbol (Дублировать символ), введите имя для дубликата символа, убедитесь, что переключатель типа символа установлен в положение Movie Clip, и щелкните на кнопке ОК.

При этом дубликат клипа будет автоматически добавлен в библиотеку, после чего вы сможете манипулировать им как обычным символом.

Пример практического создания клипа

Теперь, когда вы изучили основные техники и методы создания клипов и манипулирования ими, можно приступить к практической работе. Даже без сценариев ActionScript (которые будут рассмотрены в главе 16), используя клипы, можно создавать сложную анимацию, которую трудно было бы сделать с применением только временной шкалы.

 На прилагаемом компакт-диске в каталоге Chapter 10 находятся все файлы рисунков (в формате SWF), относящиеся к данному разделу. Кроме того, посмотреть готовый проект можно, открыв файл `under_the_sea.swf` или `under_the_sea fla.`

В следующих разделах мы создадим 2D-сцену с изображением подводного мира, в котором плавает одна рыбка.

Создание изображения океана

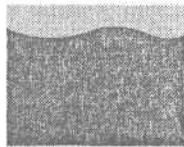
Работу над проектом начнем с создания изображения океана.

1. Создайте новый документ, выбрав команду New меню File.
2. Выберите команду Document меню Modify. Когда откроется диалоговое окно Document Properties (Свойства документа), установите в поле Width значение ширины 500 и в поле Height значение высоты 400.
3. Для фона документа выберите светло-голубой цвет.
4. Чтобы закрыть диалоговое окно Document Properties, щелкните на кнопке ОК.
5. Выберите команду Import меню File.
6. Когда откроется диалоговое окно Import, найдите на компакт-диске и откройте каталог Chapter 10, выберите файл `water.swf` и щелкните на кнопке ОК.

СОВЕТ

Если вы не можете найти требуемый файл в диалоговом окне Import, выберите пункт All Files раскрывающегося списка Files of Type.

7. С помощью инструмента Arrow выровняйте изображение воды таким образом, чтобы его края совпали с краями области действия.
8. Поскольку в данном фильме будет использовано несколько слоев, необходимо присвоить каждому из них уникальное имя. Щелкните два раза мышью на единственном пока значке слоя и в открывшемся диалоговом окне введите в поле Name имя `water`. Щелкните на кнопке ОК.



Создание изображения рыбки

На данном этапе в единственном пока ключевом кадре вы создали изображение воды. Теперь нужно создать изображение рыбки.

1. Выберите команду New Symbol меню Insert или воспользуйтесь комбинацией клавиш Cmd/Ctrl+F8.
2. В диалоговом окне Create New Symbol установите переключатель типа символа в положение Movie Clip, в поле Name укажите имя fish1 и нажмите кнопку ОК.
3. После переключения в режим редактирования символа откройте диалоговое окно Import, выбрав команду Import меню File, найдите на компакт-диске и откройте каталог Chapter 10, выберите файл fish.swf и щелкните на кнопке ОК. При этом импортируется изображение, которое вы будете использовать для создания клипа.

СОВЕТ

Если изображение не появилось, возможно, Flash поместил его за пределы области видимости. С помощью мыши переместите изображение. Помните, что крестик обозначает точку регистрации символа.

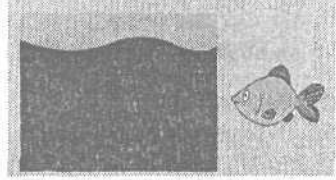
4. На данном этапе необходимо создать анимацию, в которой изображение рыбки будет перемещаться горизонтально от левого края сцены к правому. Сначала с помощью инструмента Arrow выберите изображение рыбки.
5. Откройте панель Properties, выбрав одноименную команду меню Window.
6. Введите значения 470 и -78,5 в поля X и Y соответственно.
7. Щелкните на кадре с номером 90 и вставьте ключевой кадр, выбрав команду Keyframe меню Insert.
8. В созданном ключевом кадре выберите изображение рыбки.
9. Введите значения -400 и -78,5 в поля X и Y соответственно.
10. Возвратитесь к первому ключевому кадру и создайте промежуточные кадры, выбрав команду Create Motion Tween меню Insert.

ПРИМЕЧАНИЕ

Изначально изображение рыбки должно находиться на рабочей области справа от области действия, затем переместиться вдоль сцены и выйти за пределы области действия на другую сторону рабочей области. При этом будет создаваться иллюзия, что рыбка проплыла мимо вас и вышла из поля зрения.

11. Если вы хотите несколько оживить сцену, можете заставить рыбку двигаться извилистым путем. Способ создания направляющей движения подробно описан в главе 9.
12. Теперь, когда вы создали клип с движущейся рыбкой, необходимо поместить ее в воду. Возвратитесь к основной сцене, выбрав команду Document меню Edit или щелкнув на имени сцены в строке Scene and Symbol.
13. Создайте новый слой, выбрав команду Layer меню Insert или щелкнув на кнопке Insert Layer, расположенной в левом нижнем углу временной шкалы.

14. Присвойте новому слою имя *fish*.
15. Убедитесь, что слой *fish* находится на временной шкале выше слоя *water*.
16. Выберите первый ключевой кадр слоя *fish*.
17. Откройте окно библиотеки, выбрав команду *Library* меню *Window*, и с помощью мыши перетащите клип на область действия. Затем, используя инструмент *Aggow*, переместите изображение рыбки за пределы области действия, как показано на следующем рисунке.



Чтобы протестировать созданную сцену, выберите команду *Test Movie* или *Test Scene* меню *Control*.

Добавление пузырьков

Завершим данный учебный фильм созданием изображения всплывающих пузырьков.

1. Выберите команду *New Symbol* меню *Insert*.
2. Когда откроется диалоговое окно *Create New Symbol*, установите переключатель типа символа в положение *Graphic*, введите имя *bubble template* в поле *Name* и щелкните на кнопке *OK*.
3. После перехода в режим редактирования символа щелкните на первом ключевом кадре единственного слоя временной шкалы и с помощью инструмента *Oval* создайте небольшой кружок в нижней части экрана. Этот кружок будет служить шаблоном всплывающего пузырька. Постарайтесь подобрать цвет заливки и обводки таким образом, чтобы изображение максимально напоминало пузырек воздуха в воде.
4. Далее необходимо использовать созданный графический символ с изображением пузырька для создания клипа, в котором пузырек будет подниматься вверх. Выберите команду *New Symbol* меню *Insert*.
5. Когда откроется диалоговое окно *Create New Symbol*, установите переключатель типа символа в положение *Movie Clip*, введите имя *bubble* в поле *Name* и щелкните на кнопке *OK*.
6. После перехода в режим редактирования символа щелкните на первом ключевом кадре единственного слоя временной шкалы и с помощью мыши перетащите графический символ с изображением пузырька из окна библиотеки на область действия.
7. Выберите изображение пузырька и откройте панель *Info*. Введите значения 14.7 и 76.3 в поля *X* и *Y* соответственно.
8. Выберите кадр с номером 30 и вставьте ключевой кадр, выбрав команду *Keyframe* меню *Insert*.
9. Во втором ключевом кадре измените значение в поле *X* на -9.8 и в поле *Y* на -202.7.

10. Еще раз выберите первый ключевой кадр и активизируйте команду Create Motion Tween меню Insert — будет создана анимация, в которой пузырек воздуха поднимается вверх.
11. Чтобы несколько оживить этот процесс, выберите один из промежуточных кадров (например, кадр с номером 15).

ПРИМЕЧАНИЕ

Пусть вас не смущает, что при двойном щелчке на одном из промежуточных кадров будут выбраны все промежуточные кадры. На выбранный вами кадр будет указывать головка воспроизведения.

12. Вставьте ключевой кадр, выбрав команду Keyframe меню Insert. Ваша временная шкала должна выглядеть, как показано на следующем рисунке.



13. Выберите изображение пузырька в созданном промежуточном ключевом кадре. Для увеличения размера пузырька используйте либо инструмент Free Transform в режиме Scale (Масштабирование), либо панель Transform, которая открывается с помощью команды Transform меню Window. Выполнив данную процедуру, вы создадите анимацию, в которой пузырек в процессе подъема сначала слегка увеличивается, а затем возвращается к прежнему размеру.
14. Теперь необходимо, чтобы по достижении поверхности воды пузырек исчезал. Для этого сначала выберите последний ключевой кадр.
15. Выберите изображение пузырька с помощью инструмента Arrow.
16. Откройте панель Properties, выбрав команду Properties меню Window.
17. Откройте меню Color и выберите пункт Alpha. В поле Alpha Value введите значение 0. При этом в последнем ключевом кадре пузырек станет прозрачным.

Увеличение числа пузырьков

Используя клип с изображением воздушного пузырька, вы можете создать несколько пузырьков, которые при воспроизведении фильма будут поочередно подниматься вверх.

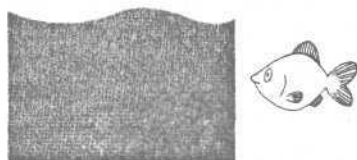
1. Откройте клип с изображением пузырька в режиме редактирования символа и создайте дополнительный слой, выбрав команду Layer меню Insert или щелкнув на кнопке Insert Layer, расположенной в левом нижнем углу временной шкалы. Присвойте новому слою уникальное имя.
2. Щелкните на кадре с номером 5 и вставьте ключевой кадр, выбрав команду Keyframe меню Insert либо нажав клавишу F6. Теперь второй пузырек начнет движение, когда головка воспроизведения дойдет до кадра с номером 5, в результате пузырьки будут подниматься поочередно.
3. С помощью мыши перетащите графический символ с изображением пузырька из окна библиотеки на область действия. С помощью панели Info установите в полях X и Y значения -51.5 и 91.5 соответственно.

- Щелкните на кадре с номером 35 и вставьте ключевой кадр, выбрав команду Keyframe меню Insert.
- Повторите процесс, но для второго пузырька на панели Info установите в полях X и Y значения -48.5 и -120.5 соответственно.
- На данном этапе ваш фильм уже содержит два поднимающихся пузырька. Добавьте еще несколько, вставляя ключевые кадры и помещая на область действия экземпляры символа. Для разнообразия сделайте пузырьки различного размера.

Добавление пузырьков на основную сцену

Теперь необходимо добавить клип с изображением поднимающегося пузырька на главную сцену.

- Вернитесь на основную сцену, выбрав команду Document меню Edit, либо щелкните на имени сцены в строке Scene and Symbol.
- Создайте новый слой, выбрав команду Layer меню Insert или щелкнув на кнопке Insert Layer, расположенной в левом нижнем углу временной шкалы.
- Присвойте новому слою имя bubbles.
- Выберите первый ключевой кадр слоя bubbles.
- Откройте окно библиотеки, выбрав команду Library меню Window, и с помощью мыши перетащите клип с движущимся пузырьком на область действия. Выровняйте изображение пузырька таким образом, чтобы оно находилось на рабочей области ниже области действия. При воспроизведении фильма будет создаваться впечатление, что пузырек появляется снизу сцены. Сцена должна выглядеть, как показано на рисунке.
- Чтобы увидеть, как клип будет воспроизводиться в фильме, выберите команду Test Movie или Test Scene меню Control.



Поздравляем! Вы создали первую сцену с использованием клипов. Можете попробовать несколько усложнить фильм, добавив еще несколько рыбок или качающиеся подводные растения.

Разработки

Применение клипов позволяет максимально эффективно использовать возможности Flash. Однако не все разработчики умеют применять поистине неограниченные возможности клипов в полной мере. Хорошим примером использования клипов служит Web-узел компании Neostream www.neostream.com.

Компания Neostream, основанная в городе Сидней (Австралия), является одним из лидеров рынка мультимедийных продуктов и специализируется на выпуске мультимедийных компакт-дисков, приложений для Web. Кроме того, компания

занимается разработкой фирменных стилей для других компаний. Web-узел компании Neostream является примером аккуратного Flash-дизайна (рис. 10.1) – каждый элемент экрана связан с клипом, включая окно предварительной загрузки и элементы интерфейса. Следует отметить, что особенно удачно анимированы кнопки. Рекомендуем посетить данный Web-узел и познакомиться с прекрасной анимацией, созданной его разработчиками.

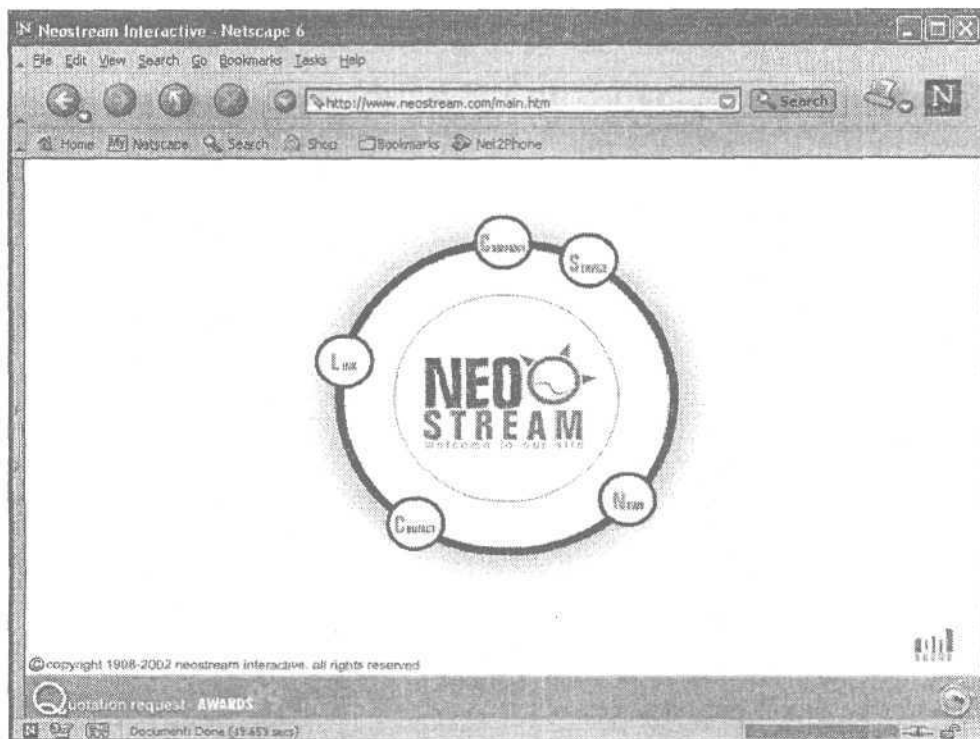


Рис. 10.1. Web-узел компании Neostream

Резюме

В этой главе вы пополнили свои знания, познакомились с базовыми свойствами клипов и основными техниками их создания. В начале главы были приведены основные характеристики клипов, затем рассказано, как создавать новые и преобразовывать существующую анимацию в клипы. Далее вы узнали, как вставлять клипы в кадры главной временной шкалы и редактировать их с помощью панели Properties и других инструментов. Наконец, используя инструкцию из последнего раздела данной главы, вы создали фильм с использованием клипов.

Глава 11

Использование сцен при работе с анимационными фильмами

Вы уже знаете, что Flash предоставляет возможности для создания достаточно сложных фильмов. Даже небольшой по размеру фильм может содержать интерактивные элементы, анимацию, клипы и т. д. Следовательно, возникает необходимость в разработке метода для организации хранения и поиска различных элементов фильма.

Представьте, что вы можете разбить свой проект на несколько управляемых фрагментов, каждый из которых содержит определенную серию событий (анимацию, интерактивный диалог с пользователем и т. д.). Понятно, что такая структура значительно упростит работу над проектом. Именно для этого во Flash-фильмах используются *сцены*. Как в театре, сцены Flash представляют собой отдельные фрагменты, то есть логические части, составляющие фильм. Сцены напоминают миниатюрные фильмы, последовательное воспроизведение которых и создает единый Flash-фильм. При создании больших проектов использование сцен значительно экономит время и помогает избежать многих трудностей.

В данной главе рассматриваются следующие темы:

- ▶ знакомство со сценами;
- ▶ создание сцен и манипулирование ими;
- ▶ переключение между сценами;
- ▶ воспроизведение сцен;
- ▶ управление сценами с помощью сценариев `ActionScript`.

Знакомство со сценами

Сцены — это эффективный инструмент, помогающий разделять общее содержимое фильма на отдельные, управляемые фрагменты. Каждая сцена представляет собой миниатюрный фильм. Сцены воспроизводятся последовательно и, являясь самостоятельными, тесно связаны между собой. При воспроизведении фильма все сцены связываются в порядке, указанном на панели Scene (Сцена). Во время этого процесса между ними никогда не возникает заметных задержек или скачков изображения.

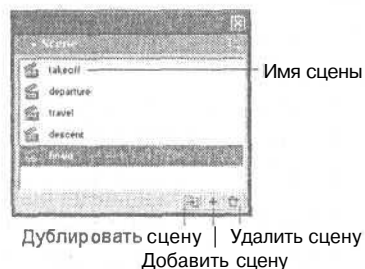
ПРИМЕЧАНИЕ

Число используемых сцен Flash-фильма ограничено только объемом памяти компьютера.

Возможности применения сцен разнообразны и практически безграничны. Например, работая над проектом Web-узла, вы можете реализовать разделы и подразделы в виде сцен. Или другой пример, сейчас все большую популярность в Web завоевывают небольшие анимационные Flash-фильмы, содержимое которых можно разделять на логические части также посредством сцен.

Панель Scene

Доступ к списку содержащихся в фильме сцен осуществляется с помощью панели Scene, которая открывается выбором команды Scene меню Window. Данная панель позволяет просматривать список имеющихся сцен и устанавливать порядок их воспроизведения в фильме. Кроме того, можно дублировать, добавлять, удалять и перемещать сцены в пределах списка. Порядок воспроизведения сцен в фильме определяется порядком расположения сцен в списке панели Scene.



В отличие от большинства других панелей программы меню Options панель Scene содержит только три команды: Maximize Panel (Сделать размер панели максимальным), Close Panel (Закреть панель) и Help (Справка).

Строка Scene and Symbol

Строка Scene and Symbol (рис. 11.1) расположена ниже временной шкалы и выше рабочей области. Вы уже умеете использовать эту строку при работе с окном Symbol Editor (Редактор символов) для выхода из режима редактирования символа. В данном разделе мы познакомим вас с функциональными возможностями строки Scene and Symbol, касающимися работы со сценами.



Рис. 11.1. Кнопка Edit Scene

Строка Scene and Symbol содержит имя текущей сцены (на рис. 11.1 текущей является сцена *finale*). Переключиться на другую сцену можно с помощью кнопки Edit Scene (Правка сцены), расположенной в правой части строки Scene and Symbol, при этом в строке будет выведено имя новой сцены. Более подробно кнопка Edit Scene будет рассмотрена в разделе «Переключение между сценами».

Создание сцен и манипулирование ими

Как уже говорилось, панель Scene, которая открывается с помощью команды Scene меню Window, предназначена для работы со сценами. В этом разделе мы расскажем, как с помощью данной панели можно добавлять, дублировать, переименовывать сцены, а также изменять их порядок.

Добавление сцены

По мере увеличения размера и сложности проекта вам придется добавлять сцены для распределения логических фрагментов содержимого фильма. Используя панель Scene, вы можете добавлять столько сцен, сколько необходимо. Чтобы добавить сцену, выполните следующие действия.

1. Убедитесь, что необходимый документ открыт.
2. Откройте панель Scene, выбрав команду Scene меню Window.
3. Щелкните на кнопке Add Scene (Добавить сцену), которая расположена в правом нижнем углу панели, либо выберите команду Scene меню Insert. После этих действий в окне панели Scene появится еще одна сцена. По умолчанию Flash всегда добавляет новую сцену ниже выбранной в настоящий момент и присваивает ей имя со следующим порядковым номером (например, Scene 1, Scene 2 и т. д.).
4. Выберите новую сцену на панели Scene и создайте содержимое.

ПРИМЕЧАНИЕ

При создании новой сцены Flash переключается на нее автоматически.

Удаление сцены

Для удаления сцены выполните следующие действия.

1. Откройте панель Scene, выбрав команду Scene меню Window.
2. Выберите сцену, которую хотите удалить.
3. Щелкните на кнопке Delete Scene, расположенной в правом нижнем углу панели.
4. Для подтверждения удаления сцены щелкните на кнопке OK в открывшемся диалоговом окне.

Дублирование сцены

В предыдущих главах мы уже говорили о копировании и создании дубликатов различных элементов. Копирование целой анимации — достаточно громоздкий процесс, а создание копии сложного проекта с большим числом сцен вообще представляется трудновыполнимым.

Но компания Macromedia решила эту проблему, внедрив простую функцию создания дубликата, позволяющую создавать точные копии любой сцены нажатием всего одной кнопки.

1. Откройте панель Scene, активизировав команду Scene меню Window, и выберите сцену, дубликат которой хотите создать.
2. Щелкните на кнопке Duplicate Scene (Дублировать сцену), расположенной в правом нижнем углу панели Scene.
3. В окне панели Scene появится дубликат сцены. Обратите внимание, что его имя состоит из названия исходной сцены и слова сору (копия).

Переименование сцен

Поскольку имена, используемые по умолчанию для дубликатов и новых сцен, отличаются только порядковым номером, при поиске определенного содержимого сцену трудно идентифицировать. Поэтому в больших проектах целесообразно присваивать сценам специальные имена, которые характеризуют их содержимое. Для переименования сцены выполните следующие действия.

1. Откройте панель Scene, выбрав команду Scene меню Window, и выполните двойной щелчок на имени сцены, которое хотите изменить. При этом появится возможность редактировать имя сцены.
2. Введите новое имя и нажмите клавишу Return/Enter либо выполните двойной щелчок за пределами панели Scene.

Изменение порядка размещения сцен

Как уже говорилось ранее, порядок размещения сцен на панели Scene определяет последовательность их воспроизведения. Сцены на этой панели можно перемещать, устанавливая порядок их воспроизведения независимо от последовательности, в которой они были созданы. Для изменения порядка размещения сцен выполните следующие действия.

1. Откройте панель Scene, выбрав команду Scene меню Window.
2. Установите указатель мыши на имени сцены, нажмите левую кнопку мыши и переместите указатель в требуемое положение. Обратите внимание, что при перемещении указателя появляется голубая линия, указывающая возможное новое положение сцены в списке.

Переключение между сценами

Для переключения между различными сценами фильма в процессе работы над Flash-проектом можно использовать следующие навигационные средства.

Панель Scene

Чтобы перейти к нужной сцене фильма, щелкните на имени сцены на панели Scene. Помните, что имя текущей сцены выводится в строке Scene and Symbol.

Кнопка Edit Scene

При щелчке на кнопке Edit Scene (рис. 11.1), находящейся в правой части строки Scene and Symbol, вы получаете доступ к меню с именами всех сцен, имеющихся в фильме. Для перехода к нужной сцене выберите соответствующий пункт меню.

Панель Movie Explorer

Как уже говорилось в главе 7, панель Movie Explorer (открыть которую можно с помощью команды Movie Explorer меню Window) служит для отображения иерархической структуры фильма. С помощью данной панели можно осуществлять поиск символов и экземпляров символов, а также производить замену текста и шрифтов.

Поскольку фильм состоит из сцен, они представлены на панели Movie Explorer как элементы высшего порядка в иерархии.

Для перехода к определенной сцене найдите ее имя на панели Movie Explorer и выполните щелчок.

СОВЕТ

По умолчанию на панели Movie Explorer выводится содержимое только текущей сцены. Чтобы одновременно просматривать содержимое всех сцен, выберите из меню Options данной панели команду Show All Scenes (Показывать все сцены).

Тестирование сцен

Вам уже известны несколько способов тестирования созданного фильма. Для проверки результата непосредственно в среде разработки Flash можно нажать клавишу Return/Enter, но при этом возможности просмотра будут ограничены только текущей сценой. Последовательное воспроизведение всех сцен возможно только после публикации фильма, при этом уже не требуется среда Flash.

Чтобы произвести тестирование, необходимо выполнить следующие действия.

- ▶ Для тестирования отдельной сцены выберите ее на панели Scene и нажмите клавишу Return/Enter либо после выбора сцены откройте меню Control и активизируйте команду Test Scene.
- ▶ Для тестирования фильма целиком выберите команду Test Movie меню Control либо используйте комбинацию клавиш Cmd/Ctrl+Return/Enter. При этом откроется новое окно, где все сцены фильма будут воспроизводиться в последовательности, определенной на панели Scene.

Воспроизвести все сцены фильма можно также, выбрав команду Play All Scenes меню Control.

Возможности управления сценами с помощью сценариев ActionScript

Использование сцен значительно облегчает общую организацию содержимого фильма, заданная последовательность воспроизведения сцен приводит к созданию линейных фильмов. Но наряду с рядом преимуществ (например, две сцены не могут воспроизводиться одновременно) в этом есть и свои недостатки, связанные с невозможностью изменить порядок воспроизведения сцен. Эту проблему можно решить, используя сценарии ActionScript (в частности действия кадров), которые позволяют управлять сценами и изменять их порядок.

Более подробно о сценариях языка ActionScript и использовании действий кадров будет рассказано в главе 12. В этом же разделе речь пойдет о *возможностях* использования некоторых действий, с помощью которых можно управлять сценами.

- ▶ gotoAndPlay(). Вызов действия gotoAndPlay() приводит к переходу головки воспроизведения к определенной сцене и кадру. Воспроизведение фильма продолжается с этого кадра.
- ▶ gotoAndStop(). При вызове данного действия происходит переход к определенной сцене и кадру и остановка воспроизведения фильма.
- ▶ play(). Это действие позволяет продолжить воспроизведение фильма после остановки.
- ▶ stop(). С помощью данного действия можно остановить воспроизведение фильма.

Разработки

Сцены позволяют разбить содержимое фильма и сам творческий процесс разработки на логические управляемые блоки. Трудно выделить какой-либо проект в качестве примера эффективного использования сцен, поскольку это присуще сотням Web-узлов, созданных на базе Flash. Одним из них является Web-узел компании Mondo Media (www.mondomedia.com).

Компания Mondo Media была создана для разработки и распространения в сети Интернет развлекательных продуктов, созданных на основе анимации. Среди самых известных разработок компании можно назвать такие небольшие анимационные фильмы, как Thugs on Film (Головорезы в кино), The God and Devil Show (Шоу Господа и дьявола) и Heavy Metal Guy (Металлист). Продукты Mondo Media пользуются большой популярностью, благодаря этому компания стала одним из лидеров в области создания развлекательных продуктов для Интернета.

Web-узел компании Mondo Media (рис. 11.2) был создан на основе технологии Flash с использованием сцен. Разделы Web-узла, особенно содержащие небольшие фильмы (Mondo Mini Shows), представляют собой миниатюрные Web-узлы. Рекомендуем Web-узел компании Mondo Media как прекрасный пример использования сцен для организации хранения и доступа к содержимому.

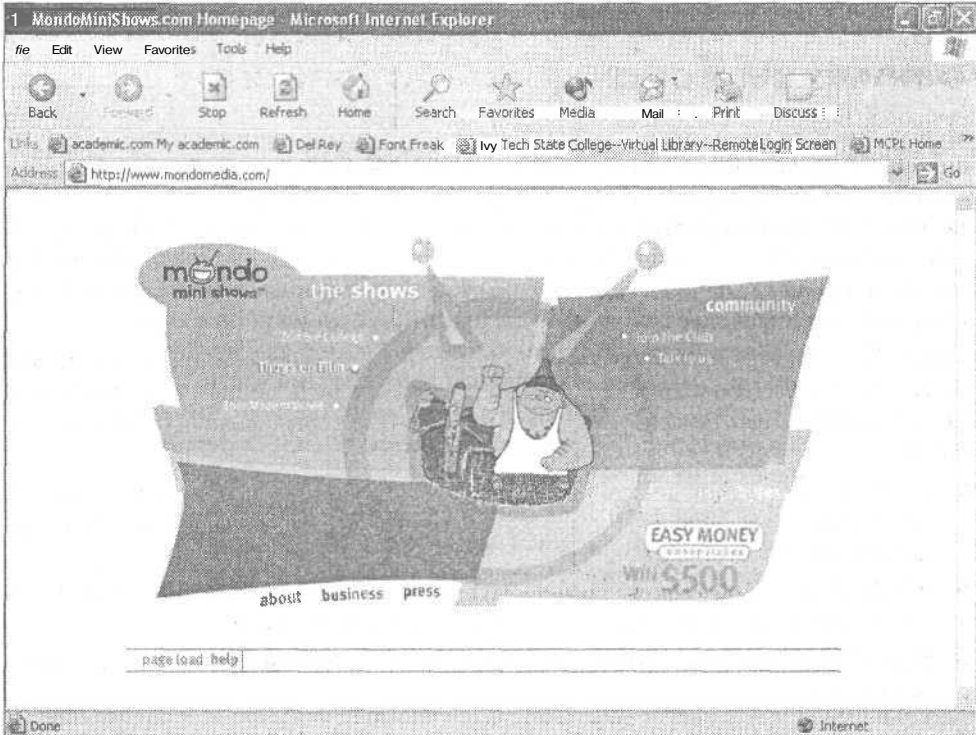


Рис. 11.2. Web-узел компании Mondo Media

Резюме

В этой главе речь шла об использовании сцен как о методе разделения содержимого фильма на логические части. Кроме того, вы узнали, что сцены позволяют создавать интерактивные продукты, где пользователь может определить необходимую для воспроизведения часть фильма. Вы познакомились с общими принципами функционирования сцен в фильме, а затем научились создавать их и манипулировать ими с помощью панели Scene, переключаться между сценами и воспроизводить их. В конце главы мы познакомили вас с возможностями управления сценами с использованием сценариев ActionScript.

Практикум 3

Создание сцены полета космической ракеты

Данный практикум предоставляет возможность совершить импровизированное космическое путешествие. Мощные средства Flash помогут вам создать анимацию полета космической ракеты. Для выполнения этой задачи понадобятся все знания и практические навыки, полученные при изучении части 3 данной книги. Используя временную шкалу, клипы и сцены, вы создадите анимацию, изображающую старт космической ракеты, ее отдаление от Земли, полет в космосе и приближение к одной из планет.



Файлы с изображением всех объектов, используемых в данном фильме, а также копия готового фильма (файлы `space_flight.swf` и `space_flight fla`) находятся на прилагаемом компакт-диске в каталоге Hands On 3.

Изображение космического корабля

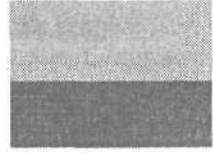
Выполнение учебного задания начните с создания первой сцены — старта космического корабля с Земли.

1. Создайте новый файл, выбрав команду New меню File.
2. В диалоговом окне Document Properties, которое можно открыть выбором команды Document меню Modify, измените частоту смены кадров, установив значение 15. В качестве фонового выберите черный цвет.
3. Щелкните на кнопке ОК.
4. На панели Tools выберите инструмент Rectangle.
5. Откройте цветовую палитру, щелкнув на образце цвета обводки Stroke Color. Затем нажмите кнопку No Color (Нет цвета), расположенную справа на панели Tools и обозначенную красной наклонной линией. После этого форма, создаваемая с помощью инструмента Rectangle, не будет иметь обводки.
6. Щелкните на образце цвета заливки Fill Color и выберите в открывшейся палитре коричневый или зеленый цвет. Создаваемый прямоугольник будет использоваться в качестве изображения земной поверхности.
7. С помощью инструмента Rectangle создайте **прямоугольник**, который должен занимать приблизительно треть области действия.
8. Присвойте существующему слою имя ground (Земля).

СОВЕТ

Вы можете проявить фантазию и дополнить рисунок, изображающий поверхность Земли, различными объектами — строениями, деревьями и т. д.

9. Создайте новый слой, назвав его background. Оставшуюся часть области действия займет фон неба светло-голубого цвета, созданный также с помощью инструмента Rectangle. На данном этапе экран должен соответствовать рисунку, приведенному ниже.



10. Теперь необходимо присвоить имя вашей первой сцене. Откройте панель Scene посредством выбора команды Scene меню Window, выполните двойной щелчок на имени сцены (ваш фильм пока содержит единственную сцену Scene 1) и введите имя takeoff (старт).
11. Следующий шаг — импортирование всех графических объектов, которые будут использоваться в сцене. Выберите команду New Symbol меню Insert. В открывшемся диалоговом окне New Symbol введите в поле Name имя rocket ship (ракета). Убедитесь также, что переключатель типа символа установлен в положение Graphic (Графический) и щелкните на кнопке ОК.
12. При включенном режиме редактирования символа выберите в меню File команду Import, после чего откроется одноименное диалоговое окно. В этом окне найдите и откройте на компакт-диске каталог Hands On 3, выберите файл rocketship.swf и щелкните на кнопке Open. Изображение космической ракеты будет помещено на область действия в режиме редактирования символа. (При импортировании сложных графических рисунков некоторые элементы изображения могут появиться как отдельные части, в этом случае их необходимо сгруппировать. Но в нашем примере все части изображения ракеты уже сгруппированы.)

ПРИМЕЧАНИЕ

Иногда Flash не помещает рисунок в центр окна редактора символов. Если после импортирования изображение объекта не появилось в окне редактора символов, используйте полосы прокрутки, а затем с помощью инструмента Arrow выберите изображение и переместите его мышью в центр окна. Помните, что небольшой крестик в центре окна обозначает точку регистрации объекта.

13. На данном этапе необходимо превратить статическое изображение ракеты в клип, демонстрирующий, как из сопел ракетных двигателей вырывается пламя. Прежде всего, выберите в окне библиотеки графический символ с изображением ракеты. Откройте меню Options и выберите пункт Properties. В качестве альтернативы после выбора символа щелкните на кнопке Properties, расположенной в нижней части окна библиотеки.
14. Когда откроется диалоговое окно Symbol Properties, установите переключатель типа символа в положение Movie Clip (Клип) и щелкните на кнопке ОК. Теперь необходимо добавить изображение пламени.
15. Присвойте существующему слою имя rocket ship (помните, что вы продолжаете работать с клипом в режиме редактирования символа).
16. Создайте новый слой и назовите его flames (пламя).
17. Переместите слой flames ниже слоя rocket ship.

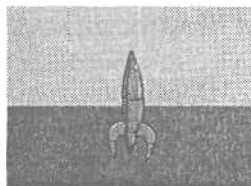
18. Выберите кадр 5 слоя `rocket ship` и произведите вставку кадра, используя команду `Frame` меню `Insert`.
19. Теперь выберите кадр 2 слоя `flames` и вставьте ключевой кадр командой `Keyframe` меню `Insert`. Почему первый ключевой кадр будет содержать только изображение ракеты без пламени из сопел двигателей мы объясним позже.
20. Выберите второй ключевой кадр (который вы добавили) и нарисуйте пламя, используя инструмент `Pencil`. Для создания заливки используйте инструмент `Paint Bucket` с красным цветом. Изображение должно выглядеть, как показано на следующем рисунке.
21. Щелкните на кадре 3 слоя `flames` и вставьте ключевой кадр, выбрав команду `Keyframe` меню `Insert`.
22. Теперь, щелкнув на кадре 4 слоя `flames`, вставьте пустой ключевой кадр, используя команду `Blank Keyframe` меню `Insert`.



ПРИМЕЧАНИЕ

Использование инструментов `Pencil` и `Paint Bucket` подробно описано в главе 4.

23. Выберите только что созданный пустой ключевой кадр и нарисуйте дополнительные языки пламени. Изображение пламени должно быть похоже на предыдущее, но не идентично ему. Для заливки выберите оранжевый цвет. Изображение должно выглядеть, как показано на рисунке ниже.
24. Щелкните на кадре 5 слоя `flames` и вставьте ключевой кадр, выбрав команду `Keyframe` меню `Insert`. Поскольку клип в фильме будет воспроизводиться непрерывно, изменение изображения пламени создаст эффект горения.
25. На данном этапе необходимо вставить клип `rocket ship` в кадры сцены запуска ракеты `takeoff`. Для этого нужно выйти из режима редактирования символа, щелкнув на имени сцены в строке `Scene and Symbol`.
26. Вставьте новый слой и назовите его `rocket ship`.
27. Выберите первый ключевой кадр слоя `rocket ship` и с помощью мыши перетащите клип из окна библиотеки на область действия. Расположите изображение приблизительно в центре области действия. Сцена должна выглядеть, как показано на рисунке.
28. Выберите кадр 30 слоя `background` и вставьте ключевой кадр, используя команду `Keyframe` меню `Insert`. Теперь объясним, почему при создании клипа мы оставили первый ключевой кадр слоя `flames` без изображения пламени. Если бы этого не было сделано, при помещении клипа на область действия ракета изображалась бы с работающими двигателями. Но использование пустого ключевого кадра позволяет создать эффект запуска двигателей при старте.
29. Выберите кадр 30 слоя `ground` и вставьте кадр, используя команду `Frame` меню `Insert` или нажав клавишу `F5`.



30. Теперь выберите кадр 30 слоя `rocket ship` и вставьте ключевой кадр командой `Keyframe` меню `Insert`.
31. На этом этапе выберите кадр 30 слоя `rocket ship` и переместите изображение ракеты вверх за пределы области действия.
32. Выберите первый ключевой кадр слоя `rocket ship` и создайте промежуточные кадры анимации, выбрав команду `Create Motion Tween` меню `Insert`.
33. Чтобы старт ракеты выглядел более реально, выберите первый ключевой кадр слоя `rocket ship`, откройте панель `Properties`, используя команду `Properties` меню `Window`, и установите в поле `Ease` (Замедлять ход) значение `-100`. При этом создается эффект медленного старта и постепенного ускорения движения ракеты при подъеме.
34. Сохраните файл, выбрав команду `Save` меню `File`. Когда появится диалоговое окно `Save`, выберите каталог, где вы хотите сохранить файл, введите имя `rocket_animation` в поле `Name` и щелкните на кнопке `Save`.
35. Для проверки сцены, нажмите клавишу `Return/Enter`. Если вы хотите, чтобы при тестировании воспроизводился клип, выберите команду `Test Movie` меню `Control`.

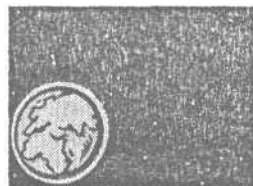
Отправление и удаление от Земли

Теперь можно приступать к созданию второй сцены анимации. Вы продолжите космическое путешествие и разработаете сцену, в которой ракета уходит далеко в космическое пространство, все дальше и дальше удаляясь от Земли. Здесь потребуется проявить мастерство для создания иллюзии расстояния и глубокого космического пространства.

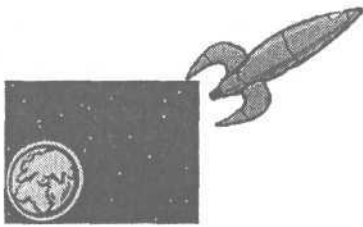
1. Откройте файл `rocket_animation`.
2. Выберите команду `Scene` меню `Window`, чтобы открыть панель `Scene`. Добавьте еще одну сцену, щелкнув на кнопке `Add Scene` (в нижней части панели `Scene`) либо выбрав команду `Scene` меню `Insert`.
3. Выполните двойной щелчок на имени созданной сцены и назовите ее `departure` (отправление).
4. Выберите сцену `departure`.
5. Чтобы сделать изображение космического пространства темным, откройте меню `Modify` и выберите команду `Document`. Щелкните на образце цвета фона `Background Color` и с помощью открывшейся палитры цветов выберите черный цвет.
6. Теперь необходимо создать изображение звезд, разбросанных в космическом пространстве. Выберите команду `New Symbol` меню `Insert`. Когда откроется диалоговое окно `Create New Symbol`, введите в поле `Name` имя `star` (звезда), установите переключатель типа символа в положение `Graphic` и щелкните на кнопке `OK`.
7. После переключения в режим редактирования символа выберите инструмент `Ellipse` и нарисуйте небольшой кружок желтого цвета. Убедитесь, что его обводка также желтая. Как обычно, вам предоставляется возможность проявить

фантазию — сделать изображения звезд разнообразными, создать другие космические объекты и т. д. Все зависит от вашего воображения.

8. Завершив создание изображения звезды, которое в дальнейшем будет многократно использовано в сцене, возвратитесь к главной временной шкале.
9. На данном этапе необходимо импортировать графическое изображение Земли, которое вы будете использовать. Выберите команду New Symbol меню Insert. В открывшемся диалоговом окне Create New Symbol введите в поле Name имя earth (Земля), установите переключатель типа символа в положение Graphic и щелкните на кнопке ОК.
10. После перехода в режим редактирования символа откройте диалоговое окно Import, выбрав команду Import меню File. В этом окне найдите и откройте на компакт-диске каталог Hands On 3, выберите файл earth.swf и щелкните на кнопке Open.
11. Используйте кнопку Edit Scene для перехода к сцене departure. Данная кнопка расположена в правой части строки Scene and Symbol.
12. Присвойте единственному слою имя stars (звезды).
13. Откройте окно библиотеки, активизировав команду Library меню Window, выберите первый ключевой кадр и с помощью мыши несколько раз (количество звезд определите самостоятельно) перетащите символ с изображением звезды на область действия.
14. Теперь добавьте еще один слой, выбрав команду Layer меню Insert или щелкнув на кнопке Insert Layer, расположенной в правом нижнем углу временной шкалы. Выполните двойной щелчок на имени слоя и введите имя earth. В качестве альтернативы выполните двойной щелчок на значке слоя, расположенном слева от его имени. При этом откроется диалоговое окно Layer Properties, где в поле Name нужно ввести имя earth.
15. С помощью мыши перетащите символ earth (с изображением Земли) из окна библиотеки на область действия в левый нижний угол. На данном этапе сцена должна выглядеть, как показано на рисунке.
16. Создайте третий слой и назовите его rocket ship.
17. Выберите первый ключевой кадр слоя rocket ship, перетащите клип rocket ship из окна библиотеки на область действия. (Данный клип был создан в предыдущем разделе.)
18. Размер и ориентация клипа не подходят для данной сцены, поэтому необходимо внести некоторые изменения. Прежде всего, выберите экземпляр символа с изображением ракеты и зеркально отразите его по горизонтали, выполнив последовательность команд Modify ▶ Transform ▶ Flip Horizontal.
19. На панели Tools выберите инструмент Free Transform и в секции Option нажмите кнопку Rotate and Skew (Вращение и наклон).
20. Поверните изображение ракеты приблизительно на 45° по часовой стрелке. В качестве альтернативы используйте панель Transform, открыть которую можно с помощью последовательности команд Window ▶ Panels ▶ Transform.



Установите переключатель в положение Rotate (Вращение) и введите в поле Rotate Value (Угол вращения) значение 45.

21. Убедитесь, что выбран первый ключевой кадр. Уменьшите изображение ракеты с помощью инструмента Free Transform или панели Transform. Зрителю должно казаться, что в начале полета ракета находится далеко. Во время движения она будет увеличиваться в размерах, как бы приближаясь из глубины пространства.
22. Используя инструмент Animate, переместите экземпляр символа с уменьшенным изображением ракеты ближе к центру изображения Земли.
23. После помещения на сцену всех необходимых элементов можно приступить к созданию анимации. Выберите кадр 35 слоя stars и вставьте кадр, активизировав команду Frame меню Insert.
24. Выберите кадр 35 слоя earth и вставьте кадр, используя ту же команду.
25. Теперь выберите кадр 35 слоя rocket ship и вставьте ключевой кадр командой Keyframe меню Insert.
26. Выберите только что созданный ключевой кадр слоя rocket ship. С помощью мыши перетащите изображение ракеты в правый верхний угол за пределы области действия.
27. Используя инструмент Free Transform или панель Transform, увеличьте изображение ракеты таким образом, чтобы оно стало несколько больше исходного. Убедитесь, что после увеличения изображение полностью находится за пределами области действия. Все эти действия приведут к созданию эффекта удаления ракеты от Земли. На данном этапе кадр 35 должен выглядеть, как показано на рисунке.
 
28. Выберите первый ключевой кадр слоя rocket ship и создайте промежуточные кадры движения ракеты, используя команду Create Motion Tween меню Insert. При этом будет создана анимация, изображающая полет космической ракеты, которая, увеличиваясь в размере, приближается к зрителю и пролетает мимо. Однако (в зависимости от положения ракеты в конечном ключевом кадре) может получиться так, что направление носа ракеты не совпадет с направлением ее движения во время полета, и такая анимация будет выглядеть несколько забавно. Для решения данной проблемы следует создать направляющую движение.
29. Выберите первый ключевой кадр слоя rocket ship и добавьте направляющую движение, щелкнув на кнопке Add Motion Guide, расположенной в нижнем левом углу временной шкалы.
30. Выберите первый ключевой кадр слоя motion guide (направляющая движение) и с помощью инструмента Line создайте траекторию движения ракеты: от середины изображения Земли и за пределы правого верхнего угла области действия. Фоновый цвет данной сцены черный, поэтому для линии выберите яркий цвет, используя образец цвета обводки Stroke Color, либо сделайте линию

черного цвета, но при этом временно измените фоновый цвет документа, выбрав команду **Modify** ▶ **Document** и щелкнув на образце цвета фона **Background Color**.

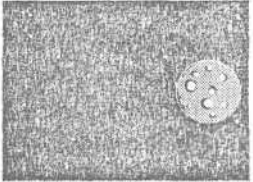
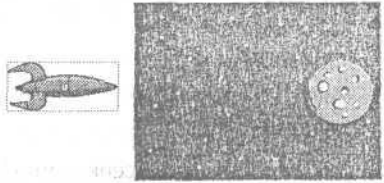
31. Выберите первый ключевой кадр слоя `rocket ship` и с помощью инструмента **Aggow** переместите изображение ракеты таким образом, чтобы точка регистрации символа совпала с началом траектории движения. Поскольку изображение ракеты в первом ключевом кадре небольшое, вам может понадобиться инструмент **Zoom**.
32. Выберите последний ключевой кадр слоя `rocket ship` и, используя инструмент **Aggow**, переместите изображение ракеты таким образом, чтобы точка регистрации совпала с окончанием траектории движения. Если у вас возникнут трудности при совмещении точки регистрации экземпляра символа с направляющей движения, проверьте, активизирована ли команда **Snap to Objects** (Привязка к объектам) меню **View**. Кроме того, прежде чем перемещать объект, удерживайте левую кнопку мыши нажатой в течение нескольких секунд.
33. Выберите первый ключевой кадр слоя и откройте панель **Properties**, активизировав одноименную команду меню **Window**.
34. Установите флажок **Orient to Path** и введите значение `-100` в поле **Ease**. Это приведет к тому, что ракета всегда будет ориентирована соответственно траектории полета. Кроме того, в начале движения скорость ракеты будет меньше, чем в конце, а это создаст иллюзию пространства.
35. Теперь для создания эффекта удаления необходимо анимировать изображение Земли. Щелкните на кадре 35 слоя `earth` и вставьте ключевой кадр, выбрав команду **Keyframe** меню **Insert**.
36. Убедитесь, что выбран только что созданный ключевой кадр. Используя инструмент **Aggow**, выберите и переместите изображение Земли в левый нижний угол за пределы области действия. С помощью инструмента **Free Transform** или панели **Transform** уменьшите размеры изображения Земли в несколько раз. При просмотре анимации у зрителя должно создаваться впечатление, что Земля удаляется.
37. Выберите первый ключевой кадр слоя `earth` и, используя команду **Create Motion Tween** меню **Insert**, создайте промежуточные кадры.

Создание второй сцены завершено. Для ее проверки нажмите клавишу **Return/Enter**. Если вы хотите, чтобы при тестировании воспроизводился клип, активизируйте команду **Test Scene** меню **Control**. Выбрав команду **Test Movie** этого же меню, можно просмотреть весь фильм, включая первую сцену.

Путешествие в космосе

Теперь можно приступить к созданию новой сцены, в которой космический корабль путешествует в космосе.

1. Откройте файл `rocket_animation`.
2. Выбрав команду **Scene** меню **Window**, откройте панель **Scene** и добавьте третью сцену, щелкнув на кнопке **Add Scene** (в нижней части панели **Scene**) либо выбрав команду **Scene** меню **Insert**.

3. Выполните двойной щелчок на имени только что созданной сцены и переименуйте ее в travel (путешествие).
4. Выберите пока еще единственный слой на временной шкале и присвойте ему имя stars (звезды). Выберите кадр 30 и командой Frame меню Insert вставьте кадр.
5. Выберите первый ключевой кадр слоя stars, откройте окно библиотеки и с помощью мыши перетащите несколько раз символ с изображением звезды на область действия, как при разработке предыдущей сцены.
6. Теперь необходимо создать изображение планеты, рядом с которой будет пролетать космический корабль. Выберите команду New Symbol меню Insert. Когда откроется диалоговое окно Create New Symbol, установите переключатель типа символа в положение Graphic, введите в поле Name имя planet 1 и щелкните на кнопке ОК.
7. Переключившись в режим редактирования символа, откройте меню File и выберите команду Import. После появления одноименного диалогового окна найдите и откройте каталог Hands On 3 на компакт-диске, выберите файл planet 1.swf и щелкните на кнопке Open.
8. Переместите изображение к его точке регистрации в центре окна редактора символа.
9. Выйдите из режима редактирования символа и перейдите на основную временную шкалу. Создайте новый слой, щелкнув на кнопке Insert Layer или выбрав команду Layer меню Insert, и присвойте ему имя planet.
10. Выберите первый ключевой кадр слоя planet и с помощью мыши перетащите символ planet 1 с изображением планеты из окна библиотеки на область действия. Сцена должна выглядеть, как показано на следующем рисунке.
 
11. Теперь необходимо анимировать изображение планеты. Выберите кадр 30 слоя planet и вставьте ключевой кадр, используя команду Keyframe меню Insert.
12. Выберите только что созданный ключевой кадр и с помощью инструмента Animate переместите изображение планеты влево примерно на 4 сантиметра. Далее щелкните на первом ключевом кадре слоя planet и выберите команду Create Motion Tween меню Insert. Будет создан эффект полета ракеты на большой скорости.
13. Создайте новый слой и назовите его rocket ship.
14. Выберите первый ключевой кадр слоя rocket ship и с помощью мыши перетащите клип rocket ship из окна библиотеки на рабочую область слева от области действия. С помощью инструмента Free Transform или панели Transform расположите ракету горизонтально. На данном этапе сцена должна выглядеть, как показано на рисунке.
 

15. Выберите кадр 30 слоя `rocket ship` и вставьте ключевой кадр, используя команду `Keyframe` меню `Insert`.
16. Щелкните на только что созданном ключевом кадре и переместите изображение ракеты на рабочую область справа от области действия.
17. Выберите первый ключевой кадр слоя `rocket ship` и, используя команду `Create Motion Tween` меню `Insert`, создайте промежуточные кадры движения ракеты.

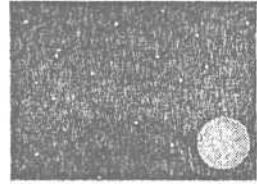
Вы завершили создание анимации, в которой космический корабль путешествует в космосе. Для проверки сцены нажмите клавишу `Return/Enter`. Если вы хотите, чтобы при тестировании воспроизводился клип, активизируйте команду `Test Scene` меню `Control`. Выбрав команду `Test Movie` этого же меню, можно просмотреть весь фильм, включая первую и вторую сцены.

Приближение к планете X

Теперь вам предстоит создать сцену, в которой космический корабль приближается к конечному пункту своего путешествия — планете X.

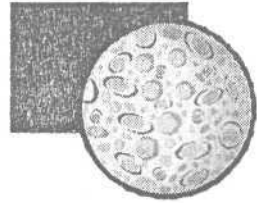
1. Откройте файл `rocket_animation`.
2. Выбрав команду `Scene` меню `Window`, откройте панель `Scene` и добавьте четвертую сцену, щелкнув на кнопке `Add Scene` (в нижней части панели `Scene`) либо активизировав команду `Scene` меню `Insert`.
3. Выполните двойной щелчок на имени только что созданной сцены и переименуйте ее в `descent` (снижение).
4. Выберите пока еще единственный слой на временной шкале и присвойте ему имя `stars` (звезды). Выберите кадр 30 и, используя команду `Frame` меню `Insert`, вставьте кадр.
5. Щелкните на первом ключевом кадре слоя `stars`, откройте окно библиотеки и с помощью мыши перетащите несколько раз символ с изображением звезды на область действия, как при создании предыдущих сцен.
6. На данном этапе необходимо создать изображение планеты X. Выберите команду `New Symbol` меню `Insert`. Когда откроется диалоговое окно `Create New Symbol`, установите переключатель типа символа в положение `Graphic`, введите в поле `Name` имя `planet 2` и щелкните на кнопке `OK`.
7. Переключившись в режим редактирования символа, откройте меню `File` и выберите команду `Import`. После появления одноименного диалогового окна найдите и откройте каталог `Hands On 3` на компакт-диске, выберите файл `planet 2.swf` и щелкните на кнопке `Open`.
8. Переместите изображение к его точке регистрации в центре окна редактора символа.
9. Выйдите из режима редактирования символа и перейдите на основную временную шкалу. Создайте новый слой, щелкнув на кнопке `Insert Layer` или выбрав команду `Layer` меню `Insert`, и присвойте ему имя `planet x`.
10. Выберите первый ключевой кадр созданного слоя и с помощью мыши перетащите символ `planet 2` с изображением планеты из окна библиотеки в правый

нижний угол на область действия. Поскольку изображение планеты слишком большое, уменьшите его с помощью инструмента Free Transform или панели Transform до размера примерно 123x123 пиксела. На данном этапе сцена должна выглядеть, как показано на рисунке.



12. Выберите кадр 30 слоя planet x и вставьте ключевой кадр, используя команду Keyframe меню Insert.

13. Щелкните на последнем ключевом кадре только что созданного слоя planet 2, выберите экземпляр символа planet 2 и с помощью инструмента Free Transform или панели Transform увеличьте размер изображения, как показано на следующем рисунке.

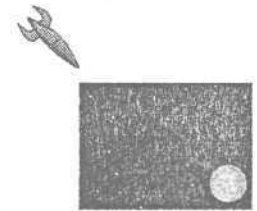


14. Затем щелкните на первом ключевом кадре слоя planet x и выберите команду Create Motion Tween меню Insert.

15. Создайте новый слой и присвойте ему имя rocket ship.

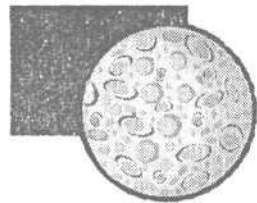
16. Выберите первый ключевой кадр этого нового слоя и с помощью мыши перетащите клип rocket ship из окна библиотеки на рабочую область, расположенную слева сверху по отношению к области действия.

17. Используя инструмент Free Transform или панель Transform, поверните изображение ракеты таким образом, чтобы ее нос был направлен к планете. На данном этапе сцена должна выглядеть, как показано на рисунке ниже.



18. Выберите кадр 30 слоя rocket ship и командой Keyframe меню Insert вставьте ключевой кадр.

19. Щелкните на только что созданном ключевом кадре и поместите изображение ракеты над изображением планеты. Используя инструмент Free Transform или панель Transform, уменьшите изображение ракеты. Теперь сцена должна выглядеть, как показано на рисунке.



20. Щелкните на первом ключевом кадре слоя rocket ship и выберите команду Create Motion Tween меню Insert.

21. Вновь, как при создании первой сцены, может возникнуть несоответствие между направлением носа ракеты и направлением ее движения во время полета. Для устранения этого недостатка опять создайте направляющую движения.

22. Выберите первый ключевой кадр слоя rocket ship и щелчком на кнопке Add Motion Guide, расположенной в нижнем левом углу временной шкалы, добавьте направляющую движения.

23. Щелкните на первом ключевом кадре слоя motion guide (направляющая движения) и с помощью инструмента Line создайте траекторию движения ракеты.

24. Выберите первый ключевой кадр слоя rocket ship и с помощью инструмента Animate переместите изображение ракеты так, чтобы точка регистрации символа совпала с началом траектории движения.
25. Затем выберите последний ключевой кадр слоя rocket ship и, пользуясь этим же инструментом, переместите изображение ракеты таким образом, чтобы точка регистрации совпала с окончанием траектории движения.
26. Теперь щелкните на первом ключевом кадре слоя rocket ship и откройте панель Properties, выбрав одноименную команду меню Window.
27. Установите флажок Orient to Path и введите значение -100 в поле Ease. Теперь ракета всегда будет ориентирована соответственно траектории полета. Кроме того, в начале движения скорость ракеты будет больше чем в конце, что создаст иллюзию глубокого пространства.

Вы завершили создание анимации, в которой космический корабль приближается к планете X. Для проверки сцены нажмите клавишу Return/Enter. Если вы хотите, чтобы при тестировании воспроизводился клип, активизируйте команду Test Scene меню Control. Выбрав команду Test Movie этого же меню, можно посмотреть весь фильм.

Конечная сцена

Итак, первые четыре сцены созданы. В последней сцене вы создадите эффект, присущий голливудским фильмам.

1. Откройте файл rocket_animation.
2. Выбрав команду Scene меню Window, откройте панель Scene и добавьте пятую сцену, щелкнув на кнопке Add Scene (в нижней части панели Scene) либо активизировав команду Scene меню Insert.
3. Выполните двойной щелчок на имени только что созданной сцены и переименуйте ее в finale (конец).
4. Выберите команду New Symbol меню Insert.
5. Когда откроется диалоговое окно Create New Symbol, установите переключатель типа символа в положение Graphic, введите в поле Name имя To Be Continued и щелкните на кнопке OK.
6. На панели Tools выберите инструмент Text.
7. До ввода каких-либо символов откройте панель Properties, выбрав одноименную команду меню Window.
8. На панели Properties откройте список Font, выберите подходящий шрифт и задайте его размер в поле Size.
9. Поскольку фоновый цвет фильма черный, для символов надписи выберите один из светлых цветов, например белый.
10. Щелкните на области действия и введите надпись To Be Continued.

11. Переместите указатель мыши за пределы текстового поля и выполните щелчок. С помощью инструмента **Align** выровняйте текстовое поле в окне редактора символа. Если края символов получатся ступенчатыми, выберите текст, откройте меню **View** и активизируйте команду **Antialias Text** (Сглаживать текст).
12. Теперь создайте эффект плавного исчезновения надписи. Для этого переключитесь на основную временную шкалу и выберите первый ключевой кадр единственного слоя. (Можете присвоить слою какое-нибудь информативное имя, но поскольку в данной сцене используется только один слой, в этом нет необходимости.)
13. С помощью мыши перетащите символ с надписью **To Be Continued** из окна библиотеки на центр области действия.
14. Щелкните на кадре 15 и вставьте ключевой кадр, активизировав команду **Keyframe** меню **Insert**.
15. Выберите только что созданный ключевой кадр и откройте панель **Properties**, активизировав одноименную команду меню **Window**.
16. Откройте список **Color** (Цвет) и выберите пункт **Alpha** (Прозрачность). В поле **Alpha Value** введите значение 0.
17. Щелкните на первом ключевом кадре и выберите команду **Create Motion Tween** меню **Insert**. При этом будет создан эффект медленного исчезновения текста надписи.

Если вы хотите, чтобы после воспроизведения всех сцен фильм не повторялся циклически, а был остановлен, добавьте действие кадра **stop()**. Более подробно о действиях кадра будет рассказано в главе 12.

Поздравляем! Вы завершили создание последней сцены анимации. Для просмотра созданного фильма целиком выберите команду **Test Movie** меню **Control**.

ЧАСТЬ 4

Интерактивные фильмы

В дискуссиях, посвященных созданию цифровых мультимедийных приложений, часто можно услышать термин интерактивность. Данное слово нередко встречается в книгах, учебных пособиях и статьях, а также часто используется для оценки и определения категории создаваемого мультимедийного продукта. Указанный термин начал приобретать характеристики модного, всеми употребляемого слова. Создается угроза, что такое частое употребление может привести либо к потере первоначального смысла, либо к его искажению.

Так что же такое интерактивность? Словарь Merriam-Webster's Collegiate Dictionary дает определение интерактивности как «двухсторонней электронной системе связи... в которой в качестве информации передаются команды или ответы пользователя». Для начала нашей дискуссии такое определение вполне подходит — двухсторонняя связь между человеком и машиной. Однако какое это имеет отношение к Flash? В данном случае вам как создателю Flash-фильмов отводится главная роль, поскольку именно разработчик определяет характер и параметры связи между фильмами и зрительской аудиторией. Ваша задача состоит в том, чтобы определить, как именно фильм будет отвечать на запросы пользователя. В данной части книги разговор пойдет об интерактивных свойствах создаваемых фильмов. Изучая возможности ActionScript, вы познакомитесь со множеством техник, обеспечивающих связь между аудиторией и фильмом Flash.

Глава 12. Базовые действия ActionScript.....	222
Глава 13. Создание интерактивных элементов управления.....	247

Глава 12

Базовые действия ActionScript

До настоящего момента вы использовали Flash как инструмент для создания быстро загружаемой графики и анимации. Однако с помощью Flash можно превратить графические элементы в динамические компоненты игр, учебных фильмов и интерактивных интерфейсов. Такие функциональные возможности фильмов Flash обеспечиваются благодаря использованию действий.

Действие — это набор команд, написанных на языке ActionScript, который является объектно-ориентированным языком сценариев Flash-фильмов. Действия указывают Flash, как именно должен воспроизводиться фильм. Действия могут быть как простыми (например, переход к определенному кадру и его воспроизведение), так и достаточно сложными (допустим, выполнение вычислений и принятие решений на основе полученного результата).

В этой главе рассматриваются следующие темы:

- ▶ знакомство с панелью Actions;
- ▶ работа с действиями объектов;
- ▶ работа с действиями кадра;
- ▶ связывание действий с объектами;
- ▶ связывание действий с кадрами;
- ▶ использование действий для создания основных элементов управления воспроизведением фильма.

Действия

Действия (Actions) — это одни из самых динамичных функций Flash, имеющиеся в распоряжении разработчика. Действия используются для создания интерактивных элементов фильма и позволяют зрителям управлять ходом воспроизведения фильма при помощи мыши или ввода соответствующих команд с клавиатуры. Используя действия, разработчик может установить определенный порядок воспроизведения сцен фильма, учитывая реакцию аудитории.

Чтобы использовать действия ActionScript, вам не нужно быть программистом. Безусловно, при создании сложных сценариев опыт программирования окажется полезным. Однако при написании сценариев ActionScript предыдущий опыт не является обязательным. Для выбора действий и связывания их с объектами фильма создатели Flash предусмотрели простой и понятный интерфейс — панель Actions. С помощью данной панели вы получаете доступ к библиотеке команд ActionScript, что позволяет вам значительно экономить время при вводе кода сценариев.

Таким образом, ActionScript — это объектно-ориентированный язык программирования, являющийся «родным» для среды Flash. В данном контексте термин

объектно-ориентированный означает, что сценарии связываются с «объектами» или элементами фильма. Если во время воспроизведения фильма эти объекты принимают сообщение об определенном событии, их сценарии (или действия) выполняются. Например:

```
on(press){
    gotoAndPlay("nextStop")
}
```

В этом сценарии при наступлении события щелчка мыши `press` головка воспроизведения перейдет к кадру с меткой `nextStop` и воспроизведение продолжится с этого места. Другими словами, событие вызывает действие, которое изменяет порядок воспроизведения фильма.

Связь между событиями и действиями является главной особенностью объектно-ориентированного программирования. Если у вас нет опыта написания сценариев либо вы совсем *незнакомы* с программированием, такие формулировки могут вас несколько смутить. Однако не стоит впадать в унынье. Как только вы познакомитесь с действиями, синтаксис и структура языка `ActionScript` станут для вас менее загадочными.

Чтобы использовать действия для написания сценариев фильмов `Flash`, необходимо прежде всего изучить панель `Actions`.

Знакомство с панелью `Actions`

Панель `Actions` (Действия) — это интерфейс создания сценариев, которые будут выполняться во время воспроизведения `Flash`-фильма. Работать с панелью `Actions` можно в двух режимах: `Normal` (Обычный) и `Expert` (Эксперт). В режиме `Normal` пользователю доступны некоторые автоматические функции создания кода, поэтому данный режим рекомендуется для новичков. Работая в режиме `Normal`, вы можете создавать действия и добавлять их в фильм посредством выбора соответствующих пунктов меню и перетаскивания элементов с помощью мыши. Режим `Expert` больше подходит для опытных разработчиков. В данном режиме панель `Actions` больше похожа на окно текстового редактора, в котором пользователь самостоятельно вводит код сценариев.

Чтобы открыть панель `Actions`, выберите команду `Actions` меню `Window` или нажмите клавишу `F9`.

MX Многие улучшения, появившиеся в новой версии `Flash MX`, связаны именно с панелью `Actions` (рис. 12.1). Данная панель теперь обладает рядом новых возможностей, которые значительно облегчают написание сценариев.

Ниже приведен перечень элементов панели `Actions`:

Область элементов `ActionScript` содержит раскрывающийся список категорий, с помощью которого вы получаете доступ к библиотеке `ActionScript`. Выбрав соответствующее действие, вы можете с помощью мыши перетащить его на область отображения сценария.

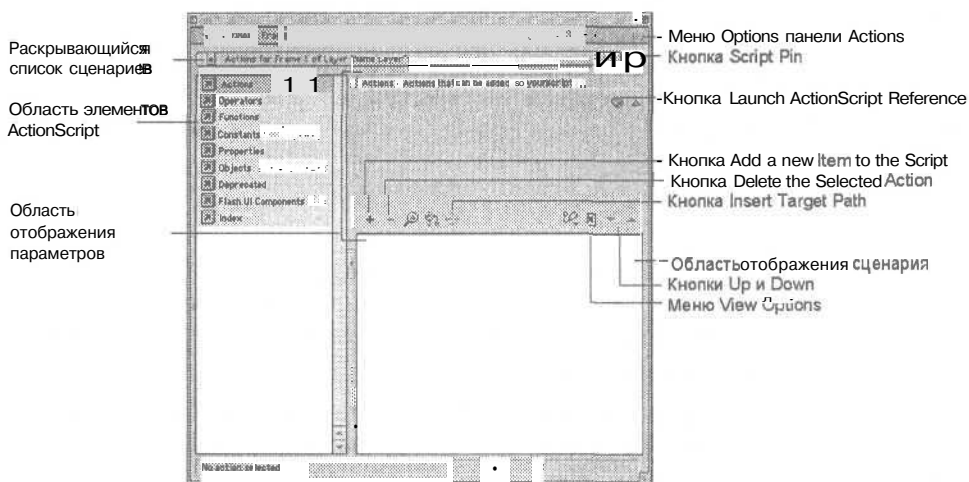


Рис. 12.1. Панель Actions

Область отображения сценария служит для отображения всех строк-операторов текущего сценария. В режиме Эксперт данная область используется в качестве окна ввода сценария.

Область отображения параметров служит для ввода специфической информации (например, номера кадра или имени сцены), которая будет вставлена в сценарий. Область отображения параметров доступна для ввода только в режиме Normal. Информация, отображаемая в данной области, может быть свернута или развернута посредством щелчка на небольшом треугольнике в правой верхней части панели Actions.

Кнопки Add a new Item to the Script (Добавить новый элемент сценария) и Delete the Selected Action (Удалить выбранное действие) используются для добавления и удаления строк-операторов сценария ActionScript. При щелчке на кнопке Add a new Item to the Script открывается меню с элементами ActionScript. При щелчке на кнопке Delete the Selected Action удаляются все выделенные строки-операторы сценария.

Кнопки Up (Вверх) и Down (Вниз) используются для перемещения строк-операторов в пределах окна отображения сценария.

Кнопка Insert Target Path (Указать путь к объекту) служит для доступа к диалоговому окну Insert Target Path. В данном окне отображается структура расположения клипов в вашем фильме. Вы можете выбрать экземпляр клипа (или кнопки), которым вы собираетесь управлять с помощью действий.

MX Ниже перечислены элементы панели Actions, появившиеся в новой версии Flash.

Раскрывающийся список сценариев позволяет переключаться между сценариями, связанными с различными элементами фильма. При выборе на области действия элемента, которым можно управлять с помощью сценария, панель Actions автоматически переходит к отображению сценария, связанного с этим элементом. Например, если выбран кадр, то в поле списка будет отображаться номер этого кадра,

а также имя и номер слоя, к которому относится данный кадр. Если на области действия выбран экземпляр кнопки или клипа, то в поле списка будет указано имя экземпляра, а также название родительского символа.

Меню View Options (Параметры отображения окна) позволяет переключаться между режимами Normal и Expert, а также отображать или скрывать линейку с нумерацией строк сценария. Более подробно о нумерации строк сценария мы расскажем в главе 14.

Кнопка Launch ActionScript Reference позволяет получить доступ к окну панели Reference (Описание), с помощью которой можно получить справку и найти описание всех встроенных элементов языка ActionScript. Более подробно о работе с панелью Reference мы расскажем в главе 14.

Кнопка Script Pin используется для фиксации отображаемого сценария на панели Actions. Данное свойство позволяет отображать один сценарий на панели Actions при выборе различных элементов на области действия. Напомним, что в предыдущих версиях Flash на панели Actions мог отображаться только сценарий выбранного в настоящий момент объекта.

Создание сценариев в режиме Normal

Как уже отмечалось ранее, с панелью Actions можно работать в двух режимах: Normal и Expert. В зависимости от сложности задания и уровня вашей подготовки выберите режим, который вам больше подходит. В табл. 12.1 представлены основные различия между двумя режимами работы с панелью Actions.

Таблица 12.1. Поддерживаемые свойства при работе в режимах Normal и Expert

<u>Поддерживаемое свойство</u>	<u>Режим Normal</u>	<u>Режим Expert</u>
Поддержка функции перетаскивания из области элементов ActionScript на область отображения сценария	+	+
Использование области отображения параметров для ввода данных	+	–
Доступ к кнопкам добавления и удаления строк-операторов сценария	+	Доступна только кнопка добавления строк-операторов
Доступ к кнопкам перемещения строк-операторов	+	–
Доступ к кнопке Insert Target Path	+	+
Использование раскрывающегося списка сценариев	+	+
Доступ к кнопке Script Pin	+	+

В общем, в режиме Expert ответственность за корректное написание сценария возлагается в основном на разработчика. И хотя некоторые функции остаются недоступными, для опытных пользователей режим Expert обеспечивает больше гибкости при написании сценариев. Более подробно о режиме Expert мы расскажем в главе 14.

Режим Normal несколько ограничивает возможности разработчика, поскольку весь сценарий создается посредством выбора элементов меню с автоматическим вводом кода. Такая техника не требует глубоких знаний языка ActionScript и больше подходит для начинающих пользователей.

ПРИМЕЧАНИЕ —

Средства Flash позволяют создавать сценарии, не содержащие ошибок. Кроме того, в процессе работы вы можете переключаться между режимами Normal и Expert, когда это необходимо.

Чтобы установить режим Normal панели Actions, выполните следующие действия:

1. Выберите команду Actions меню Window.
2. Выполните одно из следующих действий.
 - ▷ Выполните щелчок на значке меню Options в верхнем правом углу панели Actions и выберите команду Normal Mode (Обычный режим).
 - ▷ Воспользуйтесь комбинацией клавиш Cmd+Shift+N (Macintosh) или Ctrl+Shift+N (Windows).
3. Откройте меню View Options и выберите команду Normal Mode.

Работа с действиями объектов

Во Flash-фильмах действия могут быть связаны с объектами двух типов: кнопками и клипами. Связанное действие выполняется тогда, когда объект принимает сообщение о выполнении события. Для кнопок эти сообщения в основном поступают в виде сигналов от мыши или клавиатуры. Сообщения о событии клипа появляются в результате любых изменений или взаимодействий, которые произошли с самим клипом. Когда вы связываете действие с объектом и создаете сценарий, вы определяете источник события для каждого объекта.

Действия связываются с кнопками и клипами, поскольку эти элементы являются интерактивными компонентами Flash-фильма. Как объекты, они могут принимать события мыши, данные и другого типа информацию во время воспроизведения фильма. Структура, создаваемая с помощью объектов и связанных с ними действий, будет определять способ взаимодействия аудитории с фильмом.

Важно помнить, что кнопки и клипы хранятся в библиотеке Flash в виде *символов*. Каждый раз при использовании символов в фильме фактически используется отдельная копия символа, или его *экземпляр*. (Более подробно об экземплярах символов мы рассказывали в главе 7.)

Экземпляр можно рассматривать как копию символа. Любое изменение, которое произведено с экземпляром, никак не повлияет на родительский символ или на другие экземпляры этого символа. Тот же принцип сохраняется и при связывании действий с экземпляром символа. Любое действие, ассоциированное с экземпляром символа, будет выполняться только тогда, когда в фильме используется данный экземпляр. Это означает, что действия связываются только с определенным экземпляром символа.

Однако любое действие выполняется только тогда, когда в фильме происходит определенного рода событие. В языке `ActionScript` (и во многих других объектно-ориентированных языках) действия управляются *обработчиками событий*.

Обработчики событий

Объектно-ориентированные языки работают по схеме причина-следствие, где действие выступает в роли «следствия» или результата, произведенного сценарием, а обработчик события является «причиной», которая инициирует действие.

Например, рассмотрим следующий сценарий:

```
on(thisEvent){
    doAnAction( );
}
```

Каждый, кто хоть немного знаком с языком `ActionScript`, знает, что нет события с именем `thisEvent` и нет действия с именем `doAction`. Данный фрагмент кода — всего лишь демонстрация конструктора для последовательности событий и действий. Обработчик события — это практически всегда первая строка-оператор. Имени обработчика предшествует слово `on`, которое можно интерпретировать как фразу «когда произойдет событие...». Обработчик составляет первую часть предложения, содержащего инструкцию для Flash-фильма. А действие в этой инструкции определяет реакцию фильма на произошедшее событие. Полный сценарий можно описать предложением: «Когда произойдет событие `thisEvent`, выполнить действие `doAction`». В языке `ActionScript` существует множество различных типов обработчиков событий. Специфику каждого из них мы обсудим позже в этой главе.

Теперь вы знаете, что собой представляет обработчик события. Однако важно также понимать и суть самого события, когда и как оно происходит в контексте фильма. Как разработчику интерактивного продукта, вам необходимо знать и учитывать влияние каждого события на воспроизведение фильма. Например, если бы вы создавали интерактивный музыкальный микшер, какой бы обработчик события вы выбрали для кнопок, предназначенных для выключения и включения звуковых дорожек — нажатие кнопки мыши или отпускание кнопки мыши? Мы бы рекомендовали использовать событие нажатия кнопки мыши, поскольку речь идет о музыкальном микшере, где важно обеспечить пользователю возможность синхронизировать звуковые дорожки и поддерживать устойчивый ритм. При использовании события нажатия кнопки мыши пользователю будет легче работать, поскольку будет создаваться впечатление игры на музыкальном инструменте.

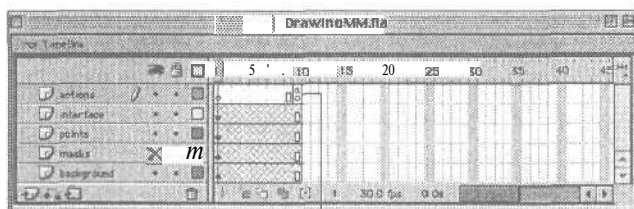
Таким образом, для эффективной работы с интерактивным фильмом для каждого действия необходимо выбирать наиболее подходящий обработчик события.

Работа с действиями кадра

Как вы можете догадаться, действия кадров — это действия, связанные с кадрами вашего Flash-фильма. Действия кадров можно связывать только с *ключевыми кадрами*. Ключевые кадры используются для обозначения «точек» фильма, в которых

происходят изменения как анимации, так и основных событий фильма. (Более подробно о ключевых кадрах речь шла в главе 9.)

Если с ключевым кадром связано действие, то такой кадр на временной шкале можно узнать по небольшой букве а (рис. 12.2).



Значок действия кадра

Рис. 12.2. Действие кадра на временной шкале

Посредством связывания действий с кадрами вы можете изменить способ воспроизведения фильма. Рассмотрим следующий пример:

```
gotoAndPlay(1);
```

Если вы свяжете это действие с кадром 10, будет создан бесконечный цикл воспроизведения кадров от 1 до 10. Такой режим будет продолжаться до тех пор, пока другое действие не изменит ход воспроизведения Flash-фильма.

Самое важное отличие действий кадров от действий объектов состоит в отсутствии обработчика события. Действия кадров не требуют специальных обработчиков, поскольку они выполняются при воспроизведении кадров во Flash-фильме.

Использование действий

Обсудив некоторые теоретические вопросы, можно переходить к практическому использованию действий. Чтобы использовать действия в фильме Flash, необходимо связать их с объектом или кадром. Данный процесс достаточно простой и включает выбор объекта или кадра, выбор необходимого действия и определение специальных параметров, которые могут требоваться для действия. Все действия влияют только на связанный с ними экземпляр символа объекта или единственный кадр, содержащий действия.

Действия используются для управления схемой воспроизведения, интерактивными свойствами, потоком и направлением воспроизведения фильма. Одним словом, с помощью действия можно точно определить порядок воспроизведения, а также реакцию фильма на пользовательский ввод. Благодаря действиям Flash меньше напоминает анимационную программу и в большей степени — среду разработок интерактивных мультимедийных приложений. Вот лишь некоторые возможности, которыми обладает Flash благодаря использованию действий.

- ▶ Создание сложных, интерактивных схем навигации.
- ▶ Изменение качества воспроизведения фильма.

- ▶ Связь с документами HTML, размещенными в Интернете, и обмен данными с приложениями сервера.
- ▶ Обмен данными с хост-приложениями или другими Flash-фильмами.
- ▶ Воспроизведение других фильмов в пределах главного фильма.

Связывание действий с объектами

Связывание действий с объектами, такими как кнопки или клипы, — процесс достаточно простой. Однако при его выполнении следует помнить о некоторых вещах.

- ▶ Прежде чем связывать действие, необходимо выбрать объект на временной шкале (или на области действия). Если ни один объект не выбран, все действия на панели Actions будут недоступными.
- ▶ Действия связываются только с конкретным экземпляром символа кнопки или клипа; все другие экземпляры объекта остаются неизменными.
- ▶ Важно произвести тестирование действия после его связывания. В режиме редактирования корректно могут работать только простые действия (такие как `goAndPlay()`). Чтобы протестировать действие, необходимо протестировать весь фильм. Выберите для этого команду Test Movie меню Control. (Более подробно о тестировании фильмов мы расскажем в главе 28.)

Для связывания действий с объектами в режиме Normal выполните следующие действия.

1. Выберите объект на временной шкале или на области действия.
2. Активизируйте команду Actions меню Window или нажмите клавишу F9.
3. Когда откроется панель Actions, выполните следующие действия:
 - > Выберите действие в меню категорий области элементов ActionScript.
 - > Установите параметры, требуемые для данного действия.
 - > Обработчик события для каждого действия устанавливается автоматически. Если необходимо, то на области отображения параметров вы можете изменить обработчик для данного действия.

Более подробно о том, как связывать действия с элементами управления фильма и устанавливать параметры первых, мы расскажем в разделе «Использование действий для создания основных элементов управления воспроизведением фильма».

События мыши

При связывании действия с экземпляром кнопки в режиме Normal Flash автоматически присваивает этому действию обработчик события мыши. Этот обработчик определяет, какое событие при воспроизведении фильма будет инициировать выполнение действия (или действий), содержащихся в сценарии. Обработчик события мыши определяется действием `on()`. В большинстве случаев оптимальным будет обработчик, установленный по умолчанию. Однако, если в этом есть необходимость, вы можете изменить обработчик события мыши.

Для установки или изменения обработчика события мыши, выполните следующие действия.

1. Выберите кнопку, с которой необходимо связать действие либо для которой необходимо изменить обработчик событий.
2. Откройте панель Actions и выполните одно из следующих действий.
 - Если вы хотите установить обработчик события мыши, выберите действие `on()` из категории Actions ▶ Movie Control.
 - Если вы хотите изменить обработчик события мыши для уже существующего действия, выберите строку-оператор, содержащую название события, и на области отображения параметров установите соответствующий флажок.

ПРИМЕЧАНИЕ

При работе в режиме Normal вы можете добавить действие в сценарий четырьмя различными способами. 1. Найдите действие в меню категорий области элементов ActionScript и выполните на его значке двойной щелчок. 2. С помощью мыши перетащите значок действия из области элементов ActionScript на область отображения сценария. 3. Щелкните на кнопке Add (+) и выберите действие в меню. 4. Введите с клавиатуры комбинацию символов, соответствующую требуемому действию. Для отображения набора клавиш, соответствующих каждому действию, в области элементов ActionScript панели Actions откройте меню Options названной панели и активизируйте команду View Esc Shortcut Keys.

3. В области параметров отображается список различных обработчиков события мыши. Установите флажок рядом с названием соответствующего обработчика, и ваш сценарий автоматически будет обновлен. На рис. 12.3 показан внешний вид панели Actions, когда в области отображения сценария выбрана строка-оператор, содержащая обработчик события.

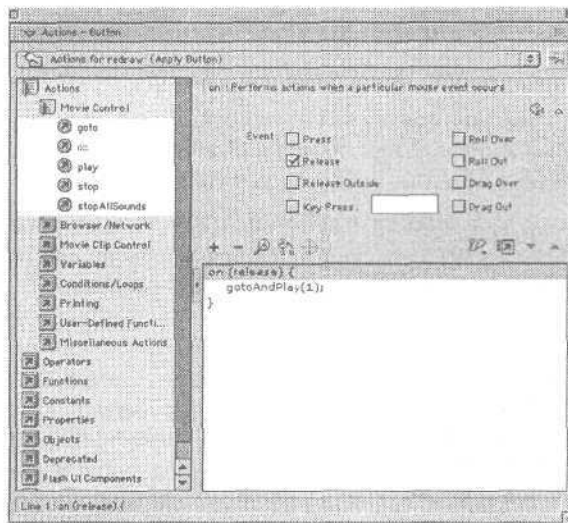


Рис. 12.3. Вид панели Actions, когда в области отображения сценария выбран обработчик события

Ниже представлены имена и описания событий мыши в языке ActionScript.

- ▶ **Press.** Событие `press` инициирует действие при нажатии кнопки, когда указатель мыши находится над объектом кнопки.
- ▶ **Release.** Событие `release` инициирует действие при отпускании кнопки, когда указатель мыши находится над объектом кнопки. Это событие принимается по умолчанию.
- ▶ **ReleaseOutside.** Событие `releaseOutside` инициирует действие при отпускании кнопки, когда указатель мыши *не находится* над объектом кнопки.
- ▶ **KeyPress.** Событие `keyPress` не связано с мышью. Оно инициирует действие, когда на клавиатуре будет нажата клавиша, указанная в поле слева от данного флажка.
- ▶ **Roll Over.** Событие `rollOver` инициирует действие, когда указатель мыши помещается на объект кнопки.
- ▶ **Roll Out.** Событие `rollOut` инициирует действие, когда указатель мыши помещается за пределы объекта кнопки.
- ▶ **DragOver.** Событие `dragOver` может показаться сначала несколько сложным. Фактически оно включает комбинацию из серии различных событий. Чтобы инициировать действие с помощью события `dragOver`, необходимо установить указатель мыши над объектом кнопки и выполнить щелчок, *переместить* указатель за пределы объекта, а затем снова поместить указатель на объект.
- ▶ **DragOut.** Чтобы инициировать действие с помощью события `dragOut`, выполните щелчок, когда указатель мыши находится над объектом кнопки, а затем переместите указатель за пределы объекта.

Помните, что любой из этих обработчиков событий может использоваться для инициации нескольких действий в одном сценарии. Кроме того, любой объект может иметь более одного связанного с ним обработчика. Например:

```
on(press){
    gotoAndPlay(1);
}
on(release){
    gotoAndPlay(2);
}
```

С помощью данного сценария при нажатии кнопки мыши будет воспроизводиться кадр 1, а когда кнопка будет отпущена — кадр 2.



Чтобы увидеть, как функционируют различные обработчики событий, просмотрите фильм `HandlerMan.swf`, который вы найдете на прилагаемом компакт-диске в каталоге Chapter 12.

События клипа

Клип — это еще один тип объекта, с которым можно связать действие. Поскольку поведение клипов во Flash-фильме отличается от поведения кнопок, клипы имеют свой собственный обработчик событий, инициирующий действия. Обработчик событий клипа инициализируется действием `onClipEvent()`. При выборе действий для клипа Flash автоматически производит вставку обработчика события, принятого по умолчанию. Однако, в отличие от обработчика события мыши,

обработчик события клипа принимает аргументы, которые определяют, какого рода событие клипа будет инициировать действие. Структура кода с обработчиком события клипа выглядит так:

```
onClipEvent (someMovieEvent){
    doAction(s);
}
```

Вместо слова `someMovieEvent` можно указать любое число аргументов. Ниже приведен список событий, которые можно использовать в качестве аргументов.

- ▶ **Load.** Событие `load` происходит, когда клип впервые вызывается на область действия, либо когда клип находится в текущем кадре временной шкалы.
- ▶ **Unload.** Событие `unload` происходит, когда воспроизведение клипа завершено и головка воспроизведения переходит к следующему кадру, либо при удалении клипа с области действия.
- ▶ **EnterFrame.** Событие `enterFrame` происходит при воспроизведении каждого кадра клипа.
- ▶ **MouseMove.** Событие `mouseMove` происходит при любом перемещении указателя мыши.
- ▶ **MouseDown.** Событие `mouseDown` происходит при нажатии на кнопку мыши, когда указатель последней находится на области действия фильма.
- ▶ **MouseUp.** Событие `mouseUp` происходит при отпускании кнопки мыши, когда ее указатель находится на области действия фильма.
- ▶ **KeyDown.** Событие `keyDown` происходит при нажатии на клавишу. Более подробную информацию об использовании клавишного ввода вы найдете в приложении Б.
- ▶ **KeyUp.** Событие `keyUp` происходит при отпускании клавиши. Более подробную информацию об клавишном вводе вы найдете в приложении Б.
- ▶ **Data.** Событие `data` происходит тогда, когда клип принимает данные в результате выполнения действия `loadVariables()` или `loadMovie()`.

Чтобы установить или изменить обработчик события клипа, выполните указанную последовательность действий.

1. Выберите клип, с которым необходимо связать действие либо для которого необходимо изменить обработчик события.
2. Откройте панель **Actions** и выполните одно из следующих действий.
 - ▷ Если вы хотите установить обработчик событий для клипа, выберите действие `onClipEvent` из категории **Actions** ▶ **Movie Clip Control**.
 - ▷ Если вы хотите изменить аргумент обработчика события клипа для уже существующего действия, выберите строку-оператор, содержащую название события, и на области отображения параметров установите соответствующий флажок.

MX При работе в среде Flash MX нет необходимости связывать с каждым экземпляром кнопки или клипа множество сценариев, каждый из которых содержит одно действие. Все действия могут быть связаны с объектом в одном

сценарии. Более подробно о методах обработчиков событий для кнопок и клипов мы расскажем в главе 16.

Связывание действий с кадрами

Действия кадров используются для управления порядком воспроизведения Flash-фильма. Когда на временной шкале встречается ключевой кадр со связанным действием, это действие выполняется, а фильм соответственно реагирует. Более подробно о действиях кадров мы рассказывали в разделе «Работа с действиями кадра» ранее в этой главе.

ПРИМЕЧАНИЕ

Действия кадров могут использоваться как в пределах клипа, так и на главной временной шкале. Их часто используют для создания внутренних циклов бесконечного воспроизведения клипов.

Чтобы связать действие с кадром, выполните следующее.

1. Выберите ключевой кадр, с которым вы хотите связать действие. Если на временной шкале будет выбран не ключевой кадр, Flash свяжет действие с ближайшим предшествующим ключевым кадром (слева) на временной шкале. Для вставки ключевого кадра выберите команду Keyframe меню Insert.
2. После того как ключевой кадр выбран, откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. Выбрать действие можно одним из перечисленных ниже способов.
 - > Выполните двойной щелчок на значке действия в области элементов ActionScript.
 - ⊞ С помощью мыши перетащите значок действия из области элементов ActionScript на область отображения сценария.
 - > Щелкните на кнопке Add (+) и выберите действие в меню.
 - > Введите с клавиатуры набор клавиш, соответствующий требуемому действию.
4. Произведите необходимые настройки посредством ввода значений в области отображения параметров. Помните, что параметры требуются не для всех действий. Свернуть и развернуть область отображения параметров можно посредством щелчка на небольшом треугольнике в правой верхней части панели.
5. Если необходимо добавить в сценарий дополнительные действия, повторите шаги 3 и 4.
6. Действия кадров не требуют специальных обработчиков, поскольку они выполняются при воспроизведении кадров во Flash-фильме. Для тестирования фильма выберите команду Test Movie меню Control.

Использование действий для создания основных элементов управления воспроизведением фильма

Категория Actions области элементов ActionScript содержит несколько подкатегорий действий, предназначенных для управления воспроизведением фильма. Мы уже рассмотрели процесс связывания действий с различными элементами фильма. Теперь давайте рассмотрим возможности различных действий.

MX В новой версии Flash MX в организации панели Actions произошли значительные изменения. Категория Basic Actions (Базовые действия), знакомая пользователям предыдущих версий Flash, была преобразована в более сложную структуру. Возможно, пользователям Flash 5 потребуется некоторое время, чтобы привыкнуть к новшествам. Однако, внимательно изучив новую структуру панели Actions, вы обнаружите, что все требуемые действия по-прежнему доступны и их легко найти.

В данном разделе мы дадим обзор самых важных и полезных действий языка ActionScript. Эти базовые действия перечислены в том порядке, в котором они расположены в своих подкатегориях.

Подкатегория Movie Control

- ▶ gotoAndPlay(). Переход к указанному в качестве параметра кадру или сцене и воспроизведение фильма с этого места.
- ▶ gotoAndStop(). Переход к указанному в качестве параметра кадру или сцене и остановка фильма.
- ▶ on(). Определяет обработчик события мыши, который будет инициировать какое-то действие.
- ▶ playO. Воспроизведение фильма.
- ▶ stopO. Остановка фильма.
- ▶ stopAllSoundsO. Отключение звукового сопровождения фильма.

Подкатегория Browser/Network

- ▶ fscommand(). Управление приложением Flash Player.
- ▶ getURL(). Вызов фильма с указанным URL.
- ▶ loadMovie(). Загрузка других Flash-фильмов.
- ▶ unloadMovie(). Выгрузка других Flash-фильмов.

Подкатегория Movie Clip Control

- ▶ setProperty(). Установка атрибутов клипа: размеры, положение и т. д.
- ▶ onClipEvent(). Определяет обработчик события клипа, который будет инициировать действие.

Устаревшие действия

В связи с появлением новых, более эффективных и функциональных действий, некоторые действия, которые использовались в предыдущих версиях, можно

считать устаревшими. Они по-прежнему функционируют, но их задачи можно выполнять с помощью более эффективных новых действий.

- ▶ `tellTarget()`. Данное действие служит для управления клипом или фильмами. Его можно заменить новым действием `with` или указать путь к объекту с помощью диалогового окна `Insert Target Path`.
- ▶ `toggleHighQuality()`. Данное действие служит для управления качеством воспроизведения фильма. В новой версии Flash MX используйте свойство `_quality`.
- ▶ `ifFrameLoaded()`. Данное действие использовалось для отслеживания загрузки определенных кадров фильма и воспроизведения простой анимации во время загрузки фильма на локальный компьютер. В новой версии Flash MX используйте свойство `_framesloaded`. Более подробно об использовании свойства `_framesloaded` мы расскажем в приложении А.

Переход к кадру или сцене фильма

Действие `goto()` используется для немедленного перехода к указанному кадру или сцене. После перехода к новой сцене вы можете либо остановить, либо продолжить воспроизведение фильма.

Чтобы использовать действие `goto()`, выполните следующее.

1. Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель `Actions`, выбрав команду `Actions` меню `Window` или нажав клавишу `F9`.
3. В области элементов `ActionScript` откройте подкатегорию `Actions` ▶ `Movie Control` и выполните двойной щелчок по значке действия `goto()` либо перетащите его на область отображения сценария.
4. Теперь действие `goto()` появится в области отображения сценария. Установите для него следующие параметры.
 - > Если вы хотите, чтобы после перехода к указанному объекту воспроизведение фильма продолжалось, оставьте установленным переключатель `Go to and Play`. Если после перехода воспроизведение фильма должно быть остановлено, активизируйте переключатель `Go to and Stop`. Сценарий фильма будет автоматически обновлен.
 - > В раскрывающемся списке `Scene` (Сцена) выберите сцену, к которой необходимо перейти. При выборе элемента `next scene` или `previous scene` переход будет осуществлен к первому кадру следующей или предыдущей сцены. При выборе элемента `current scene` или элемента с именем сцены необходимо будет указать номер кадра указанной сцены, к которому необходимо перейти.
 - > В раскрывающемся списке `Type` (Тип) установите способ идентификации кадра, к которому будет осуществлен переход. Идентифицировать кадр можно по номеру (`Frame Number`), метке кадра (`Frame Label`), с помощью выражения (`Expression`), либо выбрав элемент `Next Frame` (Следующий кадр) или `Previous Frame` (Предыдущий кадр).

- ▷ Если в раскрываемом списке Type выбран элемент Frame Number (Номер кадра) или Frame Label (Метка кадра), то необходимо ввести номер кадра или имя метки в поле Frame (Кадр). Если в списке Type выбран элемент Expression (Выражение), то необходимо ввести в поле Frame выражение, в котором описана метка кадра или его номер. (Более подробно о написании выражений мы расскажем в главе 14.)

Установка обработчика события мыши

В некоторых ситуациях вам может потребоваться сначала указать обработчик события, а затем выбрать действие, которое обработчик будет инициировать. Чтобы вручную установить обработчик события, используйте действие `on()`. При выборе данного действия в области отображения сценария появится принятый по умолчанию обработчик события `on(release)`. Вы можете изменить обработчик, а затем выбрать действие, которое должно быть инициировано.

Для связывания действия `on()` с какой-либо кнопкой выполните следующее.

1. Выберите экземпляр кнопки, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Movie Control и выполните двойной щелчок на значке действия `on()` либо перетащите его на область отображения сценария.
4. В этой области появится принятый по умолчанию обработчик события `release`.


```
on(release){
}
```

Для изменения обработчика выберите строку `on(release)` и установите соответствующий флажок на области отображения параметров. Более подробно об обработчиках события мыши мы расскажем в главе 13.

5. После того как обработчик события выбран, добавьте в сценарий действие, которое он должен инициировать.
6. В области отображения параметров введите все необходимые значения, если они необходимы для добавленного действия.
7. Один обработчик может инициировать выполнение нескольких действий. Чтобы добавить дополнительные действия, повторите шаги 5 и 6. Используя кнопки Up (Вверх) и Down (Вниз), установите порядок выполнения действий.

Воспроизведение и остановка фильма

Существуют всего два действия для управления воспроизведением и остановкой фильма. Воспроизведение Flash-фильма *продолжается* до тех пор, пока все кадры на временной шкале не будут воспроизведены либо пока фильм не будет остановлен с помощью действия `stop()`. Независимо от причины остановки фильма для продолжения воспроизведения необходимо использовать действие `play()`. Действия `play()` и `stop()` могут использоваться для управления воспроизведением кадров как главной временной шкалы фильма, так и клипа. Более подробно об управлении клипами с помощью сценариев ActionScript мы расскажем в главе 16.

Для связывания действия `playO` или `stop()` с каким-либо объектом выполните следующее.

1. Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие. За исключением случаев, когда указан путь к объекту, действие будет управлять только клипом или временной шкалой, с которой оно связано.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Movie Control и выполните двойной щелчок на значке действия `play()` или `stopO`.
4. Если в этом есть необходимость, выберите новые параметры для обработчика события посредством ввода значений в области отображения параметров. Если действия связаны с кадром, то остановка или воспроизведение фильма будет выполняться, когда головка воспроизведения перейдет к этому кадру. Действие, предназначенное для возвращения головки воспроизведения к первому кадру фильма, будет выглядеть так:

```
on(release){
    gotoAndStop(1);
}
```

Отключение звукового сопровождения

Действие `stopAllSoundsO` позволяет отключить все звуковые элементы фильма. При выполнении данного действия воспроизведение всех звуков будет остановлено. Звуковое сопровождение остается отключенным до тех пор, пока оно снова не будет включено другим действием. После выполнения действия `stopAllSoundsO` звук начнет воспроизводиться, если он снова встретится на временной шкале. (Более подробно о звуке мы расскажем в главе 21.)

Чтобы связать действие `stopAllSoundsO` с каким-либо объектом, выполните следующее.

1. Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Movie Control и выполните двойной щелчок на значке действия `stopAllSounds()`. Действие `stopAllSoundsO` не принимает аргументов и будет выполнено, когда произойдет иницилирующее его событие либо при воспроизведении связанного с ним кадра.

Связь с URL

С помощью действия `getURL()` можно установить связь с URL непосредственно из Flash-фильма. Связь с URL позволяет выполнять две вещи. Во-первых, вы можете использовать Flash для создания Web-ссылки и загрузки информации в окно браузера из источника, соответствующего указанному URL. Во-вторых, действие можно использовать для передачи данных ресурсу, который находится

по определенному URL. Таким ресурсом может быть фильм типа Macromedia Director, сценарий CGI или сценарий другого типа, размещенный на сервере.

Чтобы связать действие `getURL()`, необходимо выполнить следующее.

1. Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Browser/Network и выполните двойной щелчок на значке действия `getURL()`. Действие `getURL()` имеет три параметра, которые должны быть определены: URL, Window (Окно) и Variables (Переменные).

Существует также возможность использовать выражение, в котором описывается местоположение URL. Для этого необходимо установить флажок Expression (Выражение) справа от поля URL. (Более подробно о выражениях мы расскажем в главе 14.)

4. В поле URL необходимо ввести абсолютный или относительный путь к ресурсу. Абсолютный путь включает полный адрес URL, например `http://www.myflashlink.com/myfile.html`. Относительный путь содержит информацию о нахождении файла в пределах файловой структуры, где сохранен Flash-фильм, например такую: `currentfolder/folder1/folder0`.
5. Раскрывающийся список Window позволяет определить, в каком окне браузер будет загружать документ, на который указывает URL. Вы можете ввести определенное имя фрейма или окна, ввести выражение, которое описывает местоположение окна, либо выбрать один из следующих элементов раскрывающегося списка.
 - ▷ `_self`. Документ загружается в текущем окне, где в настоящий момент воспроизводится фильм.
 - ▷ `_blank`. Документ загружается в новом окне.
 - ▷ `_parent`. Данный элемент указывает на текущее окно и позволяет заменить только фрейм, где в настоящий момент воспроизводится фильм.
 - ▷ `_top`. Данный элемент целесообразно использовать тогда, когда ваш фильм воспроизводится во фрейме, а новый документ, на который указывает URL, должен заполнить все окно браузера.

Например, чтобы при щелчке на кнопке документ `myinfo.html` загружался в новом окне браузера, необходимо написать следующий код с указанием относительного URL этого документа:

```
on(release){
    getURL("myinfo.html", "_blank");
}
```

6. Раскрывающийся список Variables (Переменные) используется для передачи переменных на URL, с которым вы устанавливаете связь.
 - ▷ Don't Send (не передавать). Данный элемент установлен по умолчанию и используется в тех случаях, когда нет необходимости в обмене данными.
 - ▷ Send Using GET (передавать с использованием метода GET). Данный элемент списка выбирают в тех случаях, когда необходимо передать несколько переменных и объем передаваемых данных небольшой.
 - ▷ Send Using Post (передавать с использованием метода POST). Данный элемент списка выбирают в тех случаях, когда необходимо передавать большой объем информации.

Управление приложением Flash Player

Действие `fscommand()` используется для управления приложением Flash Player, когда последнее работает в режиме воспроизведения фильма формата Flash projector. Действие `fscommand()` также можно использовать для обмена данными с приложением, которое выполняет роль хоста вашего фильма, например с Web-браузером или операционной системой компьютера.

Для установки параметров воспроизведения в приложении Flash Player выполните следующее.

1. Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Browser/Network и выполните двойной щелчок на значке действия `fscommand0`.
4. В области отображения параметров выберите один из следующих элементов раскрывающегося списка Commands for standalone player (Команды для автономного проигрывателя), служащих аргументами действия `fscommand0`.
 - ▷ `fullscreen`. Со значением `true` данный аргумент позволяет фильму типа Flash projector заполнить всю площадь экрана без отображения строки меню. Если установить значение `false`, то фильм будет воспроизводиться в окне, размеры которого будут соответствовать параметрам Dimensions, установленным в диалоговом окне Document Properties.
 - ▷ `allowscale`. Со значением `true` данный аргумент позволяет изменять масштаб отображения кадров анимации. Если установить значение `false`, то изменять масштаб отображения кадров анимации будет невозможно.
 - ▷ `showmenu`. Аргумент `showmenu` со значением `true` позволяет получать доступ к командам контекстного меню посредством щелчка при нажатой клавише Ctrl (Macintosh) или щелчка правой кнопкой мыши (Windows). При установке значения `false` данное меню будет недоступно.
 - ▷ `quit`. Данный аргумент позволяет завершить работу приложения.
 - ▷ `trapallkeys`. Со значением `true` данный аргумент блокирует любой ввод с клавиатуры. Если ваш фильм должен управляться с помощью мыши, то ввод с клавиатуры желательно заблокировать. Если вы используете этот

аргумент совместно с аргументом `showmenu` со значением `false`, убедитесь в наличии элемента, который завершает работу (`quit`).

- > `exec`. Данный аргумент предназначен для запуска другого приложения. В качестве его значения необходимо указать путь к приложению, которое необходимо запустить.

Более подробно о командах действия `fscommand()`, автономных фильмах формата Flash projector и о том, как их использовать для записи Flash-фильмов на компакт-диски, мы расскажем в главе 29.

Создание сценариев для нескольких фильмов

Во Flash-проекте могут использоваться несколько фильмов. Для управления последовательностью воспроизведения фильмов используются действия `loadMovie()` (Загрузка фильма) и `unloadMovie()` (Выгрузка фильма). Такая техника позволяет устанавливать последовательность воспроизведения серии фильмов, а также размещать кадры одного фильма поверх кадров другого для создания эффекта быстрого переключения между ними. Кроме того, за счет использования действий `loadMovie()` и `unloadMovie()` можно воспроизводить различные фильмы в одном окне браузера, что создает эффект плавного перехода от показа одного фильма к показу другого.

Загрузка фильма

Чтобы загрузить дополнительные фильмы в окне приложения Flash Player, выполните следующее.

1. Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▸ Browser/Network и выполните двойной щелчок на значке действия `loadMovie()` (рис. 12.4).
4. На области отображения параметров в поле URL введите имя SWF-файла, который вы хотите загрузить. Рекомендуется хранить все файлы фильмов в одном каталоге (основной файл и дополнительные, которые должны быть загружены).
5. В раскрывающемся списке Location (Расположение) выберите один из элементов, Level (Уровень) или Target (Объект).
6. При выборе элемента Level необходимо ввести номер уровня.
 - > Значение 0 установлено по умолчанию и соответствует уровню фильма, который был загружен первым. Фильм уровня 0 устанавливает частоту смены кадров, фоновый цвет и размер для всех остальных загружаемых фильмов. Если новый фильм загружается на уровень 0, он заменит начальный фильм и выгрузит все остальные уровни.
 - > Чтобы загрузить фильм на новом уровне, выберите следующий доступный положительный номер.

- > Чтобы заменить загруженный фильм, необходимо определить его уровень и загрузить новый фильм на этот же уровень.
- Если в поле списка Location выбран элемент Target, необходимо указать имя клипа, который будет заменен загружаемым фильмом. Левый верхний угол области воспроизведения нового фильма будет позиционирован в точке регистрации клипа на области действия. Новый фильм также унаследует атрибуты заменяемого клипа (масштаб, вращение и т. д.).
 - По умолчанию в поле списка Variables (Переменные) выбран элемент Don't Send (Не передавать). Однако, если вы хотите разрешить передачу переменных, выберите один из элементов, Send Using GET или Send Using POST. Об особенностях данных методов мы говорили ранее, когда обсуждали действие `getURL()`.

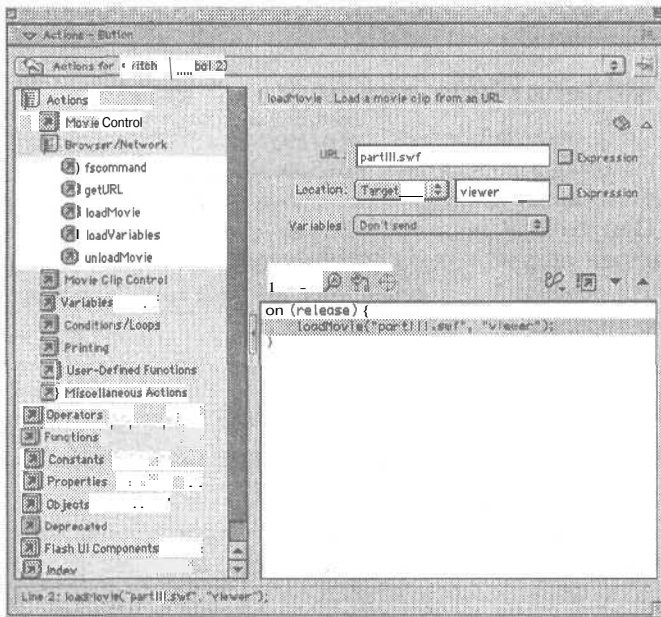


Рис. 12.4. Действие `loadMovie()` на панели Actions

Выгрузка фильма

Чтобы выгрузить фильм из окна приложения Flash Player, выполните следующее.

- Выберите кнопку, кадр или экземпляр клипа, с которым вы хотите связать действие.
- Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
- В области элементов ActionScript откройте подкатегорию Actions ► Browser/Network и выполните двойной щелчок на значке действия `unloadMovie()` (рис. 12.4).
- В области отображения параметров в раскрывающемся списке Location (Расположение) выберите один из элементов, Level (Уровень) или Target (Объект).

5. При выборе элемента Level необходимо ввести номер уровня фильма, который вы хотите выгрузить.
6. Если в раскрывающемся списке Location выбран элемент Target, то необходимо указать имя либо путь к экземпляру клипа, который вы хотите выгрузить.

ПРИМЕЧАНИЕ

Возможно, вы заметили, что в сценариях действия `loadMovie()` и `unloadMovie()` иногда пишутся как `loadMovieNum()` и `unloadMovieNum()`. В режиме Normal Flash автоматически делает эти изменения. Каждое действие с измененным именем производит практически такую же работу, однако использует другой синтаксис. Действие `loadMovie()` передает аргументы как строку, тогда как действие `loadMovieNum()` использует число, чтобы указать уровень, на который будет загружен новый фильм. В режиме Normal действие `loadMovie()` используется по умолчанию для загрузки фильмов взамен экземпляров клипов, а действие `loadMovieNum()` используется для загрузки фильмов на определенный уровень. (Более подробно о работе с несколькими временными шкалами мы расскажем в главе 16.)

Установка обработчика для событий клипа

В некоторых ситуациях вам может потребоваться сначала указать обработчик события, а затем выбрать действие, которое обработчик будет инициировать. Чтобы вручную установить обработчик события клипа, используйте действие `onClipEvent()`. При выборе данного действия в области отображения сценария появится принятый по умолчанию обработчик события `onClipEvent (load)`. Вы можете изменить обработчик, а затем выбрать действие, которое должно быть инициировано.

Для связывания действия `onClipEvent()` выполните следующее.

1. Выберите экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Movie Clip Control и выполните двойной щелчок на значке действия `onClipEvent()`.
4. Принятый по умолчанию обработчик события `load` можно изменить в области отображения параметров. Изначально сценарий выглядит так:

```
onClipEvent(load){
}
```

5. После того как обработчик события выбран, добавьте в сценарий действие, которое он должен инициировать.
6. В области отображения параметров ведите все необходимые значения, если они требуются для добавленного действия.
7. Один обработчик может инициировать выполнение нескольких действий. Чтобы добавить дополнительные действия, повторите шаги 5 и 6. Используя кнопки Up (Вверх) и Down (Вниз), установите порядок выполнения действий.

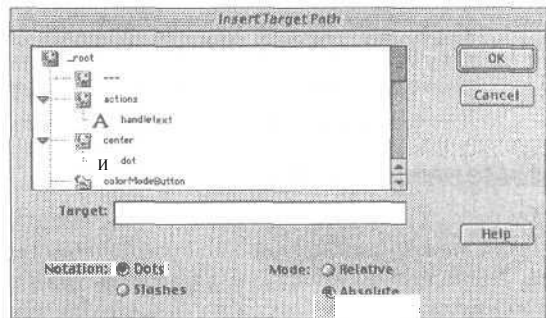
Управление клипами и другими фильмами

Управление клипами во Flash-фильме осуществляется с помощью действия `tellTarget()`. Хотя технически данное действие можно считать устаревшим, его по-прежнему можно эффективно использовать для управления клипами в режиме Normal. Данное действие может быть связано как с объектами, так и с кадрами. Например, для остановки клипа можно использовать кнопку, а действие ключевого кадра может инициировать начало воспроизведения клипа.

Некоторые установки необходимо сделать до использования действия `tellTarget()`. Во-первых, ваш фильм должен содержать клип. Кроме того, экземпляр клипа должен иметь уникальное имя. Чтобы присвоить его, выберите экземпляр клипа, откройте панель Properties, выбрав команду Properties меню Window, и введите имя в поле Instance Name (Имя экземпляра). Имя экземпляра символа может состоять из букв, а также из комбинации букв и цифр. В больших проектах рекомендуется присваивать экземплярам символов значащие имена, чтобы затем легче было вспомнить назначение каждого экземпляра символа. Кроме того, клип должен находиться в том же кадре, что и элемент (кадр или объект), с помощью которого вы будете управлять клипом. Кнопка в кадре с номером 1 не может управлять клипом, находящимся в кадре с номером 2.

Для связывания действия `tellTarget()` выполните следующее.

1. Выберите экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Deprecated ▸ Actions и выполните двойной щелчок на значке действия `tellTarget()`.
4. Чтобы указать нужный экземпляр клипа, щелкните на кнопке Insert a target path (Указать путь к объекту), которая находится в нижней части области отображения параметров. Если кнопка недоступна, выполните щелчок на поле Target (Объект), после чего кнопка станет доступной.
5. В появившемся диалоговом окне Insert Target Path (Указать путь к объекту) будет показана структура расположения клипов в вашем фильме.
6. Переключатель Notation (Запись) установите в положение Dots (Точки). При этом путь к объекту будет записан с использованием современного точечного синтаксиса (в качестве разделителей будут использоваться точки). Если вы привыкли к старому синтаксису, где в качестве разделителей используются косые линии, переключатель Notation установите в положение Slashes (Косые линии).



7. Переключатель Mode (Режим) позволяет установить, какие клипы будут видны в окне Insert Target Path.
 - При установке переключателя Mode в положение Relative (Относительный) в диалоговом окне Insert Target Path будут отображаться только клипы текущей сцены. В этом режиме наверху структуры вы будете видеть элемент `this`. Слово `this` используется в качестве ссылки на текущую временную шкалу или клип.
 - При установке переключателя Mode в положение Absolute (Абсолютный) будут отображаться все клипы фильма. В этом режиме наверху структуры вы будете видеть элемент `_root`. Свойство клипа `_root` содержит ссылку на главную временную шкалу фильма.

Вы можете указывать путь в любом из двух режимов, однако относительный путь более краток и в некоторых случаях легче для понимания.

8. При выборе клипа в области Target появится имя экземпляра клипа. Щелкните на кнопке ОК и возвратитесь к панели Actions.
9. После того как экземпляр клипа выбран, необходимо добавить действие, которое будет им управлять. Убедитесь, что в области отображения сценария выбрана строка `tellTarget()`. Перейдите к области элементов ActionScript и выберите одно или несколько действий, которые должны управлять клипом посредством действия `tellTarget()`. Конечный сценарий действия, которое останавливает воспроизведение экземпляра клипа с именем `icon`, будет выглядеть так:

```
on(release){
tellTarget("icon"){
    stop();
}
}
```

ПРИМЕЧАНИЕ

Как мы уже говорили, действие `tellTarget` считается устаревшим. Однако его по-прежнему можно использовать при работе в режиме Normal. Более подробно об альтернативных действиях, способных выполнять ту же задачу, мы расскажем в главе 16.

Изменение свойств клипа

Используя действие `setProperty()`, вы можете создавать сценарии, с помощью которых можно манипулировать характеристиками и свойствами экземпляра клипа. Безусловно, свойства и характеристики экземпляра клипа можно установить при его создании. Однако вы можете предоставить возможность зрителям фильма динамически изменять свойства клипа, используя интерактивные элементы управления. Необходимо помнить следующее: чтобы можно было управлять клипом с помощью сценария ActionScript, экземпляр клипа должен иметь уникальное имя.

Для связывания действия `setProperty()` выполните следующее.

1. Выберите экземпляр клипа, с которым вы хотите связать действие.
2. Откройте панель Actions, выбрав команду Actions меню Window или нажав клавишу F9.
3. В области элементов ActionScript откройте подкатегорию Actions ▶ Movie Clip Control и выполните двойной щелчок на значке действия `setPropertyO`. Для данного действия необходимо указать три параметра.
 - В раскрывающемся списке Property (Свойство) выберите свойство фильма, которое вы хотите изменить.
 - В поле Target (Объект) введите имя экземпляра клипа, свойство которого вы хотите изменить. В качестве альтернативы, щелкните на кнопке Insert a target path и в открывшемся диалоговом окне Insert Target Path укажите путь к экземпляру клипа. О том, как работать с диалоговым окном Insert Target Path, мы рассказывали при обсуждении действия `tellTargetO`.
 - В поле Value (Значение) введите значение изменяемого параметра в пределах допустимых величин для данного свойства. Более подробно о диапазоне значений для свойств клипа мы расскажем в приложении Б.

ПРИМЕЧАНИЕ

Альтернатива использованию действия `setPropertyO` будет рассмотрена в главе 16.

Разработки

Поскольку сценарии ActionScript широко используются практически во всех Web-проектах, нам было трудно выбрать какой-либо один в качестве примера. Действия — это основа интерактивного Flash-дизайна и основная составляющая каждого интерактивного фильма, анимации или игры. Благодаря действиям разработчик может точно определять порядок воспроизведения фильма и предоставлять зрителям возможность интерактивно управлять свойствами и последовательностью воспроизведения отдельных сцен или фрагментов фильма.

Вашему вниманию мы хотели бы предложить разработку Web-узла Michael Schuster Associates, анимация для которого была создана компанией Core Five Creative (www.corefive.com) с использованием действий и клипов Flash. Хотя конечный продукт внешне ничем не отличается от других проектов, такая техника позволяет разбить процесс разработки на несколько этапов, а отдельные клипы совместить в конечном продукте. С данным проектом вы можете познакомиться по адресу www.msaarch.com.

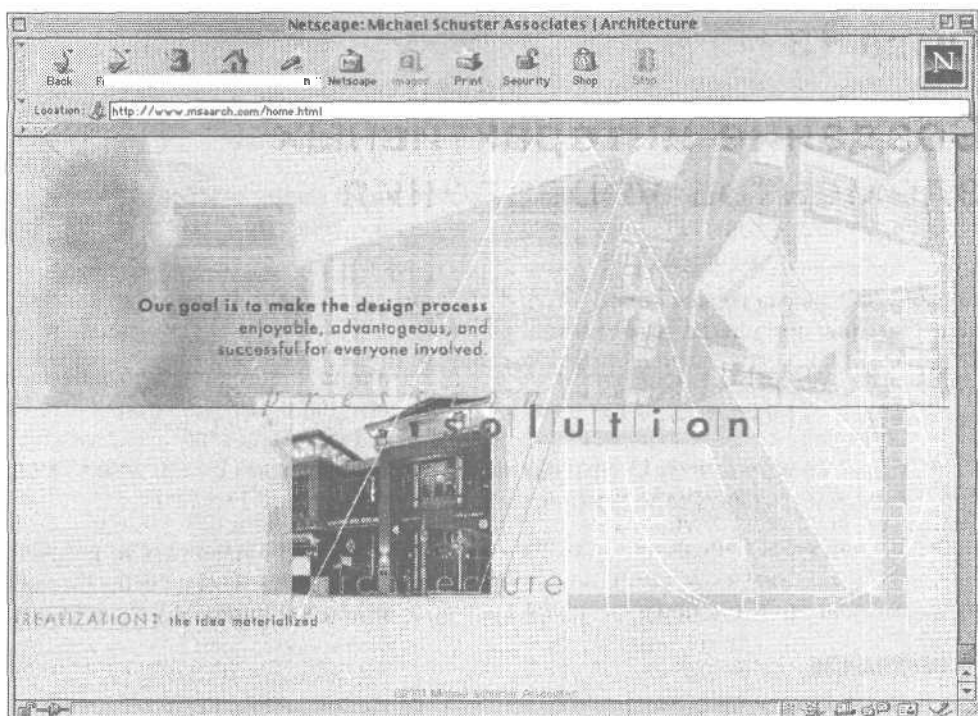


Рис. 12.5. Web-узел компании Michael Schuster Associates

Резюме

В данной главе вы познакомились с основами языка ActionScript. Вы узнали, насколько важным является процесс создания сценариев ActionScript. Сценарии позволяют не только управлять воспроизведением фильма, но и обмениваться информацией с другими фильмами или приложениями, открывать окна браузера и определять способ взаимодействия с аудиторией. По мере приобретения опыта работы с ActionScript вы будете создавать все более сложные сценарии для интерактивных фильмов и Web-узлов.

Глава 13

Создание интерактивных элементов управления

В предыдущей главе вы познакомились с действиями и изучили некоторые базовые компоненты языка ActionScript. Теперь настало время на практике реализовать ваши знания. Вы знаете, что действия позволяют управлять воспроизведением фильма. Однако, чтобы управлять, необходимо иметь элементы управления. Такими элементами могут быть кнопки или переключатели, созданные специально для инициирования действий, которые и будут управлять фильмом.

Создание элементов управления — это один из важных этапов работы над проектом, поскольку от их качества зависит внешний вид панели, на которой они размещаются, и простота управления фильмом. Наиболее простыми элементами управления являются кнопки. С их помощью можно воспроизводить и останавливать фильм, а также переключаться между различными сценами. Панель управления фильмом (иногда ее называют навигационной панелью) может быть такой же простой, как и применяемая в видеомагнитофонах. Кроме того, элементы управления могут быть частью общего дизайна фильма. Приобретя некоторый опыт, вы сможете создавать надежные интерфейсы с гибкой схемой навигации и изящным внешним видом.

В данной главе будут рассмотрены следующие темы:

- ▶ библиотека кнопок;
- ▶ создание пользовательских кнопок;
- ▶ предварительный просмотр кнопок;
- ▶ тестирование кнопок;
- ▶ редактирование кнопок;
- ▶ создание кнопок, содержащих несколько слоев;
- ▶ создание анимированных кнопок;
- ▶ создание многокнопочных интерфейсов;
- ▶ создание «некнопочных» кнопок.

Управление интерактивными фильмами с помощью простых кнопок

Flash позволяет создавать сложные схемы интерактивного взаимодействия, однако это не всегда необходимо. Иногда для управления фильмом достаточно несколько простых элементов. Для тех разработчиков и дизайнеров, которые только осваивают Flash и желают сделать управление как можно более простым, достаточно будет создать несколько кнопок для выполнения самых элементарных

действий. Процесс создания простых элементов управления не требует каких-либо особых навыков, а результат может быть вполне удовлетворительным.

Библиотека кнопок

Flash предлагает пользователю несколько библиотек стандартных элементов. Эти библиотеки включают такие стандартные элементы, как звуки, клипы и, конечно же, кнопки. Данные ресурсы связаны с программой Flash и всегда находятся в распоряжении разработчика. С символами из библиотеки кнопок легко экспериментировать и проводить тесты. Хотя некоторые из них могут не соответствовать вашим личным стандартам или целям дизайнера, все они достаточно функциональны и являются ценными компонентами среды Flash.

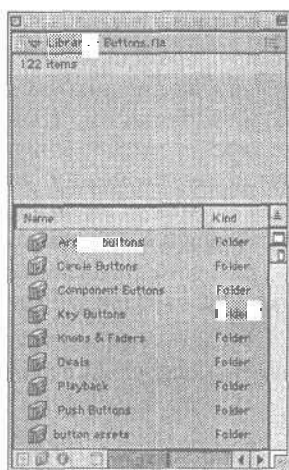


Рис. 13.1. Библиотека кнопок

Чтобы открыть окно библиотеки кнопок, выберите команду **Window** ▶ **Common Libraries** ▶ **Buttons** (рис. 13.1). Эта библиотека выглядит так же, как и все остальные библиотеки Flash. Единственное отличие — ее наполнение. Библиотека содержит широкий ассортимент кнопок, готовых для использования в фильме.

ПРИМЕЧАНИЕ

Более подробно о работе с библиотеками и о совместно используемых компонентах Flash мы рассказывали в главе 7.

Кнопки хранятся в каталогах. Каждый каталог содержит свою группу кнопок. В такую группу входят кнопки, выполненные в одном стиле. При создании фильмов лучше использовать кнопки одной группы, что позволит сохранить общий стиль элементов управления.

Чтобы воспользоваться библиотекой кнопок, выполните следующие действия:

1. Откройте окно библиотеки кнопок, выбрав команду **Window** ▶ **Common Libraries** ▶ **Buttons**.

2. Откройте каталог группы кнопок, которые вы хотите использовать. Развернуть и свернуть содержимое каталога можно посредством двойного щелчка на значке каталога (рис. 13.2).

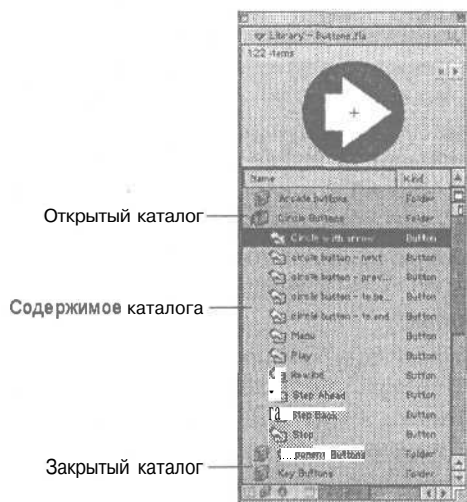


Рис. 13.2. Структура каталогов библиотеки кнопок

3. Выполните одно из следующих действий.
- > Откройте библиотеку вашего Flash-фильма, выбрав команду Library меню Window либо нажав клавишу F11. С помощью мыши перетащите кнопку из окна библиотеки кнопок в окно библиотеки вашего фильма.
 - > С помощью мыши перетащите кнопку из окна библиотеки кнопок непосредственно на область действия. Значок кнопки автоматически появится в окне библиотеки элементов, используемых в вашем фильме.

Данная кнопка теперь является символом в вашем фильме. При перетаскивании кнопки непосредственно на область действия создается экземпляр символа кнопки, а родительский символ остается в библиотеке фильма.

С созданным экземпляром символа кнопки можно связать действие. О том, как связывать различные действия с экземплярами символов, речь шла в главе 12. Позже, в разделе «Редактирование кнопок», мы расскажем, как можно производить редактирование стандартных кнопок и настраивать их для собственных фильмов.

Создание пользовательских кнопок

Хотя использовать стандартные кнопки очень удобно, иногда возникает потребность в разработке собственных кнопок. В конце концов, вы изучаете Flash именно для того, чтобы получать удовольствие от процесса создания. Однако, кроме удовольствия, применение собственных кнопок дает еще и некоторые преимущества. Самое большое из них состоит в том, что, создав самостоятельно кнопку, вы получаете именно тот результат, который хотите. Разрабатывая собственные кнопки, вы сможете учесть все особенности дизайнера и технические

характеристики элементов управления фильмом. Элементы интерактивного управления — важная составляющая Flash-фильма, поэтому они должны точно соответствовать выбранному дизайну, а также обладать функциональными возможностями, соответствующими задачам проекта.

Кнопка во Flash-фильме — это короткий интерактивный фильм, содержащий всего четыре кадра. Каждый кадр представляет различное «состояние», или внешний вид кнопки, который соответствует определенным действиям пользователя. Эти различные состояния сохранены в виде отдельных изображений в ключевых кадрах на временной шкале кнопки. Если вы переместите указатель мыши на изображение кнопки и щелкните на ней, ее внешний вид изменится, так как будет выведено изображение соответствующего ключевого кадра.

Посредством смены изображений кнопки будет создан эффект интерактивности, так как в соответствии со сменой внешнего вида кнопки будут происходить изменения и в самом фильме.





Как и клипы, кнопки хранятся в библиотеке Flash-фильма в виде символов. Следовательно, вы можете создать кнопку и многократно использовать ее в фильме. Каждый экземпляр кнопки будет действовать независимо и активизировать только связанные с ним действия.

Кнопки имеют следующие четыре состояния.

- ▶ **Up.** Первое состояние соответствует внешнему виду кнопки, когда указатель мыши находится за ее пределами. Данное состояние можно рассматривать как «пассивное».
- ▶ **Over.** Данное состояние соответствует внешнему виду кнопки, когда указатель мыши находится над кнопкой. Соответствующее изображение кнопки хранится во втором ключевом кадре.
- ▶ **Down.** Изображение в третьем ключевом кадре соответствует нажатому состоянию кнопки.
- ▶ **Hit.** Внешний контур изображения, находящегося в последнем ключевом кадре, ограничивает область кнопки, которая будет реагировать на щелчок мыши. Состояние Hit можно рассматривать как активную область, на которой необходимо произвести щелчок для выполнения любых связанных с кнопкой сценариев.

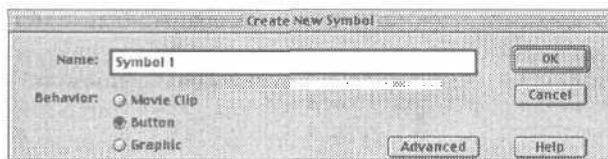
В табл. 13.1 содержатся иллюстрации и описания четырех состояний кнопки.

Таблица 13.1. Четыре состояния обычной кнопки

Состояние	Внешний вид	Описание
Up		Вид кнопки, когда указатель мыши не находится над кнопкой
Over	 flip	Вид кнопки, когда указатель мыши находится над кнопкой
Down		Вид кнопки в момент щелчка мышью
Hit		Задание области кнопки, которая реагирует на щелчок мыши

Для создания собственной кнопки выполните следующие действия.

1. Откройте диалоговое окно Create New Symbol, выбрав команду New Symbol меню Insert, либо воспользуйтесь комбинацией клавиш Cmd/Ctrl+F8.



2. Переключатель типа символа установите в положение Button (Кнопка) и введите имя символа в поле Name.
3. После щелчка на кнопке ОК вы переключитесь в режим редактирования символа (рис. 13.3). Обратите внимание на то, что данный режим характеризуется некоторыми изменениями в главном окне среды Flash. В строке Scene and Symbol (Сцена и символ) рядом с названием сцены появится имя редактируемого символа. Кроме того, вместо главной временной шкалы появится временная шкала кнопки, содержащая четыре кадра, соответствующие четырем состояниям кнопки. Первый кадр, соответствующий состоянию Up, содержит пустой ключевой кадр.

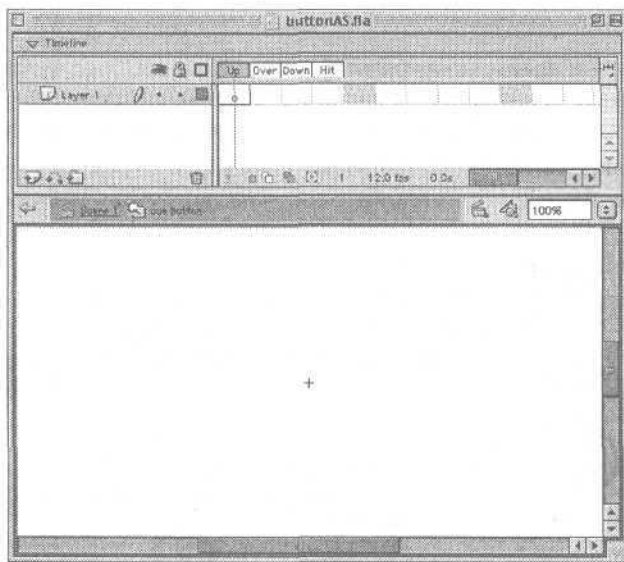
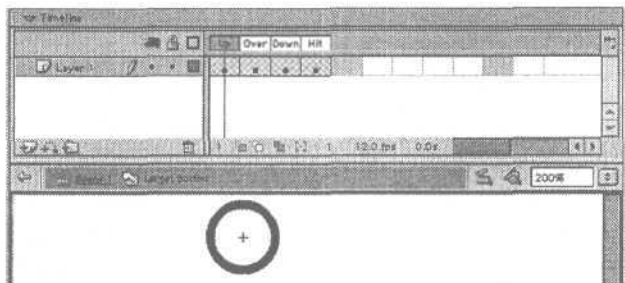


Рис. 13.3. Временная шкала в режиме редактирования символа кнопки

4. Создайте изображение кнопки, соответствующее состоянию Up, выполнив одно из нижеперечисленных действий.
 - Для создания изображения кнопки используйте инструменты рисования.
 - Импортируйте графический файл и переместите изображение кнопки к точке регистрации, в центр области действия.

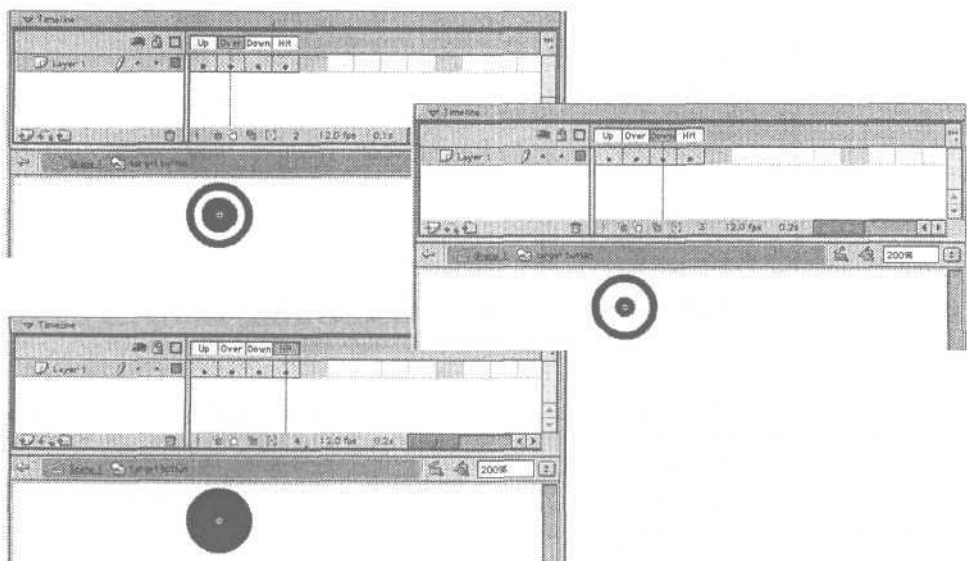
- Перетащите из окна библиотеки на область действия графический символ или клип, создав тем самым экземпляр этого объекта в пределах кнопки. Небольшое перекрестие в центре области действия указывает точку регистрации кнопки.



ПРИМЕЧАНИЕ

Вы можете использовать клипы для отображения состояний кнопок, однако нельзя помещать кнопки на другие кнопки. При использовании клипов в качестве состояний кнопок можно создавать анимированные кнопки. Более подробно об этом мы расскажем в разделе «Создание анимированных кнопок» позже в этой главе.

5. Чтобы создать новый ключевой кадр для состояния Over, выполните щелчок на втором кадре и выберите команду Keyframe меню Insert. Flash автоматически поместит изображение первого ключевого кадра в новый ключевой кадр.
6. Используя описанные ранее методы, произведите изменения изображения в кадре с номером 2, которое будет соответствовать состоянию Over.



7. Повторите шаги 5 и 6 для состояний Down и Hit.

Состояние Hit является особенным. Данное состояние определяет область, которая будет реагировать на любые щелчки мыши. Изображение в кадре состояния Hit должно быть сплошным, чтобы не допустить «провалов», то есть наличия неактивных зон в пределах кнопки. Чтобы пользователи не испытывали никаких неудобств, активная область кнопки должна быть достаточно большой для выполнения на ней щелчка.

СОВЕТ

Если вы создаете кнопку в виде фрагмента текста, рисунок в кадре состояния Hit особенно важен. Поскольку алфавитные символы содержат «полости», щелчок мыши, например в центре буквы «о», не будет воспринят, если в кадре состояния Hit также содержится изображение буквы «о». Чтобы избежать этого, необходимо в кадре состояния Hit создать изображение сплошного прямоугольника, размеры которого приблизительно соответствуют размерам фрагмента текста. Поскольку изображение в состоянии Hit невидимо, прямоугольник не изменит внешний вид кнопки в фильме, однако пользоваться такой кнопкой будет гораздо удобнее.

8. После создания изображений для всех четырех состояний кнопки выйдите из режима редактирования символа, выбрав команду Edit Document, либо воспользуйтесь для этого комбинацией клавиш Cmd/Ctrl+E, либо щелкните на имени сцены в строке Scene and Symbol.
9. С помощью мыши перетащите кнопку из окна библиотеки на область действия, создав тем самым экземпляр кнопки в фильме.
10. Чтобы управлять различными элементами фильма, необходимо связать с кнопкой действие (или действия). Основные действия и способ их связывания с объектами мы обсуждали в главе 12. Более подробно о дополнительных действиях для кнопок мы расскажем в главе 16.

В этом разделе мы рассказали об основных этапах создания простых кнопок для Flash-фильма. Эти знания пригодятся вам при разработке сложных и анимированных кнопок. О том, как связывать с кнопками звуки, мы расскажем в главе 20. В следующем разделе мы обсудим детали процесса тестирования кнопок, а также расскажем, как вносить в них изменения или редактировать отдельные компоненты.

Предварительный просмотр, тестирование и редактирование кнопок

После создания кнопок необходимо убедиться в их работоспособности и функциональности. Для тестирования кнопок Flash предлагает два основных способа: тестирование кнопок в среде Flash и тестирование всего фильма. В зависимости от особенностей проекта вам необходимо будет выбрать один из вариантов. Таким образом, для овладения навыками эффективной и быстрой работы вы должны познакомиться как с одной, так и с другой техникой.

Предварительный просмотр кнопок

Существует возможность предварительного просмотра кнопок непосредственно в среде разработки Flash (в режиме разработки). По умолчанию в процессе работы над фильмом Flash сохраняет пассивное состояние всех кнопок. Однако при необходимости вы можете активизировать кнопки и проверить их непосредственно во время работы над фильмом.

Для тестирования кнопок в режиме разработки активизируйте команду Enable Simple Buttons меню Control. Около имени этой команды появится небольшой флажок, указывающий на включение данного режима. Чтобы отменить команду, повторите процесс еще раз.

ПРИМЕЧАНИЕ

Работать в среде Flash с активизированными кнопками не всегда удобно. Например, чтобы выбрать такую кнопку, необходимо поместить указатель мыши за пределы кнопки, нажать кнопку мыши и, удерживая ее, переместить указатель так, чтобы требуемая кнопка оказалась внутри прямоугольника выделения. Вокруг выделенной кнопки появляется рамка выделения.

Тестирование кнопок

Предварительный просмотр кнопок в режиме разработки дает вам возможность только *приблизительно* оценить, как будут выглядеть кнопки в конечном фильме. Такой режим не *позволяет* просматривать клипы, которые могут являться элементами кнопки. Чтобы полностью оценить работу и внешние характеристики созданных кнопок, необходимо тестировать сцену или фильм. Если вы решили тестировать фильм, выберите команду Test Movie или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+Return/Enter`. Чтобы тестировать сцену, выберите команду Test Scene или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+Option/Alt+Return/Enter`. Более подробно о тестировании фильма мы расскажем в главе 27.

Редактирование кнопок

Если при тестировании кнопки вы установили, что ей нужны некоторые доработки, необходимо *вернуться* в режим редактирования символа кнопки. Если в вашем фильме вы использовали несколько экземпляров одного родительского символа кнопки, то для внесения изменений во все экземпляры достаточно доработать родительский символ, хранящийся в библиотеке. Все экземпляры данного символа будут автоматически обновлены. Более подробно о символах и их экземплярах речь шла в главе 7.

Существуют три способа редактирования кнопок во Flash-фильме: правка кнопки на области действия («местная правка»), в режиме редактирования символа и редактирование в отдельном окне. Результаты всех способов правки одинаковы.

Чтобы *отредактировать* кнопку на области действия, выполните одну из следующих операций.

- ▶ Выполните двойной щелчок на экземпляре символа кнопки.

- ▶ Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и выберите команду Edit in Place (Правка на месте) раскрывшегося контекстного меню.

Все графические элементы на области действия, кроме редактируемой кнопки, несколько побледнеют, а вместо главной временной шкалы появится временная шкала кнопки с четырьмя ключевыми кадрами. Такой режим позволяет редактировать кнопку и изменять ее положение относительно других графических элементов сцены.

Переключиться в режим редактирования символа вы можете одним из следующих четырех способов.

- ▶ Щелкните на кнопке Edit Symbols (Правка символов), расположенной в правой части строки Scene and Symbol, и в раскрывшемся меню выберите имя символа кнопки, который вы хотите редактировать.
- ▶ Выберите на области действия экземпляр символа кнопки, щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише Ctrl (Macintosh) и активизируйте команду Edit (Правка) раскрывшегося контекстного меню.
- ▶ Выберите на области действия экземпляр символа кнопки, откройте меню Edit и вызовите команду Edit Symbol (Правка символа).
- ▶ В окне библиотеки фильма выполните двойной щелчок на значке символа либо выберите символ кнопки, откройте меню библиотеки Options и активизируйте команду Edit (Правка).

Чтобы отредактировать кнопку в отдельном окне, выделите экземпляр символа кнопки на области действия, выполните щелчок правой кнопкой мыши (Windows) или нажмите клавишу Ctrl (Macintosh) и выберите команду Edit in New Window (Правка в новом окне) раскрывшегося контекстного меню. Новое окно будет открыто непосредственно над главным окном среды Flash.

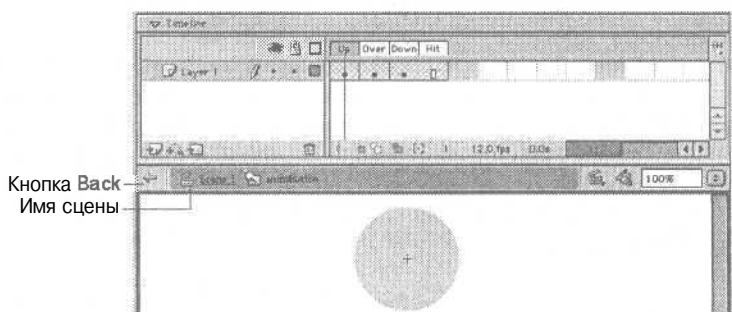


Рис. 13.4. Режим редактирования кнопки

Все способы приводят практически к одному и тому же результату: доступу к временной шкале кнопки с четырьмя ключевыми кадрами, соответствующими четырем состояниям кнопки. Получив доступ к этим кадрам, вы можете вносить

любые изменения во внешний вид кнопки. Правку кнопки можно производить с помощью всех доступных инструментов Flash, а также посредством добавления экземпляров клипов и импорта растровых изображений из внешних файлов. Помните, что изменения родительского символа влияют на все экземпляры этого символа в фильме.

После завершения редактирования кнопки, для того чтобы возвратиться к главной временной шкале, выполните *одно* из следующих действий.

- ▶ Выберите команду Edit Document меню Edit либо воспользуйтесь комбинацией клавиш `Cmd/Ctrl+E`.
- ▶ Щелкните на кнопке Back (Возврат) (рис. 13.4), расположенной в левой части строки Scene and Symbol, либо выполните щелчок на имени сцены, к которой вы хотите вернуться.

Более подробно о работе с главной временной шкалой и о переключении режимов редактирования различных элементов речь шла в главе 2.

Создание сложных кнопок

В предыдущих разделах вы познакомились с техниками создания простых кнопок и использовали их во Flash-фильмах. Обладая этими базовыми знаниями, вы можете приступить к следующему этапу — созданию сложных кнопок, соответствующих общему дизайну фильма. Такие кнопки не только выполняют роль элементов навигации, но и позволяют зрителям быстро научиться управлять воспроизведением фильма. Изучив описанные в данном разделе техники, вы поймете, насколько полезны они могут быть при создании Flash-фильмов.

Создание кнопок, содержащих несколько слоев

Кнопки — это не только элементы управления фильмом. Они являются графическими объектами, от которых значительно зависит общее восприятие фильма. Приступая к созданию кнопки, вы должны решить, нужно ли сделать ее как можно более скромной, согласовав дизайн с элементами фильма, либо превратить ее в яркий и динамичный элемент, способный воспроизводить анимационные эффекты.

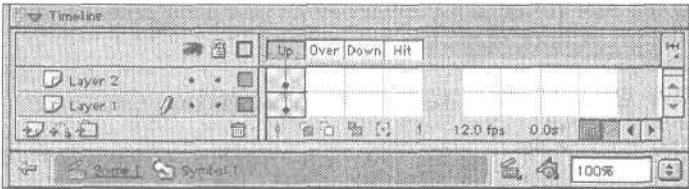


В следующем примере мы расскажем вам, как создавать динамичную кнопку, компоненты которой размещены на нескольких слоях. Исходные файлы для этого примера вы можете найти на прилагаемом компакт-диске. Файл с именем `multilayer.fla` находится в каталоге Chapter 13.

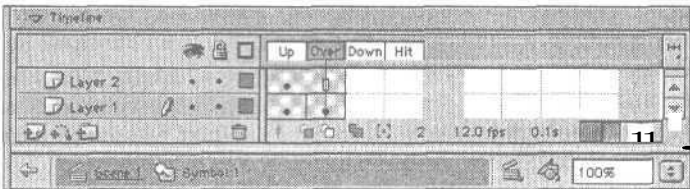
Чтобы создать кнопку, компоненты которой размещаются в нескольких слоях, выполните следующие действия.

1. Выберите команду New Symbol меню Insert либо воспользуйтесь комбинацией клавиш `Cmd/Ctrl+F8`.
2. В появившемся диалоговом окне Create New Symbol (Создание нового символа) установите переключатель типа символа в положение Button (Кнопка) и введите имя символа в поле Name, после чего нажмите кнопку ОК.
3. Flash автоматически переключится в режим редактирования символа. На временной шкале будут отображаться кадры четырех состояний кнопки: Up, Over, Down и Hit.

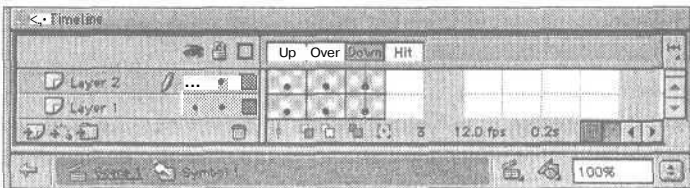
4. Преимущество разрабатываемой кнопки состоит в том, что каждый графический слой является независимым. Чтобы добавить дополнительный слой, выберите команду Layer меню Insert либо щелкните на кнопке Insert Layer (Вставить слой), расположенной в правой части временной шкалы. В результате на временной шкале появится еще один слой с пустым ключевым кадром состояния Up.
5. Используя инструменты рисования, создайте изображение кнопки в состоянии Up в слое Layer 1. Затем выберите слой Layer 2 и создайте дополнительные элементы изображения. Поскольку каждая часть изображения находится в собственном слое, вы можете легко менять изображения различных состояний кнопки.



6. Если необходимо, повторите шаги 5 и 6 для создания дополнительных слоев.
7. Закончив создание изображения кнопки для состояния Up, подготовьте изображение кнопки для состояния Over. Для этого щелкните в слое Layer 1 и вставьте ключевой кадр в кадр для состояния Over, выбрав команду Keyframe меню Insert. Flash автоматически вставит новый ключевой кадр в слой Layer 2 и создаст в нем копию изображения кнопки для состояния Up.
8. Выбрав слой Layer 1, создайте или отредактируйте изображение кнопки для состояния Over. Затем щелкните на втором кадре слоя Layer 2, вставьте в него ключевой кадр и внесите необходимые изменения в изображение.



9. Повторите процесс, описанный в пунктах 7 и 8, чтобы создать изображение кнопки для состояния Down.




10. В этом примере в качестве кнопки используется изображение помидора. Если поместить указатель мыши на кнопку, меняется цвет помидора и его корешка. При щелчке на кнопке помидор раздавливается и темнеет. Вы можете манипулировать цветами как самого помидора, так и его корешка, поскольку каждый из них находится в отдельном слое.
11. Выберите слой Layer 1 и вставьте последний, четвертый ключевой кадр для состояния Hit. Поскольку изображение кнопки в данном состоянии невидимо, необходимо вставлять ключевой кадр только в один слой.
12. По завершении создания кнопки возвратитесь к главной временной шкале, выбрав команду Edit Document меню Edit. Для предварительного просмотра кнопки активизируйте команду Enable Simple Buttons меню Control. Данный режим позволит вам увидеть, как будет меняться изображение кнопки при различных манипуляциях мышью.

Создание анимированных кнопок


В предыдущем разделе вы познакомились с техникой создания кнопок, элементы которых расположены в нескольких слоях. Такая техника позволяет создавать реалистичные изображения состояний элементов управления. Поскольку все элементы изображения находятся в различных слоях, вы можете легко изменить внешние характеристики того или иного состояния кнопки.

При перемещении указателя мыши над кнопкой ее внешний вид меняется. В момент щелчка также происходят изменения внешнего вида кнопки. Вышеописанную идею мы можем использовать при создании анимированных кнопок. В таких кнопках для представления одного из состояний используются клипы или небольшие анимации. Обычно анимация используется для отображения состояния Over, хотя любое другое состояние кнопки также может быть анимировано. Если анимация используется для представления состояния Over, имеется больше шансов, что на такую кнопку пользователь обратит внимание. Если же анимация используется для представления состояния Down, то в тот короткий период, когда на кнопке производится щелчок, зритель может не заметить анимацию. Состояние Over представляет активное состояние кнопки, когда указатель мыши находится над ней. Анимация в данном случае привлекает внимание пользователей больше, чем простое графическое изображение.

 В следующем примере мы расскажем вам, как можно создать анимированную кнопку с использованием клипа. Данный пример вы найдете на прилагаемом компакт-диске. Файл с именем `animatedButton.fla` находится в каталоге Chapter 13.

Чтобы создать анимированную кнопку, выполните следующие действия:

1. На области действия создайте изображение кнопки. Используя инструмент **Agrow**, выберите все элементы изображения кнопки. Затем преобразуйте данное изображение в символ, выбрав команду **Convert to Symbol** меню **Insert** или нажав клавишу **F8**. В появившемся диалоговом окне **Create New Symbol** установите переключатель типа символа в положение **Graphic** (Графический) и введите имя символа в поле **Name**. Щелкните на кнопке **OK**.

2. Новый символ появится в окне библиотеки. Удалите изображение из области действия. При создании каждого компонента анимированной кнопки данный графический символ будет использоваться в качестве отправной точки.
3. Создайте новый символ, выбрав команду New Symbol меню Insert либо воспользовавшись комбинацией клавиш `Cmd/Ctrl+F8`. В появившемся диалоговом окне Create New Symbol установите переключатель типа символа в положение Movie Clip (Клип) и введите имя символа в поле Name. Flash переключится в режим редактирования символа.
 
4. Теперь можно приступить к созданию клипа, который будет использоваться для отображения состояния Over. Находясь в режиме редактирования клипа, перетащите графический символ из окна библиотеки на область действия. С помощью клавиш управления курсором выровняйте изображение таким образом, чтобы его точка регистрации совпала с точкой регистрации клипа. В качестве альтернативы для автоматического выравнивания символа вы можете использовать панель Align, открыть которую можно с помощью комбинации клавиш `Cmd/Ctrl+K`. Подробно о панели Align разговор шел в главе 6.

ПРИМЕЧАНИЕ -

Поскольку графический символ будет использоваться несколько раз, необходимо осуществлять выравнивание регистрационных точек. Это позволит позиционировать каждый экземпляр символа на одном и том же месте.

5. Используя ключевые кадры, механизм автоматического построения кадров (tweening) или другие техники Flash (рис. 13.5), создайте анимацию. При необходимости внести изменения в изображение символа выберите команду Break Apart меню Modify либо воспользуйтесь комбинацией клавиш `Cmd/Ctrl+B`. Более подробно о ключевых кадрах и создании анимации разговор шел в главе 9.

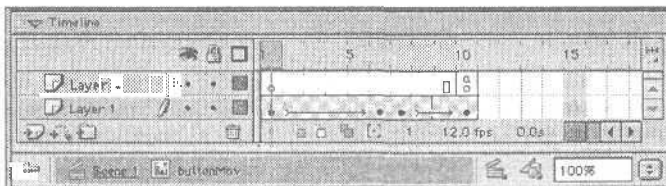


Рис. 13.5. Временная шкала клипа в файле `animatedButton.fla`

6. Завершив создание клипа, возвратитесь к главной временной шкале, выбрав команду Edit Document меню Edit.
7. На данном этапе все готово для создания кнопки. У вас есть графическое изображение и клип. Теперь необходимо их совместить в одном символе. Выберите команду New Symbol меню Insert либо воспользуйтесь комбинацией клавиш

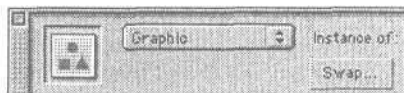
Cmd/Ctrl+F8. В появившемся диалоговом окне Create New Symbol установите переключатель типа символа в положение Button (Кнопка) и введите имя символа в поле Name.

8. Щелкните на кнопке ОК, и Flash переключится в режим редактирования символа.
9. Находясь в режиме редактирования символа кнопки, перетащите графический символ из окна библиотеки на область действия. С помощью клавиш управления курсором выровняйте изображение таким образом, чтобы его точка регистрации совпала с точкой регистрации кнопки. Обратите внимание на то, что при помещении графического символа на область действия первый кадр, соответствующий состоянию Up, превратился из пустого ключевого в ключевой кадр.

СОВЕТ

При позиционировании с помощью клавиш управления курсором объект перемещается на один пиксел при каждом нажатии на клавишу. Чтобы перемещать объект за один раз на 10 пикселей, во время операции нажмите и удерживайте клавишу Shift.

10. Итак, изображение кнопки для состояния Up готово. Можно перейти к созданию изображения кнопки для состояния Over. Вставьте в кадр, соответствующий состоянию Over, ключевой кадр, выбрав команду Keyframe меню Insert. При вставке ключевого кадра Flash автоматически перенесет графическое изображение первого ключевого кадра в новый ключевой кадр. Вставленное изображение необходимо заменить ранее созданным клипом. Для этого выберите изображение в области действия и щелкните на кнопке Swap Symbols панели Properties.



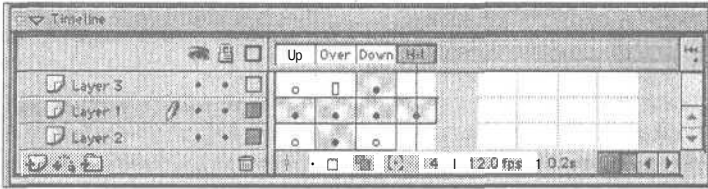
С помощью появившегося диалогового окна Swap Symbol вы сможете заменить экземпляр графического символа экземпляром клипа. Убедитесь, что на панели Properties в поле списка Symbol Behavior (Характеристика символа) выбран элемент Movie Clip, в противном случае клип работать не будет. Более подробно о замене символов речь шла в главе 7.

ПРИМЕЧАНИЕ

Вы, вероятно, заметили, что элементы кнопки в файле `animatedButton.fla` хранятся в нескольких слоях; сказанное относится как к состоянию Over, так к состоянию Down. Однако анимированные кнопки можно создавать и с использованием одного слоя.

11. Основная часть работы проделана. Ваша кнопка уже содержит изображение для состояния Up и анимацию для состояния Over. Следующим шагом будет создание изображения кнопки для состояния Down. Для этого сделайте ключевым кадр, соответствующий состоянию Down. Flash автоматически продублирует изображение предыдущего ключевого кадра в новом ключевом кадре. Используя любую из ранее описанных техник, вставьте графическое изображение, которое будет соответствовать кнопке в состоянии Down.

12. Такую же процедуру проделайте для создания изображения кнопки в состоянии Hit в четвертом кадре. Когда все ключевые кадры получат изображения всех четырех состояний кнопки, анимированная кнопка будет готова к использованию.



13. Возвратитесь к главной временной шкале, выбрав команду Edit Document меню Edit, и сохраните документ. Поскольку нами был использован клип, вы не сможете проверить работу анимированной кнопки в режиме разработки. Чтобы посмотреть на кнопку в действии, необходимо создать экземпляр этой кнопки в фильме, а затем выбрать команду Test Movie меню Control.

Анимированные кнопки ничем не отличаются от остальных кнопок. Если в библиотеке хранится символ кнопки, вы можете многократно использовать ее в фильме. С каждым экземпляром анимированной кнопки могут быть связаны действия, с помощью которых вы сможете управлять ходом воспроизведения фильма.

Создание многокнопочных интерфейсов

Работая с кнопками, вы могли заметить, что, щелкнув на одной из них, вы иногда продолжаете видеть предыдущее состояние, Over. На первый взгляд это может показаться ошибкой, однако, поскольку указатель находится над изображением кнопки, Flash все равно будет реагировать на щелчок корректно.

Существует возможность создания кнопок, при работе с которыми данный эффект возникать не будет. Кроме того, вы можете разрабатывать многокнопочные интерфейсы или навигационные панели, где каждая отдельная кнопка может вызывать воспроизведение определенной части фильма и служить в качестве «книжной закладки».



В следующем примере мы расскажем вам, как создавать навигационную панель с двумя кнопками. Хотя данная задача не является сложной, используемая техника может быть применена для создания больших и более сложных интерфейсов. Необходимые файлы вы найдете на прилагаемом компакт-диске. Файл с именем navBar.fla находится в каталоге Chapter 13.

ПРИМЕЧАНИЕ

Хотя описываемый процесс достаточно прост, в нем особое внимание уделяется обеспечению согласованности элементов панели. Все инструкции необходимо выполнять предельно внимательно!

Чтобы создать навигационную панель с несколькими кнопками, сделайте следующее.

1. Прежде всего необходимо создать кнопку. Достаточно будет создать один символ кнопки, хотя в своем проекте вы можете использовать несколько различных кнопок. Процесс создания простой кнопки был описан в одном из предыдущих разделов. После того как кнопка будет готова, возвратитесь к главной временной шкале фильма.
2. Имея в своем распоряжении кнопку, вы можете приступить к разработке навигационной панели. С помощью инструмента Rectangle создайте прямоугольник на всю ширину области действия. Высота прямоугольника должна быть достаточной для размещения кнопки.



3. Теперь можно добавить на панель кнопку. Добавьте новый слой, выбрав команду New Layer меню Insert. Нет технической необходимости размещать элементы в различных слоях, однако хорошая организация дела всегда помогает в работе. Присвойте новому слою имя buttons. Более подробно об организации содержимого слоя и работе со слоями речь шла в главе 8.
4. С помощью мыши перетащите символ кнопки из окна библиотеки в левый угол изображения навигационной панели и убедитесь, что экземпляр кнопки был помещен в слой buttons.
5. Повторите шаг 4, создав второй экземпляр кнопки. На этот раз поместите изображение справа от первой кнопки. Эти две кнопки будут служить элементами управления навигационной панели.



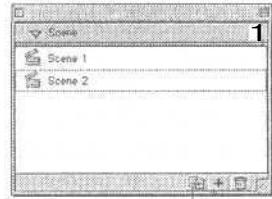
6. Далее необходимо добавить какой-либо текст, который поможет определять, с какой частью фильма связана та или иная кнопка. Создайте новый слой и назовите его text. Используя инструмент Text, создайте надписи, как указано ниже в инструкции.
 - > Рядом с первой кнопкой создайте текстовое поле и введите в нем надпись Scene 1.
 - > Рядом со второй кнопкой создайте текстовое поле и введите в нем надпись Scene 2.

- > На панели Properties увеличьте величину знаков и введите надпись Scene 1 в центре области действия. Данное текстовое поле будет использоваться в качестве временного маркера при тестировании кнопок навигационной панели.

Более подробно об использовании инструмента Text мы рассказывали в главе 5.

7. Последним этапом настройки навигационной панели будет связывание действия с кадром для корректного воспроизведения сцен фильма. Создайте новый слой, выбрав команду Layer меню Insert, и назовите его actions. Щелкните в первом пустом ключевом кадре слоя actions и откройте панель Actions, выбрав команду Actions меню Window, или нажмите клавишу F9. Откройте подкатегорию Actions ▶ Movie Control и с помощью мыши перетащите действие stop() на область отображения сценария.
8. Создание всех базовых элементов первой сцены завершено. Теперь необходимо создать еще одну сцену.
9. Для этого откройте панель Scene, выбрав команду Scene меню Window или воспользовавшись комбинацией клавиш Shift+F2. Создайте дубликат первой сцены, для чего щелкните на кнопке Duplicate Scene (Дублировать сцену) в нижней части панели Scene.

По умолчанию Flash присвоит дубликату имя Scene 1 сору. Щелкните на имени сцены и присвойте ей новое имя Scene 2.



Кнопка Duplicate Scene

СОВЕТ

При создании навигационных панелей дублирование сцен очень полезно. Поскольку новая сцена является точной копией первой сцены, все элементы навигационной панели будут расположены в тех же позициях.

10. Используя инструмент Text, исправьте надпись, расположенную в центре области действия, на Scene 2. Каждая последующая сцена должна несколько отличаться от предыдущей. При проверке фильма это позволит нам определить, работают все сцены корректно или нет.
- И. Теперь необходимо связать с кнопкой действие, чтобы при щелчке на ней осуществлялся переход к новой сцене. Перейдите к сцене Scene 1. Выберите кнопку, расположенную справа. Откройте панель Actions, для чего выберите команду Actions меню Window.
12. Свяжите с кнопкой действие, при выполнении которого будет осуществляться переход к первому кадру сцены Scene 2. Код действия будет следующим:

```
on(release){
    gotoAndPlay("Scene 2",1);
}
```

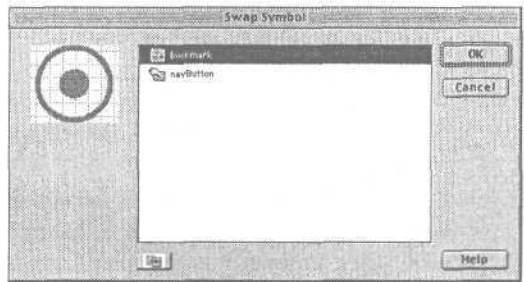
Более подробно о связывании действий мы рассказывали в главе 12.

13. Используйте панель Scene для перехода к сцене Scene 2. Повторите те же шаги для кнопки, расположенной слева. Однако код действия должен выглядеть так:

```
on(release){
    gotoAndPlay("Scene 1",1);
}
```

14. Вы создали навигационную панель, с помощью которой можно переключаться между сценами фильма. Чтобы проверить работу кнопок, выберите команду Test Movie меню Control.
15. Убедившись, что все работает корректно, вы можете дополнить навигационную панель еще одной функцией. Чтобы зритель всегда знал, какой фрагмент фильма в данный момент воспроизводится, навигационная панель может выполнять функцию закладки. Чтобы добиться этого, необходимо лишь настроить некоторые графические элементы фильма.
- Откройте окно библиотеки, выбрав команду Library меню Window либо нажав клавишу F11.
16. Выполните двойной щелчок на значке кнопки в окне библиотеки, чтобы открыть данный символ в режиме редактирования. Щелкните на третьем ключевом кадре (состояние Down).
17. Создайте новый символ, выбрав команду New Symbol меню Insert. В диалоговом окне Create New Symbol переключатель типа символа установите в положение Graphic (Графический) и присвойте новому символу имя bookmark. Щелкните на кнопке ОК и перейдите к главной временной шкале, выбрав команду Edit Document меню Edit.
18. Используя новый графический символ bookmark, вы можете создать элемент навигационной панели, который будет действовать как закладка. Перейдите к сцене Scene 1 и щелкните на первой кнопке навигационной панели.

19. На панели Properties щелкните на кнопке Swap Symbol (Замена символа). В открывшемся диалоговом окне Swap Symbol замените экземпляр символа кнопки Scene 1 на графический символ, который вы только что создали. При этом на область действия вместо кнопки будет помещен графический символ, который будет исполнять роль закладки в сцене Scene 1.



20. Перейдите к сцене Scene 2 и повторите шаги 18 и 19, однако на этот раз замените кнопку, расположенную справа.
21. Итак, создание фильма завершено. Точнее, завершено создание навигационной панели. Чтобы проверить ее работоспособность, выберите команду Test Movie меню Control. Обратите внимание на то, как символ bookmark действует в качестве закладки. При щелчке на кнопке для перехода к определенной сцене изображение кнопки не меняется.

Хотя в данном разделе описан процесс разработки навигационной панели всего с двумя кнопками, данную технику можно использовать для создания навигационной панели всего Web-узла. Используя различные графические изображения, сцены и действия, вы можете создавать навигационные средства для любого Flash-фильма.

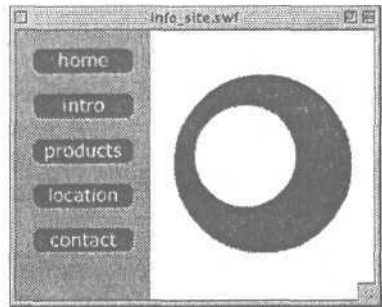
MX Для расширения возможностей интерактивных элементов управления в новой версии Flash MX вы можете использовать объект Button языка ActionScript. Более подробно об объекте Button и об альтернативном методе создания навигационных панелей с помощью сценариев ActionScript мы расскажем в главе 16.

Создание «некнопочных» кнопок

Что такое «некнопочные» кнопки? Под ними подразумеваются любые элементы, которые действуют в фильме как кнопки, но внешне на них не похожи. Давайте внесем ясность в этот вопрос.

По своей природе интерактивные продукты предусматривают наличие какого-либо элемента управления. Зрители должны иметь возможность настраивать изображение в соответствии со своими предпочтениями. В качестве элемента управления обычно используется кнопка. При щелчке на кнопке происходят те или иные изменения в фильме. Такая техника проста и эффективна.

Однако проблема состоит в том, что в некоторых проектах использование кнопок не столь эффективно. Безусловно, для создания навигационных панелей определенных информационных структур кнопки вполне пригодны. Например, если вам необходимо представить большой объем информации, вы можете разбить ее на разделы и использовать кнопки для перехода от одного раздела к другому. Такая схема удобна как для дизайнера, так и для зрителей. Но иногда бывает необходимо создать презентацию, в которой сам вид кнопки должен заинтересовать зрителя. В таком случае вам необходимо использовать навигационную структуру наподобие той, что используется для газетных интернет-изданий.



В фильме «некнопочная» кнопка должна действовать как обычная кнопка и, кроме того, выражать идею проекта. Кнопки — это элементы, хорошо знакомые зрителям. Они прекрасно работают в качестве элементов навигационных панелей. Однако в некоторых специальных фильмах использование простых кнопок может испортить впечатление от всего проекта.

Если вы хотите сразу привлечь внимание зрителей, то лучше применять так называемые «некнопочные» кнопки. Их использование в качестве элементов навигационных панелей позволяет передать идею проекта непосредственно через интерфейс, что создает эффект целостности интерфейса и содержимого фильма.

Вот несколько примеров проектов, в которых целесообразно использовать «некнопочные» кнопки:

- ▶ анимированные детские сказки и истории, где кнопка должна давать представление о характере рассказа и может выполняться в виде изображения книжки, волшебной палочки или других элементов сказочного жанра;
- ▶ музыкальные проигрыватели или микшеры, где элементы управления выглядят как кнопки реальной музыкальной аппаратуры либо как части музыкальных инструментов;
- ▶ мультфильмы, в которых используются характерные жесты или анимированные элементы диалога, помогающие зрителям запомнить проект.

Используя «некнопочные» кнопки в фильмах Flash (или в любом другом проекте), вы помогаете зрителям воспринимать идею, которая выражена в фильме. Данный подход также помогает проявить творческие способности посредством установки некоторых ограничений. Вы должны работать в рамках тех правил, которые либо установлены вами, либо обусловлены особенностями проекта. Чем меньше доступных ресурсов, тем больше изобретательности необходимо проявить для решения тех или иных задач. Попробуйте в следующем своем Flash-проекте использовать «некнопочные» кнопки и посмотрите, каков будет результат.

В разделе «Разработки» главы 14 мы познакомим вас с замечательным проектом под названием «Goodnight Mr. Snoozleberg». В этой интерактивной игре используется множество различных кнопок, помогающих мистеру Снузлебергу безопасно проходить различные уровни. Обратите внимание на то, что, хотя кнопки выполнены в виде элементов сцены, действуют они как обычные кнопки.

Разработки

Работа над интерактивными медиа-проектами состоит из нескольких этапов. Кроме создания содержимого, разработчику необходимо найти наиболее эффективный способ передачи идеи проекта зрителям. Какой бы прекрасной ни была анимация, если зрители не смогут разобраться с тем, как управлять воспроизведением фильма, они не смогут его оцепить.

Вот почему важны интуитивно-понятные интерактивные элементы управления. Хорошо знакомый дизайн может быть куда эффективнее сложных технических приемов. Помните, что зритель будет смотреть ваши фильмы впервые. Некоторые вещи, которые вам как разработчику кажутся очевидными, могут запутать зрителя, если их дизайн слишком замысловат. Однако есть люди, умеющие совмещать в одном проекте хороший дизайн и сложные технические приемы. Созданные ими фильмы, несмотря на доступность и простоту управления, способны произвести впечатление на самого искушенного зрителя. Вашему вниманию мы хотим представить Web-узел компании THUNKdesign из Сан-Франциско www.thunkdesign.com (рис. 13.6). Данный Web-узел может служить примером как безупречно подобранного содержимого, так и хорошо созданной системы навигации.



Рис. 13.6. Web-узел компании THUNKdesign

Резюме

В этой главе вы изучили техники создания кнопок, используемых для управления воспроизведением фильма. Используя базовые действия, о которых речь шла в предыдущей главе, вы теперь способны создавать динамические, интерактивные продукты Flash. В дополнение к использованию стандартных кнопок вы научились создавать собственные кнопки, тестировать их, а также редактировать и изменять символы кнопок. В данной главе также были рассмотрены некоторые техники создания нестандартных элементов управления для более сложных проектов. Используя такие техники, можно создавать элементы управления, которые будут выступать в роли фрагментов общего дизайна интерфейса, где каждый компонент является проводником идеи, выражаемой в фильме.

Практикум 4

Создание интерактивного резюме с использованием базовых действий и навигационной панели

Данный практикум позволит вам закрепить полученные знания и приобрести ряд практических навыков. Для его выполнения необходимо задействовать все техники, о которых шла речь в двух предыдущих главах. А как вы помните, эти главы были посвящены созданию интерактивных элементов с использованием базовых действий, а также работе с кнопками для создания источников интерактивного управления.

Подготовка области действия


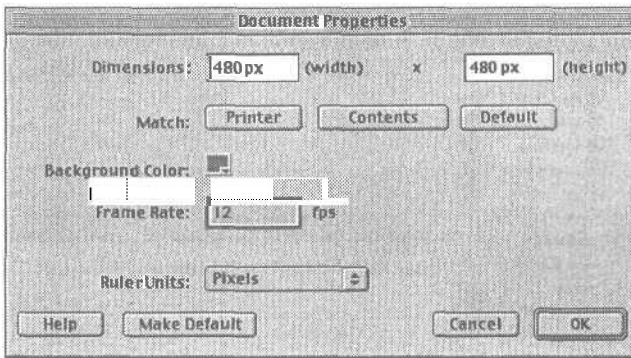
 Все необходимые для данного практикума файлы вы найдете на прилагаемом к книге компакт-диске. Файл с именем `resume.La` находится в каталоге Hands On 4. Для выполнения задания (рис. П4.1) использовать именно эти файлы не обязательно. Более того, мы настоятельно рекомендуем работать с собственными файлами или создавать содержимое непосредственно в среде Flash. Польза от выполнения настоящего задания будет куда большей, если вы самостоятельно разработаете все компоненты, а не просто начнете применять уже готовую графику, повторяя шаги инструкции.



Рис. П4.1. Внешний вид конечного интерактивного резюме

Задача проекта достаточно проста — создать интерактивное резюме, в котором была бы представлена наиболее важная персональная информация. Для выполнения поставленной цели вы должны уметь использовать базовые действия, кнопки, применять текстовый ввод, а также хорошо ориентироваться в среде разработки Flash. Так приступим же к работе!

1. Работу над проектом необходимо начать с установки размера области действия. Откройте диалоговое окно Document Properties (Свойства документа), выбрав команду Document меню Modify или воспользовавшись комбинацией клавиш **Cmd/Ctrl+J**. Задайте значения ширины и высоты в полях Width и Height, равными 480. В образце цвета фона Background color выберите темно-серый цвет (#666666). Щелкните на кнопке OK.



2. С помощью инструмента Rectangle (Прямоугольник) создайте два прямоугольника, которые будут выполнять роль верхнего и нижнего колонтитулов. Не делайте прямоугольники слишком большими. Их высота должна составлять примерно 50 пикселей. Данный фильм будет содержать большой объем информации, поэтому середина области действия должна быть свободной. Точный размер прямоугольников можно установить с помощью панели Info, открываемой посредством комбинации клавиш **Cmd/Ctrl+I**, либо с помощью линеек. Для отображения линеек вокруг области действия выберите команду Rulers меню View.
3. Каждое резюме должно содержать имя человека, информация о котором в нем содержится. С помощью инструмента Text введите в левой части верхнего прямоугольника свое имя. Для определения типа, цвета и размера шрифта воспользуйтесь панелью Properties. Об использовании инструмента Text достаточно подробно рассказано в главе 5.
4. В верхнем прямоугольнике, справа от имени, создайте кнопку для отправки электронного сообщения — в таком случае человек, прочитавший резюме, получит возможность связаться с вами и, возможно, предложить за работу солидную сумму. Процесс создания кнопки был описан в главе 13. В файле на компакт-диске используется кнопка в виде конверта, при щелчке на которой (состояние Down) изображение превращается в запечатанный конверт со штампом.

5. Когда кнопка создана, необходимо связать с ней действие. Выберите в области действия кнопку и откройте панель Actions, задав команду Actions меню Window.
6. Откройте подкатегорию Browser/Network и выберите действие `getURL()`. Выполните на значке действия двойной щелчок либо перетащите его с помощью мыши в область отображения сценария. В поле URL области отображения параметров введите `mailto:myAddress@myDomain`, заменив `myAddress` и `myDomain` реальной информацией.



Рис. П4. 2. Панель Actions с выбранным действием `getURL()`

При щелчке *на* кнопке выполняется действие `getURL()`, основанное на протоколе `mailto` (часть HTML), в результате чего запускается соответствующее приложение для отправки электронных сообщений. Данная техника работает корректно, если ваше резюме просматривается в окне браузера либо распространяется в виде файла Flash-projector.

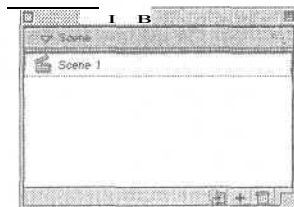
7. Сохраните файл и присвойте ему имя. Теперь можно приступать к созданию других компонентов вашего резюме.

Создание интерактивных элементов

8. Создаваемое вами резюме будет содержать четыре части: Contact (контактная информация), Education (образование), Experience (опыт работы) и References (характеристики). Мы можем использовать созданную ранее цепку в качестве

базовой для каждого раздела. Для этого необходимо создать три дубликата сцены Scene 1. Делается это следующим образом.

- > Откройте панель Scene, выбрав в меню Window команду Scene либо воспользовавшись комбинацией клавиш Shift+F2. Указанное окно содержит пока только одну сцену. Отметьте ее (сцена Scene 1) и щелкните по кнопке Duplicate Scene (Дублировать сцену) в нижней части панели.
- > На панели Scene появится сцена Scene 1 copy. Выполните на ее значке двойной щелчок и переименуйте ее в Scene 2.
- > Повторите описанные действия еще два раза и назовите новые сцены соответственно Scene 3 и Scene 4. Переключаться между сценами вы можете либо посредством панели Scene, либо с помощью кнопки Edit Scene (Правка сцены).

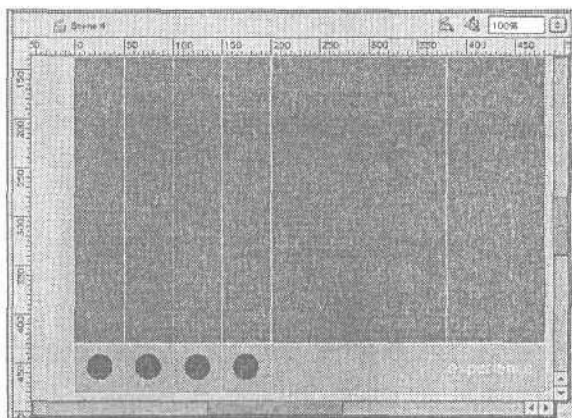


Кнопка Duplicate Scene

Теперь, когда у вас имеется подходящий шаблон для резюме, можете приступить к созданию интерактивных элементов.

9. Прежде всего необходимо создать кнопки для перехода к различным разделам резюме. Выберите в меню Insert команду New Symbol либо нажмите комбинацию клавиш Cmd/Ctrl+F8. В открывшемся диалоговом окне Create New Symbol (Создать новый символ) установите переключатель Behavior (Характеристика) в положение Button и введите имя символа в поле Name. Щелкните на кнопке OK, и Flash автоматически переключится в режим редактирования символа.
10. Далее мы создадим эффект линейного переключателя. Для этого необходимо использовать временные элементы для выравнивания и равномерного распределения кнопок.

Перейдите к сцене Scene 4, создайте четыре изображения ложных кнопок, а также надпись contact и разместите их в нижнем прямоугольнике. Помните, что текстовое поле может содержать и более длинные слова, чем contact. И изображения кнопок, и надпись будут служить временными заполнителями. На данном этапе ваша сцена должна выглядеть так, как показано на рисунке.



СОВЕТ

Для того чтобы выравнивать временные кнопки, вы можете воспользоваться линейками (View ▸ Rulers) либо панелью Align, открываемой с помощью команды Align меню Window.

11. Теперь все готово для создания первой кнопки. Удерживая нажатой клавишу Shift, выберите текстовое поле contact и первую слева кнопку в области действия. Активизируйте команду Convert to Symbol (Преобразовать в символ) меню Insert либо нажмите клавишу F8.
12. Установите переключатель типа символа в положение Button (Кнопка) и введите в поле Name имя conButton. Щелкните на кнопке ОК. Теперь новую кнопку необходимо несколько изменить. Откройте меню Edit Symbols (Правка символов) и выберите элемент conButton либо выполните двойной щелчок на значке символа conButton в окне библиотеки. Flash автоматически переключится в режим редактирования символа.
13. С помощью команды Keyframe меню Insert вставьте три ключевых кадра. Будучи созданными на временной шкале кнопки, все они являются точными копиями первого ключевого кадра.
14. Далее необходимо отредактировать изображение кнопки для каждого состояния. Состояния Up и Over на данном этапе оставьте без изменений. Изображения кнопок в этих состояниях остаются прежними и в дальнейшем они будут использованы для выравнивания всех кнопок. Для состояния Down следует изменить цвет кнопки, а для состояния Hit — удалить лишь надпись, оставив нетронутым изображение кнопки. На щелчок мыши должна реагировать только основная область кнопки.
15. Повторите действия, перечисленные в пунктах 11–14, по отношению к трем остальным кнопкам. Все созданные кнопки и соответствующие им надписи представлены на рис. П4.3.

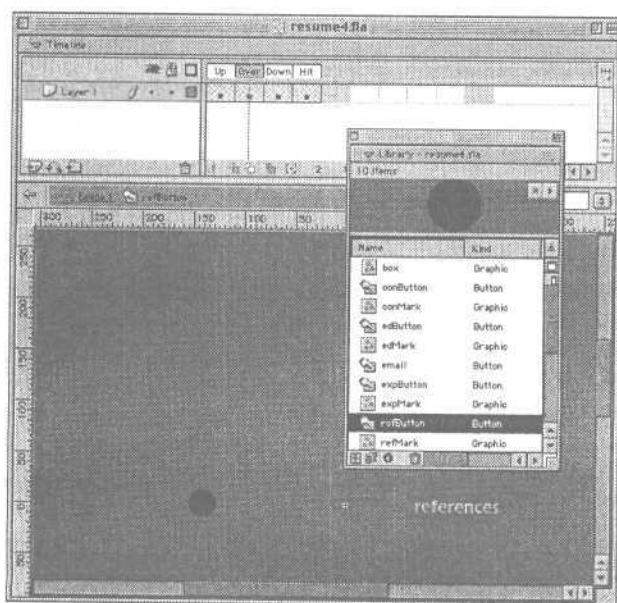


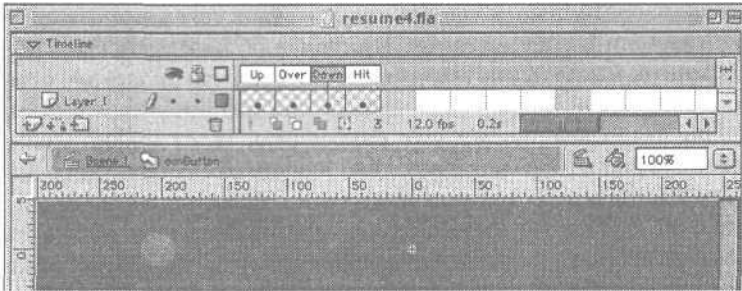
Рис. П4.3. Каждая кнопка связана с определенным текстом в резюме — education, experience и references, — поэтому им присваиваются имена edButton, expButton и refButton

Создание элементов навигации

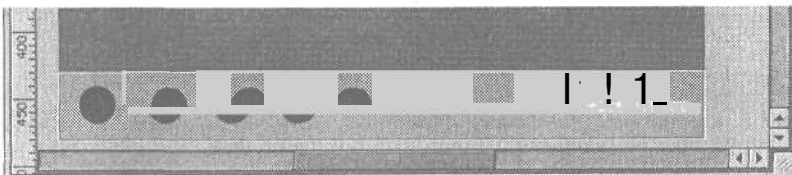
16. Создаваемая в нижней части резюме навигационная панель должна содержать кнопки с закладками, чтобы читающий резюме человек всегда знал, в каком разделе он в данный момент находится. (О создании навигационных панелей разговор шел в главе 13.)

Для создания закладки, являющейся одним из наиболее важных элементов резюме, мы воспользуемся кнопкой, которая была создана на этапе, описанном в пунктах 11-14.

17. Откройте окно библиотеки, выбрав в меню Window команду Library. Выполните двойной щелчок на значке символа кнопки `conButton`. Flash автоматически переключится в режим редактирования символа.
18. Щелкните на ключевом кадре состояния Down и выберите как графический, так и текстовый элементы кнопки. Откройте меню Insert и активизируйте команду Convert to Symbol (Преобразовать в символ) либо нажмите клавишу F8.



19. В открывшемся диалоговом окне Create New Symbol (Создать новый символ) установите переключатель Behavior в положение Graphic (Графический) и введите в поле Name имя `conMark`. Щелкните на кнопке OK.
20. Повторите эту процедуру для состояния Down трех остальных кнопок. Назовите создаваемые графические символы именами `edMark`, `expMark` и `reMark` соответственно. Сохраните фильм и возвратитесь к главной временной шкале сцены Scene 1.
21. Теперь, когда все кнопки и закладки созданы, их можно добавить в фильм. Вставьте новый слой в сцену Scene 1 и назовите его `buttons`. С помощью мыши поочередно перетащите символы кнопок из окна библиотеки на область действия и выровняйте их таким образом, чтобы они совпали с изображениями ложных кнопок. Надписи для всех кнопок будут накладываться одна на другую. Но на данном этапе это поможет правильно позиционировать все кнопки.



22. Итак, ваши кнопки размещены в сцене Scene 1. Теперь их необходимо перенести в три остальные сцены. Щелкните на первом ключевом кадре слоя buttons сцены Scene 1 и выберите в меню Edit команду Copy (Копировать). Выполните для трех остальных (2-4) сцен действия, указанные ниже:

- > Вставьте новый слой и назовите его buttons.
- > Выберите в меню Edit команду Paste in Place (Местная вставка).

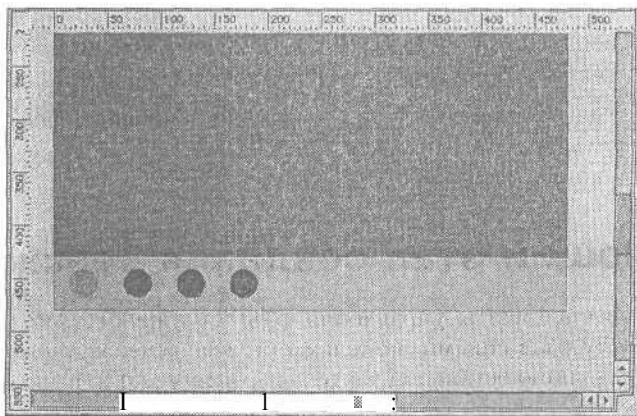
Не забудьте до вставки кнопок удалить изображения ложных кнопок из сцены Scene 4.

23. Если вы уже перенесли кнопки на сцены, приступайте к вставке изображений закладок. Перейдите к сцене Scene 1 и выберите экземпляр conButton.
24. На панели Properties щелкните на кнопке Swap Symbol (Замена символа) и в открывшемся диалоговом окне Swap Symbol замените символ conButton графическим символом conMark. Выполните указанные действия и в отношении символов в сценах Scene 2, Scene 3, Scene 4.

25. После того как все закладки будут вставлены, вам придется несколько доработать кнопки, созданные ранее. На данном этапе текстовые компоненты кнопок отображаются в состоянии Up. Причем в области действия все надписи, как вы помните, накладываются одна на другую, что делает их чтение невозможным. Ранее мы использовали надписи для позиционирования кнопок, однако теперь, когда все кнопки уже размещены, текст можно удалить. А поскольку он должен отображаться только в состояниях Over и Down, необходимо удалить надпись в состоянии Up для всех экземпляров кнопок.

Откройте окно библиотеки и выполните двойной щелчок на значке символа кнопки conButton. Flash переключится в режим редактирования символа. Убедитесь, что головка воспроизведения находится в первом кадре (состояние Up) и удалите из кадра текст contact. Повторите указанные действия и в отношении трех других кнопок. Ни одна кнопка в состоянии Up не должна содержать текст.

То же самое необходимо проделать и с графическими символами закладок. Откройте каждый графический символ в режиме редактирования и удалите текст. Все экземпляры этих символов будут автоматически обновлены.



26. После очистки кнопок от ненужных элементов ваш фильм начинает приобретать требуемый внешний вид. Каждая сцена имеет свою закладку, что помогает определить, в каком разделе мы находимся. Теперь, когда все кнопки и закладки установлены, с ними можно связать действия для переключения между различными сценами.

ПРИМЕЧАНИЕ

На данном этапе фильм обязательно нужно сохранить. Для этой цели мы рекомендуем выбрать команду `Save as` и сохранить файл под другим именем. Благодаря этому, если в дальнейшем возникнут проблемы, вы сможете вернуться к сохраненному ранее файлу и продолжить работу.

Написание сценария

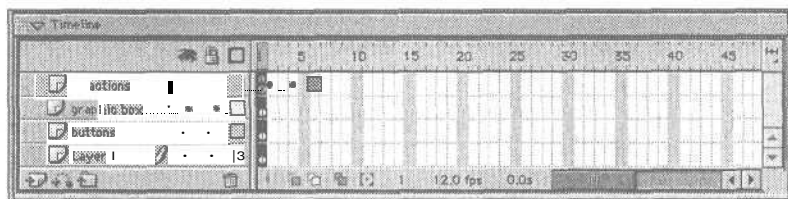
27. Для каждой сцены выполните такие действия:
- > Вставьте новый слой и присвойте ему имя `actions`.
 - > Выберите первый кадр этого слоя и свяжите с ним действие `stop()`.
28. Во всех сценах фильма свяжите с каждой из кнопок определенные действия, чтобы при щелчке на кнопке осуществлялся переход к нужной сцене:
- > При щелчке на кнопке `conButton` — переход к первому кадру сцены `Scene 1`.
 - ▷ При щелчке на кнопке `edButton` — переход к первому кадру сцены `Scene 2`.
 - > При щелчке на кнопке `expButton` — переход к первому кадру сцены `Scene 3`.
 - > При щелчке на кнопке `refButton` — переход к первому кадру сцены `Scene 4`.
- Например, сценарий для кнопки `edButton` в сценах 1, 2 и 4 должен быть следующим:
- ```
on(release){
 gotoAndPlay("scene 2",1);
}
```
- И независимо от того, какая в данный момент воспроизводится сцена, действие всегда будет перемещать головку воспроизведения к первому кадру сцены `Scene 2`.
- Это действие необходимо связать со всеми экземплярами данной кнопки в каждой сцене фильма. Успехов вам в написании сценариев!
29. Когда процесс связывания действий с кнопками будет завершен, фильм необходимо проверить, для чего следует воспользоваться командой `Test Movieview` (Тестировать фильм) меню `Control` (Управление).

## Завершающий этап создания резюме

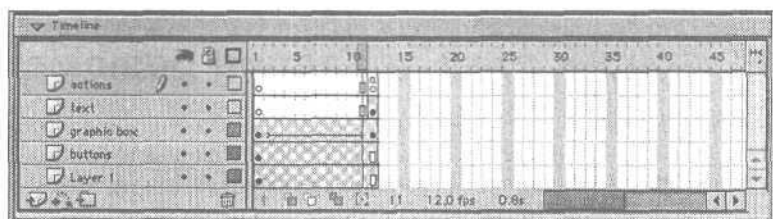
30. Теперь, когда все элементы управления функционируют и вы можете переключаться между различными разделами резюме (сценами фильма), можно приступить к созданию остальной части содержимого. Для вывода и размещения текстовой информации должно быть специальное поле.

Находясь в сцене Scene 1, вставьте новый слой и назовите его graphic box. В центре области действия, на новом слое, создайте поле для текста (в виде прямоугольника). После установки размеров и выполнения выравнивания активизируйте в меню Insert команду Convert to Symbol (Преобразовать в символ) либо нажмите клавишу F8. Установите переключатель Behavior в положение Graphic (Графический) и укажите имя в поле Name.

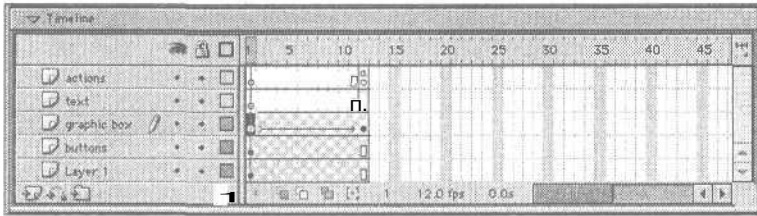
31. Выберите в области действия экземпляр символа прямоугольника, откройте меню Edit и выполните команду Copy. Затем в каждой сцене фильма добавьте по новому слою, назовите его graphic box и активизируйте в меню Edit команду Paste in Place (Местная вставка).
32. Следующим этапом работы будет анимирование символа текстового поля. Чтобы создать анимацию, во временной шкале необходимо добавить кадры. В сцене Scene 1, удерживая нажатой клавишу Shift, выполните щелчок на первом кадре каждого слоя. После того как все первые кадры будут отмечены, нажмите 11 раз клавишу F5. В результате временная шкала сцены Scene 1 должна содержать 12 ключевых кадров. Выполните указанные действия и в отношении сцен 2-4.
33. Далее действие `stop()`, связанное с первым кадром, необходимо перенести в кадр 12. Это нетрудно сделать вручную, с помощью мыши. Выберите ключевой кадр, с которым связано действие, переместите указатель мыши на значок действия в кадре, нажмите кнопку мыши и, удерживая ее, перетащите действие в кадр 12 слоя actions в каждой сцене фильма.



34. После перенесения действия в кадр 12 мы получили некоторое пространство для создания анимации. Более подробно о создании анимации с автоматическим построением промежуточных кадров мы рассказывали в главе 9. Щелкните на последнем кадре (кадр с номером 12) слоя graphic box и активизируйте в меню Insert команду Keyframe (Ключевой кадр).
35. Выберите экземпляр символа в первом ключевом кадре слоя graphic box и с помощью панели Properties установите значение параметра Alpha равным 0. Данное значение можно ввести в поле этого параметра либо установить с помощью ползунка.



36. Следующим шагом будет создание анимации с автоматическим построением промежуточных кадров. Значение Alpha для символа graphic box здесь будет плавно возрастать от 0 до 100 %. Поскольку кадр с номером 12 не был изменен, значение Alpha в нем останется равным 100 %. Для создания анимации воспользуйтесь ключевым кадром 1. В поле списка Tween на панели Properties отметьте элемент Motion (Движение), и Flash автоматически создаст промежуточные кадры анимации, просмотреть которые вы сможете, нажав клавишу Return/Enter.



37. Для того чтобы создать такую же анимацию в остальных сценах, повторите действия, перечисленные в пунктах 34–36.
38. Итак, мы подошли к заключительному этапу создания резюме — вставке текстовой информации. Текст необходимо поместить на отдельный слой, с тем чтобы с ним можно было свободно работать, не нарушая порядок расположения других элементов фильма. Прежде всего вы должны дать каждому разделу резюме название. В сцене Scene 1 вставьте новый слой и назовите его text. Выберите кадр с номером 12 и добавьте ключевой кадр.
39. На панели Properties укажите тип, размер и стиль текста титульного названия. Далее, используя инструмент Text, введите текст названия данного раздела. В файле resume, находящемся на прилагаемом компакт-диске, титульная надпись такая же, как и текст закладки этого раздела.
40. Повторите действия, описанные в пунктах 38 и 39, для остальных сцен фильма. Чтобы точно позиционировать текстовые блоки в каждой из сцен, используйте команду Paste in Place (Местная вставка).
41. С помощью инструмента Text введите дополнительную текстовую информацию о себе: предыдущие должности, опыт работы, а также другие сведения, которые могут заинтересовать работодателя. Более подробно о работе с инструментом Text речь шла в главе 5.
42. Завершив ввод текстовых данных, сохраните файл и выберите в меню Control команду Test Movie.

Поздравляем! Вы только что завершили один из своих первых больших проектов, основанных на использовании приложения Flash MX. Но данный учебный раздел — это всего лишь отправная точка. Освоив базовые принципы работы с Flash, вы сможете самостоятельно совершенствовать техники и стратегии создания сложных резюме и портфелей связанных документов. Продолжайте работать, и пусть вам улыбнется удача!



# ЧАСТЬ 5

---

## Язык ActionScript

*До настоящего момента мы использовали только те средства Flash, которые, собственно, и создали репутацию программе: полностью векторная графика, несколько способов создания анимации и интуитивно понятный пользовательский интерфейс, обеспечивающий удобство при работе. С помощью простых сценариев, которые управляют процессом воспроизведения, вы научились создавать интерактивные фильмы. Однако за последнее время Flash превратился в более мощный продукт. В частности, значительно возросли возможности программы в области создания сценариев. Язык ActionScript из простого «регулирующего», управляющего временной шкалой, превратился в «начальника фильма», который позволяет создавать увлекательнейшие мультимедийные продукты. В этой части книги мы подробно проанализируем возможности языка ActionScript, с тем чтобы научиться использовать его в качестве инструмента для создания фильмов, отвечающих разнообразнейшим требованиям аудитории.*

|                                                                                    |     |
|------------------------------------------------------------------------------------|-----|
| <b>Глава 14.</b> Использование ActionScript.....                                   | 280 |
| <b>Глава 15.</b> Написание сценариев ActionScript .....                            | 301 |
| <b>Глава 16.</b> Повышение интерактивности фильмов<br>с помощью ActionScript ..... | 319 |
| <b>Глава 17.</b> Управление вводом и выводом данных.....                           | 349 |
| <b>Глава 18.</b> Отладка сценариев ActionScript.....                               | 362 |



## Глава 14

# Использование ActionScript

Итак, вы решились приоткрыть занавес в мир ActionScript. Поздравляем! Ведь это один из самых интересных этапов работы во Flash, хотя, с другой стороны, и один из наиболее сложных. Мы говорим об этом не для того, чтобы напугать вас, а для того, чтобы вы приступили к его освоению во всеоружии. Ведь изучить ActionScript и быть с ним «на ты» — задача не из легких. Достичь такого уровня, чтобы максимально эффективно использовать этот язык, достаточно сложно, но вполне возможно. И главное здесь — это усердная работа. Процесс изучения ActionScript, как и любого другого языка программирования, подобен процессу изучения иностранного языка: чтобы бегло говорить, необходимо практиковаться ежедневно. В данном случае можно создавать простые фильмы, используя различные возможности ActionScript. Через некоторое время вы начнете совсем по-другому воспринимать и чувствовать этот язык. И чем больше будете практиковаться, тем лучше.

В настоящей главе, где описаны ключевые термины и понятия, связанные с языком ActionScript, рассмотрены следующие темы:

- ▶ ActionScript как объектно-ориентированный язык;
- ▶ компоненты сценария;
- ▶ выполнение сценария;
- ▶ планирование сценариев;
- ▶ термины, относящиеся к языку ActionScript;
- ▶ работа с панелью Actions в режиме Expert.

## Знакомство с ActionScript

Язык ActionScript очень похож на язык JavaScript. Фактически ActionScript разработан на основе спецификаций JavaScript, установленных Европейской Ассоциацией производителей компьютеров (European Computer Manufacturers Association, ECMA). Дополнительную информацию об истории, роли и членах ECMA можно получить по адресу [www.ecma.ch](http://www.ecma.ch).

ActionScript, будучи объектно-ориентированным языком подготовки сценариев, имеет набор четко организованных компонентов. Компоненты фильма объединены в *классы*. Впоследствии классы могут быть представлены *индивидуальными*, независимыми элементами, называемыми *объектами*. Объект — это *экземпляр* класса.

В классе содержится информация, которая передается в каждый создаваемый на его основе объект. Информация передается в виде *свойств*, то есть качеств, объекта и *методов* — действий, которые этот объект может выполнять. Во Flash имеется несколько predefined классов, создающих такие объекты, как Date и

Sound. Один из самых распространенных предопределенных объектов — Movie Clip (Клип) — имеет свойства и методы, например `_framesloaded` и `play()` соответственно.

С помощью ActionScript можно создавать и собственные объекты. Например, для создания класса `Dogs` (Собаки) со свойствами `_spotted` (Пятнистый) и `_scruffy` (Грязный) и методом `fetch` (Принести убитую дичь) можно использовать *конструктор* класса.

## Составные части сценария

Для того чтобы понять принцип работы сценариев, лучше всего рассмотреть сценарий, который связан с фильмом. Данный подход полезен в первую очередь по той причине, что можно наблюдать результаты выполнения. После того как вы поймете, что происходит в области действия, можно заглянуть «за кулисы» и посмотреть, какой сценарий заставляет фильм так себя вести или выполнять определенные действия.


 Рассмотрим пример, в котором стандартный указатель мыши заменяется созданным во Flash указателем в виде руки (файл `tapGlass.fla` в каталоге Chapter 14 на прилагаемом к книге компакт-диске). Когда выполняется щелчок мышью на стеклянном сосуде (что имитирует толчок), вода в сосуде покрывается рябью и раздается звук. Несмотря на то что анимация ряби состоит из 40 кадров, этот фильм помещается в один кадр (рис. 14.1). Эта анимация является символом типа Movie Clip, а стеклянная посуда — его экземпляром.



Рис. 14.1. Фильм `tapGlass`

В данном фильме сценарии, связанные с экземпляром клипа `hand`, задают ряд действий. В частности, они отвечают за связь между пользовательским указателем в виде руки, вызовом анимации сосуда и воспроизведением звука.

Откройте и запустите файл, чтобы увидеть, как он работает. Для просмотра сценариев, управляющих поведением указателя мыши, выберите экземпляр клипа `hand` и откройте панель Actions (нажатием клавиши F9). Текст сценария представлен на рис. 14.2.

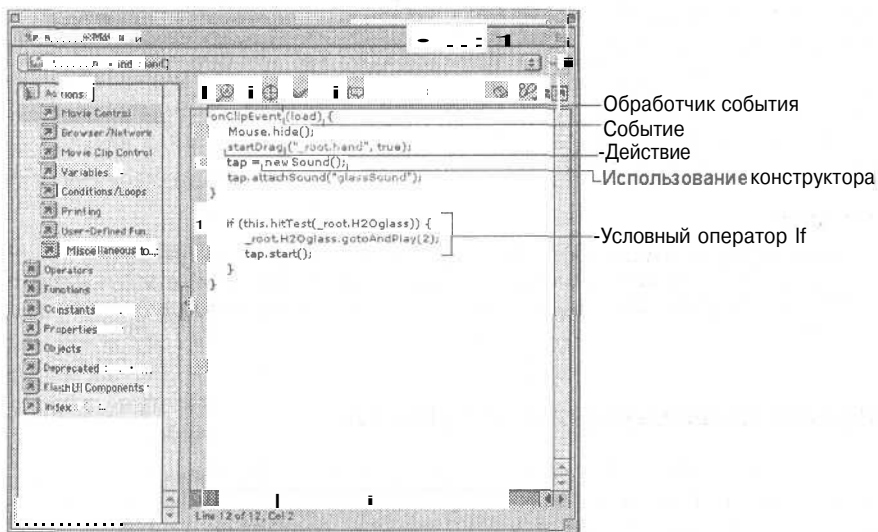


Рис. 14.2. Панель Actions со сценариями, связанными с экземпляром клипа `hand`

С действиями языка применяются два основных обработчика, а поскольку в этом фильме мы работаем с символом `Movie Clip`, обработчики связаны с событием `onClipEvent`. Действия, заданные в первом обработчике, выполняются один раз — при возникновении события, связанного с загрузкой клипа. Вот эти действия:

```
mouse.hide();
startDrag("_root.hand", true);
tap=new Sound();
tap.attachSound("glassSound");
```

Когда загружается клип, объект `Mouse` скрывает указатель мыши; затем действие `startDrag` делает символ `Movie Clip` перетягиваемым и заменяет указатель мыши. В следующих двух строках с помощью конструктора `new Sound()` активизируется объект `Sound` и с экземпляром этого объекта `tap` связывается звуковой файл `glassSound`.

Второй обработчик также является обработчиком событий клипа, но он выполняется только после получения клипом сообщения `mouseDown` (Щелчок мышью). Сообщение `mouseDown` обрабатывается с помощью условного оператора:

```
if(this.hitTest(_root.H2Oglass)){
 _root.H2Oglass.gotoAndPlay(2);
```

```
tap.start();
}
```

Оператор `if` определяет, находится ли изображение клипа, представляющего руку (`this`), над экземпляром клипа `H2Oglass`. Если они пересекаются, возвращается значение `True` и экземпляру `H2Oglass` дается команда проиграть второй кадр своей внутренней временной шкалы. После этого сценарий воспроизводит звук объекта `tap`, созданного при загрузке клипа в фильм. Условный оператор не содержит ветви `else`. Это означает, что если в момент нажатия кнопки мыши экземпляры двух клипов не пересекаются, то оператор `if` возвращает значение `false` и ничего не происходит. В случае отсутствия условного оператора, проверяющего, указывает ли рука на стакан, при каждом наступлении события `mouseDown` должны были бы вызываться анимация и звук. Изучите этот сценарий более внимательно и, изменяя различные параметры, попробуйте выяснить, какие типы поведения можно создать с его помощью.

#### ПРИМЕЧАНИЕ

В данном фильме имеется еще один сценарий. В кадре 1 клипа `glass` задано действие `stop()`. Это препятствует проигрыванию фильма до тех пор, пока не будет получено сообщение, указывающее на необходимость перехода ко второму кадру фильма и началу анимации.

## Процесс выполнения сценария

Не менее важным, чем знание синтаксических основ языка, является понимание того, как осуществляются в процессе выполнения фильма чтение и обработка сценария. Ведь причиной сбоя в его работе чаще являются ошибки в процессе выполнения, нежели неправильный синтаксис.

Как и в большинстве языков программирования, сценарии ActionScript выполняются последовательно, строка за строкой, от начала и до конца. Например, в следующем сценарии воспроизведение останавливается, значение свойства `_alpha` (Прозрачность) клипа `plane` устанавливается равным 50 %, после чего продолжается воспроизведение кадра 11.

```
stop();
setProperty("_root.plane",_alpha,50);
gotoAndPlay(11);
```

Эти действия выполняются точно в том порядке, в каком были написаны.

Сценарий может иметь несколько ветвлений. В ActionScript содержатся условные операторы, которые управляют ходом сценария. Ветвления в процессе реализации сценария обычно используются для осуществления проверки некоторого условия и выполнения в зависимости от полученного результата тех или иных действий.

Одним из самых распространенных условных операторов является оператор `if`. Ниже представлен его базовый вариант:

```
if(thisStatementIsTrue)
do thisStatement
}
```

Оператор `if` проверяет, истинно (`true`) или ложно (`false`) условие. Если оно истинно, то операторы в теле `if` будут выполнены. Оператор `if` может иметь часть `else` (рис. 14.3). В таком случае при возврате условным оператором значения `false` выполняются операторы, записанные после этого ключевого слова. Например:

```
if(thisStatementIsTrue){
 do thisStatement
}else{
 do aDifferentStatement
}
```

#### ПРИМЕЧАНИЕ

Если условие в операторе `if` не выполняется, а ключевое слово `else` отсутствует, то ни один оператор в теле условного оператора не будет выполнен.

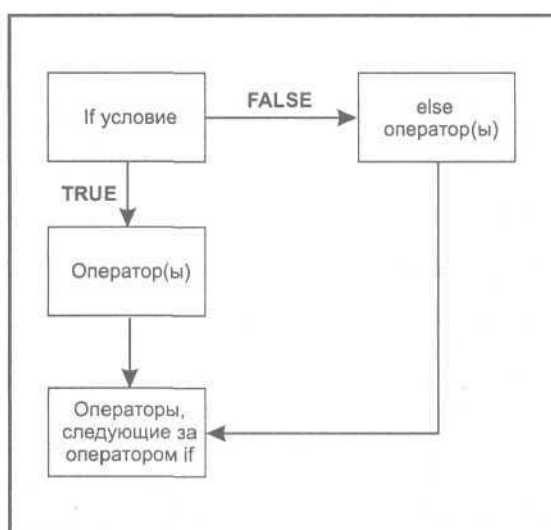


Рис. 14.3. Схема выполнения оператора `if...else`

Условные операторы удобно применять для осуществления определенных действий в зависимости от результатов проверки различных параметров фильма.

К условным операторам сценариев относятся также *циклы*. С их помощью реализуется повторение некоторых действий до тех пор, пока полученное значение не будет удовлетворять условию цикла (рис. 14.4). Например:

```
i=100;
while(i>10){
 i=i-1;
 setProperty("_root.p1ane",_alpha,i);
}
gotoAndPlay (11);
```

Этот фрагмент выполняется таким же образом, как и условный оператор `if...then`, то есть пошагово. Однако перед выполнением последнего оператора сценария

произойдет определенная задержка. Здесь используется цикл `while`, который приостанавливает дальнейшее выполнение сценария до тех пор, пока значение переменной `l` не будет уменьшено до заданной величины. Вместо того чтобы сразу установить для клипа значение свойства `_alpha` равным `0 %`, этот сценарий последовательно уменьшает его значение на единицу до тех пор, пока оно не станет равным `0`. После этого в сценарии выполняется последний оператор, который воспроизводит кадр 11. Циклы особенно удобны для реализации повторяющихся действий и обработки информации. За дополнительной информацией о циклах и синтаксисе языка можно обратиться к главе 15 и приложению Б.



Рис. 14.4. Схема выполнения цикла `while`

## Планирование сценариев

Каждому дизайнеру и аниматору известно, насколько важны такие составляющие процесса разработки сценария, как раскадровка, а особенно — планирование. Последняя стадия позволяет все продумать задолго до выпуска конечной версии. К тому же это еще и удобный случай перед началом работы над деталями взглянуть на проект в целом. Планирование дает возможность ответить на самый важный вопрос: а что же мы создаем?

Создать сценарий без планирования невозможно. Для того чтобы написать первую строку на ActionScript, вы должны знать назначение сценария и то, какие действия он будет выполнять для достижения заданной цели.

Другими словами, сначала необходимо решить, каким должен быть фильм, после чего можно будет перейти к следующему этапу и определить, какие компоненты понадобятся. Когда это станет понятно, выясните, как будут работать и взаимодействовать между собой отдельные компоненты. После этого определите, какую функцию в фильме должен выполнять ActionScript. Следующим шагом, конечно же, станет написание самих сценариев.

Лучшая стратегия при написании сценария — начать с простого. Прежде чем перейти к следующему элементу, добейтесь, чтобы работал текущий. Подобный

подход позволит вам локализовать любую проблему. Не забывайте почаще сохранять свои проекты, записывая различные их версии. В таком случае, если что-нибудь пойдет не так, вы сможете вернуться к последней версии и загрузить ее, а не начинать работу с нуля.

Если вы не знакомы с ActionScript, вам, скорее всего, придется провести немало времени, копаясь в справочниках по ActionScript в поисках компонентов, необходимых для создания сценариев. Причем вы неоднократно будете сталкиваться с терминами и понятиями, которые не нужны на данный момент, однако окажутся полезными в будущем. Следовательно, имеет смысл записывать их. Ведь когда такой термин понадобится, вы сможете обратиться к своим записям, что избавит от необходимости копаться во всей книге.

## Терминология ActionScript

ActionScript, как уже было сказано, является одним из наиболее мощных и гибких языков создания сценариев, который включает много компонентов. Ниже вы найдете ключевые термины ActionScript и их краткое описание. За дополнительной информацией обращайтесь к главе 16.

**Действие** — это оператор ActionScript, который указывает фильму или одному из его компонентов на необходимость выполнить какое-либо действие в процессе воспроизведения.

**Аргумент** (или параметр) — это контейнер, содержащий информацию и передающий ее действиям или функциям. Например, пользовательская функция `newUser()` имеет два аргумента — `userName` и `userID`:

```
function newUser(username, userID)
```

Эти аргументы сохраняются и используются затем в функции. Действия получают аргументы аналогичным образом:

```
gotoAndStop(5);
```

Для действия `gotoAndStop()` необходимо задать аргумент, определяющий, в каком месте временной шкалы требуется остановка.

**Класс** представляет категорию информации в фильме. Каждый объект принадлежит к какому-либо классу и является *экземпляром* этого класса. Чтобы определить новый объект, необходимо создать экземпляр объекта, основанный на классе. Это можно сделать с помощью функции конструктора.

**Константа** представляет собой неизменяемый элемент разных сценариев. Например, целое число является константой и может быть использовано для проверки значения выражения. Имя `SPACE` также является константой, поскольку всегда обозначает пробел.

**Конструктор** — это функция, используемая для создания объектов, которые основаны на классах. Функция, в свою очередь, имеет аргументы, которые определяют свойства класса.

**Тип данных** описывает вид информации, которую может хранить переменная или элемент ActionScript. В ActionScript поддерживаются следующие типы данных: строковый, числовой, булев (`true` или `false`), неопределенный, или нулевой

(отсутствие данных), объект, клип. Более полную информацию о типах данных можно получить в главе 15.

**Событие** происходит в процессе воспроизведения фильма. Оно порождается щелчком мыши, загрузкой клипа, нажатием клавиши и т. д. События используют для запуска сценариев и действий.

**Выражение** — это порция информации, которая продуцирует значение. Например:

```
on(release){
 gotoAndPlay(_currentframe+1);
}
```

Данный сценарий указывает фильму на необходимость перейти к кадру, номер которого равен значению выражения `_currentframe+1`.

**Функция** — это обработчик информации. Информация может передаваться в функцию в виде аргументов, а возвращаться из нее в виде значений. Функции незаменимы, если вам необходимо создать ActionScript-процедуру, которая при воспроизведении фильма будет повторяться неоднократно.

**Обработчик** предназначен для выполнения определенных действий в ответ на событие. В ActionScript существуют обработчики как для событий мыши; так и для событий клипа.

**Идентификатором** называется уникальное имя, присваиваемое функции, методу, объекту, свойству или переменной. Первым символом в имени идентификатора должна быть буква, знак доллара (\$) или подчеркивание (\_). Последующими символами наряду с перечисленными могут служить и числа.

**Экземпляр** является отдельный объект, принадлежащий классу. Например, экземпляр `today` может принадлежать классу `date`.

**Имя экземпляра** — это уникальное имя, используемое для ссылки на определенный экземпляр объекта `MovieClip` в рамках главной временной шкалы. Так, на протяжении всего фильма можно неоднократно использовать клип с именем `logoMovie`, присвоив каждому его экземпляру уникальное имя (например, `logo1`, `logo2` и т. д.). Каждый экземпляр в фильме должен быть указан с помощью своего уникального имени. Это позволит ActionScript управлять любым экземпляром клипа независимо от других его экземпляров.

**Ключевым словом** называют слово, имеющее в языке ActionScript определенное значение. Использование его в качестве имени переменной или, скажем, функции недопустимо. Чтобы просмотреть список ключевых слов, обратитесь к главе 15.

**Методом** называют действие, которое может быть выполнено объектом. Для пользовательских объектов можно создавать и пользовательские методы. В ActionScript каждый предопределенный объект (такой как `Sound` или `MovieClip`) имеет свои собственные методы. Список методов представлен в приложении Б.

**Объектом** называют экземпляр класса. ActionScript имеет несколько встроенных классов, экземплярами которых, в частности, являются объекты `Sound`, `Date` и `MovieClip`.



**Оператор** — это элемент, используемый для вычисления и сравнения: значений. Например, оператор / (косая черта) применяется для деления одного числа на другое.

**Свойство** характеризует объект или экземпляр объекта. Например, свойство `_x` определяет координату X клипа в области действия.

**Путь к объекту** предназначен для передачи в фильме информации через цепочку имен, переменных и объектов экземпляра `Movie Clip`. Например:

```
menuBar.item1.selected
```

Это путь к переменной `selected`, которая находится внутри клипа `item1`, расположенного внутри клипа `menuBar`. За дополнительными сведениями, касающимися иерархии клипов, вы можете обратиться к главе 16.

**Переменная** применяется для хранения информации. Данные из переменной можно извлекать при воспроизведении фильма и применять в сценариях.

## Использование панели Actions для создания сценариев

Как вы наверняка поняли в процессе изучения сценариев ActionScript, панель Actions используется для добавления в фильм действий. Данная панель предоставляет средства, которые позволяют выбирать, расставлять и упорядочивать элементы ActionScript, а при необходимости изменять их параметры.

До настоящего момента при написании сценариев вы использовали панель Actions только в режиме Normal. Это действительно самый удобный способ работы с таковой, особенно для новичков. Однако по мере приобретения опыта вам будет казаться, что было бы удобней иметь больше контроля над действиями, которые вы задаете. Полную свободу действий можно получить только при условии работы в режиме Expert.

## Написание сценариев в режиме Expert

Работа в режиме Expert подобна работе в режиме Normal, однако в первом случае разработчику предоставляется ряд дополнительных функций при работе с панелью Actions. Чтобы установить данную панель в режим Expert, вы должны выполнить следующие операции.

1. Выберите в меню Windows команду Actions, после чего появится панель Actions.
2. Выполните одно из следующих действий:
  - Щелкните мышью на всплывающем меню панели Actions, находящемся в правом верхнем углу экрана.
  - Выберите команду Expert Mode в меню Options панели Actions.
  - Нажмите комбинацию клавиш `Shift+Cmd+E` (Macintosh)/`Shift+Ctrl+E` (Windows).

Еще одно отличие между рассматриваемыми режимами заключается в том, что в режиме Expert отсутствует область параметров. В режиме Normal указанная область используется для ввода параметров выбранных вами действий, а в режиме Expert таковые вводятся вручную.

Возможно, вы помните из главы 12, что переключать режимы можно прямо в процессе написания сценариев. Это действительно так, однако не стоит забывать о следующих обстоятельствах.

- ▶ Если сценарий содержит ошибки, перейти из режима Normal в режим Expert можно, а обратно — нет. Вернуться в режим Normal можно лишь после устранения всех ошибок в сценарии.
- ▶ Переключение режимов может вызвать переформатирование, так что при переключении из одного режима в другой внимательно следите за тем, не появились ли ненужные изменения.
- ▶ Во Flash MX панель Actions предоставляет ряд новых возможностей, значительная часть которых доступна только в режиме Expert. Далее мы расскажем о том, как новые средства панели Actions могут ускорить вашу работу.

## Элементы панели Actions

Помимо возможности переключения режимов редактирования панель Actions предлагает ряд функций, облегчающих работу со сценариями. Вы можете производить поиск в сценариях и редактировать повторяющиеся элементы ActionScript посредством операций поиска и замены. Кроме того, существует возможность создавать сценарии вне среды Flash, а затем включать их в фильмы в качестве внешних сценариев. Кроме того, панель Actions позволяет выполнять печать сценариев и их проверку на наличие ошибок до запуска фильма на тестирование.

## Оформление текста сценария

**MX** Новые средства панели Actions позволяют управлять способом отображения сценариев, предоставляют возможность форматирования, изменения шрифта и цвета. Существуют также средства организации информации. Чтобы воспользоваться этими возможностями, панель необходимо запустить в режиме Expert. Это делается с помощью комбинации клавиш Shift+Cmd+E (Macintosh) /Shift+Ctrl+E (Windows) либо путем применения меню Options.

## Нумерация строк

Если строки пронумерованы, ориентироваться в больших по объему сценариях значительно легче. Чтобы включить/выключить нумерацию строк, выполните действия, перечисленные ниже.

1. Откройте панель Actions, выбрав команду Window ▶ Actions или нажав клавишу F9.
2. Откройте меню View Options и выполните одно из следующих действий:
  - ▷ Чтобы отобразить нумерацию строк, активизируйте команду View Line Numbers или нажмите клавиши Shift+Cmd+L (Macintosh)/Shift+Ctrl+L (Windows), после чего в меню View Options рядом с командой появится отметка.
  - ▷ При необходимости убрать нумерацию строк выберите команду View Line Numbers еще раз, и отметка, находящаяся до этого рядом с названием команды, исчезнет.

Когда нумерация строк включена, номера располагаются на вертикальной серой полосе, находящейся между областью элементов ActionScript и областью отображения сценариев (рис. 14.5). Если режим нумерации выключен, серая полоса пуста.

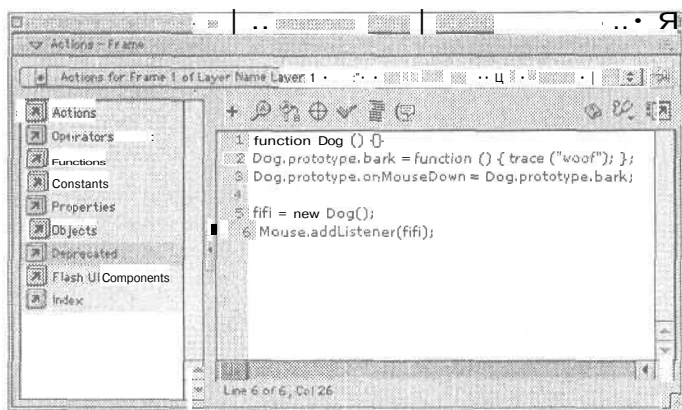


Рис. 14.5. Панель Actions с включенным режимом нумерации строк

## Автоматическое форматирование

Приведенные в нашей книге примеры на ActionScript легко читаются благодаря тому, что они были отформатированы. Однако на практике это делают далеко не всегда. Конечно, если при подготовке фильма, состоящего их множества сценариев, поджимают сроки, оформление текста сценария — последнее, что беспокоит его создателя. Однако если сценарий аккуратно отформатирован и хорошо организован, в нем легче найти синтаксические и логические ошибки.

**MX** Flash MX предлагает функцию автоматического форматирования, которая запускается посредством щелчка мыши или нажатия комбинации клавиш. Перечислим действия, которые необходимо выполнить для получения возможности применить к сценарию автоматическое форматирование.

1. Выберите объект или кадр, который содержит необходимый сценарий.
2. Откройте панель Actions, задав команду Window ▶ Actions или нажав F9.
3. Выполните одно из следующих действий:
  - > Нажмите клавиши Shift+Cmd+F (Macintosh)/Shift+Ctrl+F (Windows).
  - > В меню Options панели Actions задайте команду Auto Format.
  - > Нажмите кнопку автоматического форматирования (Auto Format).

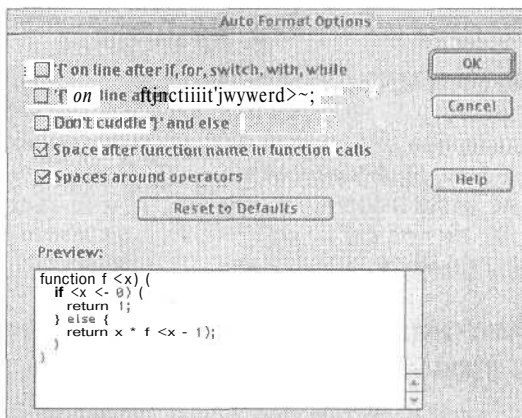
После этого сценарий будет автоматически переформатирован.

Изменить параметры автоматического форматирования можно следующим образом.

1. Щелчком мыши откройте меню панели Actions и выберите команду автоматического форматирования Auto Format Options.

2. В открывшемся окне Auto Format Options задайте нужные параметры автоматического форматирования. Чтобы увидеть, как конкретный параметр влияет на сценарий, обратитесь к области предварительного просмотра, находящейся в нижней части диалогового окна. После завершения ввода параметров нажмите кнопку ОК.

Заданные параметры будут применяться к текущему сценарию при использовании команды автоматического форматирования.



## Отображение текста сценария в панели Actions

Flash MX дает возможность оформлять текст сценария в панели Actions. Речь идет об изменении шрифта, его размера, цвета текста, об установлении отступов и т. д. Соответствующие параметры задаются на вкладке ActionScript Editor меню Preferences. Открыть указанную вкладку можно одним из двух способов.

Во-первых, можно вызвать команду Edit > Preferences и в появившемся на экране диалоговом окне выбрать нужную вкладку.

Во-вторых, посредством щелчка кнопкой мыши можно открыть меню панели Actions и активизировать в нем команду Preferences.

На вкладке ActionScript Editor вы можете установить следующие параметры.

- ▶ **Отступы.** Активируйте флажок Automatic Indentation, и Flash автоматически проставит отступы в сценарии. Размер отступа определяется значением в поле Tab Size.
- ▶ **Текст.** Для выбора типа и размера шрифта предназначена область Text. Рекомендуется устанавливать разборчивый экранный шрифт, что позволит избежать синтаксических ошибок при написании сценариев ActionScript.
- ▶ **Цвет.** У вас имеется возможность выделить отдельные синтаксические конструкции сценария различным цветом. Включите флажок Syntax Coloring, и каждый из указанных ниже элементов будет выделен индивидуальным цветом: Foreground (текст), Background (цвет окна), Keywords (ключевые слова), Comments (комментарии), Identifiers (идентификаторы) и Strings (строки). Для того чтобы выбрать нужный цвет, воспользуйтесь кнопкой с образцом текущего цвета, находящейся рядом с каждым элементом. Если же вы предпочитаете работать с текстами сценариев без цветовых выделений, отключите флажок Syntax Coloring и установите подходящие цвета для фона и переднего плана.

## Отслеживание версии

Во Flash MX можно создавать фильмы и для старых версий Flash Player. Однако предыдущие версии Flash не поддерживают всех возможностей ActionScript, характерных для версии MX. Следовательно, применять их в таких версиях нельзя. Панель Actions позволяет облегчить работу по обнаружению недопустимых для данной версии элементов сценария, поскольку они выделяются ярко-желтым цветом. Вы не сможете воспользоваться этими элементами, пока не перейдете к более поздней версии. О публикации фильмов можно прочесть в главе 28.

## Автоматический ввод кода с помощью подсказки

**MX** Во Flash MX при работе с панелью Actions можно воспользоваться *подсказками для кода*, которые представляют собой еще один способ написания сценариев. Хотя идея эта хороша, ее реализация немного неудачна.

### ПРИМЕЧАНИЕ

Подсказки для кода доступны только в режиме Expert.

Подсказки для кода предоставляют меню, из которого можно выбирать фрагменты сценария. Обычно подсказки применяются для вставки в сценарий обработчиков событий, действий и параметров действий. Причем вы сами указываете, каким образом должны появляться подсказки — автоматически или в результате вызова. В первом случае программа Flash сама определяет, когда нужна подсказка, и отображает ее в панели Actions. Вот как включаются автоматические подсказки для кода.

1. Выберите команду Edit ► Preferences и перейдите на вкладку ActionScript Editor или же выберите в меню панели Actions команду Preferences — в любом случае будет открыто диалоговое окно Preferences.
2. Чтобы включить автоматическое отображение подсказок, активизируйте флажок Code Hints. Воспользовавшись регулятором, установите время задержки перед появлением подсказки. Минимальная задержка составляет 0 (нет задержки), максимальная — 4 с.
3. Закройте окно Preferences, нажав кнопку ОК.

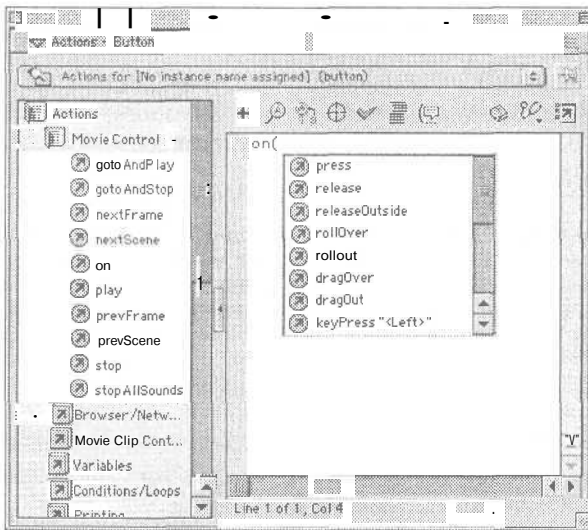
## Написание сценариев с использованием подсказки для кода

Как уже было сказано, подсказки для кода можно применять для трех элементов ActionScript: обработчиков событий, действий и параметров действий. Выполнив описанный ниже пример, вы научитесь с помощью подсказок писать сценарии с минимальным использованием клавиатуры. Предположим, вам нужно написать такой сценарий, связанный с кнопкой:

```
on(press){
 menu.gotoAndPlay(5);
}
```

Он дает указание экземпляру клипа мели начать воспроизведение фильма с кадра 5. Чтобы ввести этот сценарий с использованием подсказок, вы должны выполнить следующие действия.

1. Выберите в фильме символ Button.
2. Вызовите команду Window ► Actions или нажмите клавишу F9, чтобы открыть панель Actions.
3. В области отображения сценария этой панели введите с клавиатуры on(. Если автоматическая подсказка для кода включена, появится меню подсказки для программы обработки событий (рис. 14.6).



**Рис. 14.6.** Подсказка для кода на панели Actions

Если же подсказка выключена, активизировать ее можно одним из трех методов:

- > посредством комбинации клавиш Cmd/Ctrl+пробел;
  - ▷ с помощью кнопки Show Code Hint (Отображать подсказки для кода);
  - > путем вызова из меню панели Actions команды Show Code Hint (Отображать подсказки для кода).
4. Так как вы создаете обработчик события, подсказка для кода «знает», что вам нужен список всех обработчиков. Выбрать его из меню можно двумя способами:
    - ▷ с помощью клавиш со стрелками и клавиши Return/Enter;
    - ▷ посредством двойного щелчка на нужном событии кнопкой мыши.

Аналогичным образом вы можете выбрать из меню подсказки событие press, и оно будет вставлено в сценарий. Введите символы ) { и нажмите клавишу Return/Enter для перехода на новую строку. Незавершенный сценарий должен выглядеть так:

```
on(press){
```

5. В начале следующей строки введите `menu` — имя экземпляра клипа, для которого создается сценарий. Если вы пишете сценарий для клипа с другим именем экземпляра или другим путем к объекту, введите эту информацию в начале данной строки.
6. Когда имя экземпляра (или пути к объекту) будет указано, для ввода названия действия можно воспользоваться подсказкой для кода. Однако Flash не всегда может определить, перечень каких действий следует отображать. В данном случае вы пишете сценарий для управления клипом, а что если экземпляр `menu` является звуковым объектом или массивом? Не имея дополнительной информации, Flash не может это определить. Поэтому после слова `menu` введите `_тс`. Теперь сценарий должен выглядеть примерно так:

```
on(press){
 menu_тс
```

Суффикс класса `_тс` указывает на то, что экземпляр `menu` является клипом. Полный список суффиксов классов приведен в табл. 14.1.

---

#### ПРИМЕЧАНИЕ

Использование суффиксов типа `_тс` — это одна из особенностей функционирования Flash MX. Заменяв в сценарии имя `instanceName` именем `instanceName_тс`, вы укажете программе на необходимость искать абсолютно другой экземпляр клипа. Суффикс `_тс` интерпретируется как часть полного имени экземпляра. Если при написании сценариев вы планируете применять подсказки, понадобится иной метод. Лучшим способом является использование соответствующих суффиксов в именах экземпляров.

---

7. После суффикса класса `_тс` поставьте точку. Если функция использования подсказок включена, появится меню. В противном случае примените один из методов ее активизации, описанных в пункте 3.
8. Выберите из меню подсказки действие `gotoAndPlay()` (как это сделать, рассказывается выше, в пункте 4). Незавершенный сценарий должен выглядеть так:

```
on(press){
 menu_тс.gotoAndPlay(
```

9. На данном этапе вас ждет сюрприз. После ввода с помощью подсказки действия `gotoAndPlay()` вам придется опять выбирать подсказку, однако эта подсказка будет отличаться от тех, с которыми приходилось встречаться ранее. Она называется всплывающей и появляется в желтой рамке. Такие подсказки используются для ввода параметров действий. Действию `gotoAndPlay()` необходим один параметр — номер кадра. После появления желтой всплывающей подсказки можно ввести значение, которое будет вставлено в качестве параметра для данного действия. Введите сначала значение 5, а затем закрывающую скобку. Это приведет к исчезновению подсказки для кода. Завершается такая строка точкой с запятой.

---

#### ПРИМЕЧАНИЕ

Для некоторых действий требуется задать более одного параметра. Чтобы это сделать с помощью подсказки, достаточно ввести параметры в нужном порядке, разделив их запятыми. Однако не надейтесь, что всплывающие подсказки Flash укажут вам правильное число или порядок параметров. Синтаксис параметров описан в приложении Б.

---

10. Нажмите клавишу Return/Enter, чтобы перейти на новую строку, и введите закрывающую фигурную скобку, указывающую на конец сценария. Сценарий будет иметь такой вид:

```
on(press){
 menu_mc.gotoAndPlay(5);
}
```

#### ПРИМЕЧАНИЕ

Не забудьте изменить имена всех своих экземпляров в панели Property — лишь при этом условии будет производиться ссылка на существующие имена экземпляров при указании суффиксов подсказки кода. В данном примере вам необходимо имя экземпляра `menu` изменить на `menu_mc`.

Подсказки для кода позволяют при написании сценариев заметно сократить количество текста, вводимого при помощи клавиатуры. Применять их или нет — решать вам. Одним из достоинств Flash MX является полностью настраиваемый интерфейс, позволяющий использовать понравившиеся возможности программы и отказаться от тех ее средств, которые кажутся неудобными.

**Таблица 14.1** Суффиксы классов, которые используются в подсказках для кода

| Суффикс                  | Класс объектов                     | Суффикс                 | Класс объектов                               |
|--------------------------|------------------------------------|-------------------------|----------------------------------------------|
| <code>_array</code>      | Array (массив)                     | <code>_so</code>        | SharedObject (совместно используемый объект) |
| <code>_btn</code>        | Button (клавиша)                   | <code>_sound</code>     | Sound (звук)                                 |
| <code>_camera</code>     | Camera (камера)                    | <code>_str</code>       | String (строка)                              |
| <code>_color</code>      | Color (цвет)                       | <code>_stream</code>    | NetStream (сетевой поток)                    |
| <code>_connection</code> | NetConnection (сетевое соединение) | <code>_txt</code>       | TextField (текстовое поле)                   |
| <code>_date</code>       | Date (дата)                        | <code>_video</code>     | Video (видео)                                |
| <code>_fmt</code>        | TextFormat (формат текста)         | <code>_xml</code>       | XML                                          |
| <code>_mc</code>         | MovieClip (клип)                   | <code>_xmlsocket</code> | XMLSocket (XML гнездо)                       |
| <code>_mic</code>        | Microphone (микрофон)              |                         |                                              |

Итак, чтобы иметь возможность применять подсказки кода, введите соответствующий суффикс после имени объекта.

## Навигация, поиск и замена в сценариях ActionScript

Для перехода в определенное место сценария ActionScript удобно пользоваться следующими командами всплывающего меню, находящегося в верхнем левом углу панели Actions:

- ▶ Go To Line (Cmd+G (Macintosh)/Ctrl+G (Windows)) — обеспечивает переход к строке сценария с указанным номером.
- ▶ Find (Cmd/Ctrl+F) — предназначена для поиска конкретного слова или фразы. Для выполнения поиска можно нажать кнопку Find. С помощью команды Find



Again (Cmd+G (Macintosh) или F3 (Windows)) вы можете обнаружить следующее вхождение данного текста.

- ▶ Replace (Cmd/Ctrl+Shift+H (Macintosh)/Cmd/Ctrl+H (Windows)) обеспечивает поиск заданного текста и замену его другим. Также для выполнения этих действий предназначена кнопка Replace.

В режиме Expert операции поиска и замены осуществляются по всему сценарию. В режиме Normal это можно сделать лишь в области параметров.

Функции поиска и замены работают только в текущем окне. Чтобы выполнить поиск в пределах всего фильма, воспользуйтесь панелью Movie Explorer. За дополнительной информацией по этой теме обратитесь к главе 7.

## Работа с данными в панели Actions

Существует несколько методов перемещения данных в панель Actions и из нее. Всплывающее меню, находящееся в правом верхнем углу панели, позволяет выполнить ряд задач.

**Импорт текста.** Выберите команду Import From File или нажмите комбинацию клавиш Shift+Cmd+I (Macintosh)/Shift+Ctrl+I (Windows), а затем укажите файл сценария ActionScript. Эта операция применяется в том случае, если вы предпочитаете писать сценарии в текстовом редакторе или редакторе сценариев и хотите интегрировать их прямо в фильм.

**Экспорт сценария.** Выберите команду Export As File или нажмите клавиши Shift+Cmd+X (Macintosh)/Shift+Ctrl+X (Windows), присвойте файлу имя, нажмите кнопку Save (Сохранить), и Flash экспортирует сценарий в текстовый файл.

**Печать.** Выберите команду Print, установите параметры печати и нажмите кнопку Print.

### СОВЕТ

Когда печать сценария осуществляется из панели Actions, Flash не предоставляет никакой информации о файле, содержащем данный сценарий, или о расположении сценария в фильме. Если вы планируете распечатать несколько сценариев, полезно создать комментарии, указывающие, к какой части фильма относится конкретный сценарий. Дополнительные сведения по этой теме вы найдете в главе 15.

## Проверка синтаксиса и поиск ошибок

Одним из преимуществ режима Expert считается возможность полного управления вводом в текстовом окне. Правда, в этом случае не выполняется автоматическая проверка текста сценария на наличие ошибок в процессе ввода (в режиме Normal такая проверка производится). До запуска фильма проверку синтаксиса можно выполнить тремя способами:

- ▶ нажатием кнопки Check Syntax (Проверка синтаксиса), находящейся в верхней части панели Actions (кнопка с изображением галочки);
- ▶ путем выбора в меню Options, которое находится в верхнем правом углу панели Actions, команды Check Syntax;
- ▶ нажатием клавиш Cmd/Ctrl+T при активной панели Actions.

Если в сценарии имеются ошибки, вы будете предупреждены об этом. Информация об ошибках выводится в окне Output, достаточно подробно описанном в главе 18.

## Работа с внешним редактором сценариев

Несмотря на богатство возможностей, которые предоставляет в режиме Эксперт панель Actions, вы, возможно, сочтете необходимым воспользоваться внешним редактором и написать часть сценария ActionScript вне среды Flash. Чтобы присоединить такой фрагмент ко всему сценарию, нужно в том месте, где он должен находиться, вставить следующую строку:

```
#include "имя_сценария.as"
```

Здесь имя\_сценария.as — это имя текстового файла, содержащего необходимый код для кадра или объекта, к которому относится действие #include. Это действие импортирует строки ActionScript из внешнего файла с расширением имени .as и запускает фильм так, как если бы он был написан прямо в панели Actions. Вы сможете воспользоваться этим методом для импорта ActionScript из внешнего файла, выполнив перечисленные далее действия.

1. Создайте сценарий во внешнем редакторе и сохраните его в файле с расширением .as.
2. Перетащите действие include из области элементов ActionScript в область отображения сценария или введите в эту область строку #include.
3. В области отображения параметров (режим Normal) или в скобках после действия #include (режим Expert) введите путь ко внешнему файлу сценария. Для разделения элементов пути воспользуйтесь символом косой черты (/). К примеру, если файлы movie.Ла и extScript.as находятся в одном каталоге, путь будет выглядеть так: extScript.as. Но если файлы сценария находятся во вложенном каталоге с именем Actions, путь следует указать по-другому: actions/extScript.as.

После публикации фильма, в котором используются внешние сценарии, таковые становятся неотъемлемой частью фильма. Чтобы внести какие-либо изменения в сценарий, нужно изменить код ActionScript в AS-файле и опубликовать фильм повторно.

## Присоединение сложных сценариев ActionScript

Методики присоединения сложных и простых сценариев к объектам и кадрам не очень различаются. Однако при работе с несколькими сценариями важно знать, какое влияние оказывают они друг на друга в процессе воспроизведения фильма. Ценным помощником при этом может стать панель Movie Explorer, которая позволяет увидеть иерархию элементов фильма и проследить за тем, какие сценарии к каким элементам присоединены. Более полная информация о Movie Explorer содержится в главе 7.

Главное различие в принципах работы с простыми и сложными сценариями связано с тем, что последние в большинстве случаев создают с помощью панели

Actions в режиме Expert. Так делают по той простой причине, что данный режим предоставляет большую гибкость и свободу действий.

При работе со сценариями в режиме Normal необходимо выполнить ряд операций по перетаскиванию действий из одной области окна в другую. Сделать это совершенно не трудно, если речь идет о нескольких строках кода, но для работы со сложными сценариями данный метод не совсем подходит. Кроме того, при редактировании сценария в режиме Normal все изменения необходимо вносить в области отображения параметров. Это означает, что доступ к сценарию возможен только через элементы данной области. Конечно, для большинства задач этого достаточно, однако в режиме Expert указанное ограничение отсутствует и каждый элемент сценария доступен для изменения.

Освоив режим Expert, вы наверняка согласитесь, что он обеспечивает один из самых удобных методов присоединения сценариев к фильму.

## Присоединение сценариев к кнопкам и клипам в режиме Expert

Сценарии, как вы уже знаете, присоединяют к таким объектам Flash, как кнопки и клипы. В результате события, наступающие при щелчках мышью, одновременном нажатии нескольких кнопок или, скажем, перетаскивании, приводят к выполнению определенных действий. Для клипов с помощью сценариев можно определить такие события, как загрузка в память или покадровое воспроизведение. К тому же объекты указанных типов могут контролировать и записывать данные о том, какие клавиши нажаты пользователем при воспроизведении фильма.

Перечислим операции, которые необходимо выполнить для того, чтобы присоединить к объекту действие в режиме Expert.

1. Щелкните мышью на кнопке или объекте клипа, а затем выберите команду Window ▸ Actions или нажмите клавишу F9.
2. Выполните щелчок на кнопке View Options и выберите команду Expert Mode или нажмите Shift+Cmd+E (Macintosh)/Shift+Ctrl+E (Windows).
3. Задайте действие одним из следующих методов:
  - ▷ выберите действие из соответствующей категории области элементов ActionScript и произведите двойной щелчок на его имени. Кроме того, это действие можно перетащить в область отображения сценариев;
  - ▷ воспользуйтесь кнопкой добавления действия (+) и выберите нужное действие во всплывающем меню;
  - ▷ введите имя действия ActionScript в область отображения сценариев с клавиатуры.

### ПРИМЕЧАНИЕ

В режиме Normal обработчики событий при перетаскивании события из области элементов ActionScript вставляются в сценарий автоматически. В режиме Expert этого не происходит, поэтому вы должны сами назначить событие trigger, которое запустит сценарий на выполнение. Информация об обработчиках событий ActionScript содержится в главе 12.

4. Далее вам необходимо проверить синтаксис сценария. Для этой цели можно воспользоваться любым из трех следующих методов: выбрать команду Syntax Check во всплывающем меню панели Actions, нажать кнопку проверки синтаксиса либо же нажать клавиши `Cmd/Ctrl+T`.
5. Если в сценарии нет ошибок, выберите команду `Control ▶ Test Movie`, и вы сможете просмотреть фильм. Чтобы присоединить другие сценарии, для перехода к какому-либо объекту воспользуйтесь раскрывающимся списком сценариев и повторите действия, указанные в пунктах 3 и 4. Подобным образом, не выходя из панели Actions, можно написать несколько сценариев. Чтобы в панели Actions всегда видеть сценарий объекта, нажмите кнопку Script Pin. Даже если вы перейдете к другим объектам или кадрам фильма, «приколотый» сценарий не исчезнет.

## Присоединение действий к кадрам в режиме Expert

Действия кадров практически не отличаются от действий объектов. Наиболее существенное различие заключается в том, что для первых не нужен обработчик. По умолчанию при воспроизведении кадра со сценарием последний запускается на выполнение. При необходимости присоединить к кадру сценарий в режиме Expert, выполните следующие операции:

1. Выберите ключевой кадр на временной шкале и вызовите команду `Window ▶ Actions` (либо нажмите клавишу `F9`). Если же ключевого кадра не существует, для его вставки выберите нужную позицию на временной шкале, а затем -- команду `Insert ▶ Keyframe`.
2. Перейдите в меню `View Options` и вызовите команду `Expert Mode` или нажмите клавиши `Shift+Cmd+E (Macintosh)/Shift+Ctrl+E (Windows)`.
3. Задайте действие одним из методов, перечисленных в предыдущем разделе.
4. Проверьте синтаксис сценария. Сделать это можно путем выбора команды `Syntax Check` во всплывающем меню панели Actions, с помощью кнопки проверки синтаксиса или же нажатием клавиш `Cmd/Ctrl+T`.
5. Если в сценарии нет ошибок, выберите команду `Control ▶ Test Movie`, чтобы просмотреть фильм. Для того чтобы присоединить другие сценарии, для перехода к иному объекту воспользуйтесь раскрывающимся списком сценариев и повторите действия, указанные в пунктах 3 и 4. Таким образом, не выходя из панели Actions, вы можете написать несколько сценариев. Чтобы всегда видеть сценарий объекта, нажмите кнопку Script Pin.

## Разработки

Благодаря поддержке языка ActionScript программа Flash из простого средства создания анимации превратилась в мощнейший инструмент для разработки полноценных мультимедийных Web-узлов, интерактивных форм, мультсериалов и игр.

Прекрасным примером, демонстрирующим все возможности Flash, может служить узел [www.sarbakan.com/sybex/snooz](http://www.sarbakan.com/sybex/snooz), на котором представлена интереснейшая

интерактивная игра Goodnight Mr. Snoozleberg, разработанная компанией Sargaban Of Quebec. Посетив данный узел, вы сможете убедиться, насколько удачным может быть комбинирование возможностей векторной анимации и программирования.

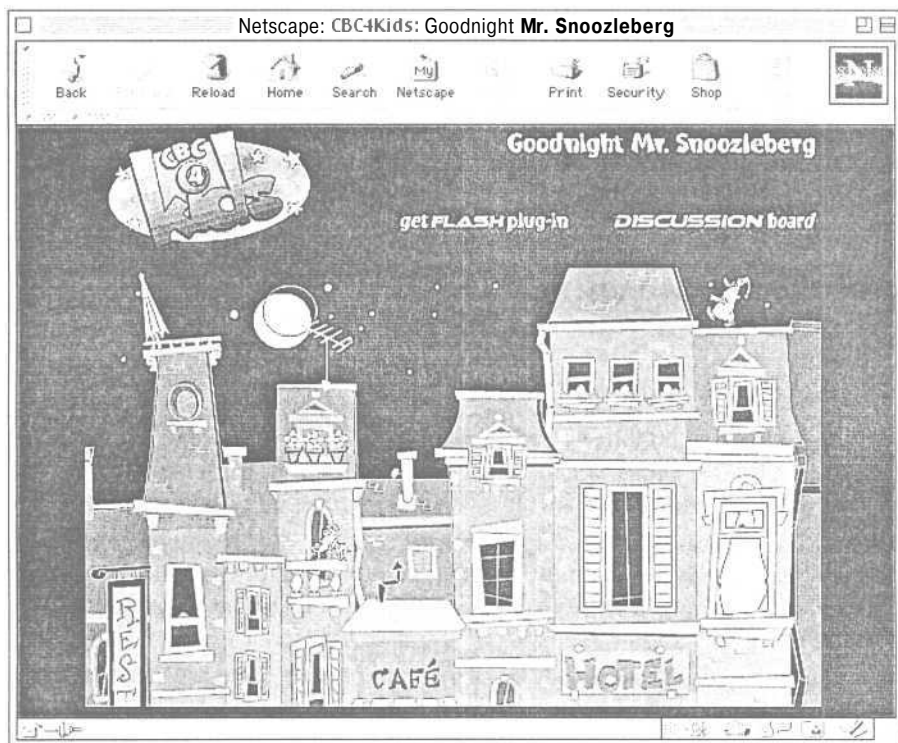


Рис. 14.7. Goodnight Mr. Snoozleberg — игра, демонстрирующая возможности Flash

## Резюме

В настоящей главе изложены наиболее важные сведения о языке ActionScript, описаны компоненты, из которых состоит сценарий, и принципы их совместной работы, перечислены основные возможности программы, предоставляемые в режиме Expert. Вооружившись этими знаниями, можно приступить к созданию сложных сценариев на языке ActionScript.

## Глава 15

# Написание сценариев ActionScript

В предыдущей главе были раскрыты базовые понятия ActionScript и описаны приемы использования этого языка. Настоящая глава поможет вам научиться писать более сложные сценарии для Flash-фильмов.

Мы уже продемонстрировали возможности сценариев ActionScript. С их помощью простые мультипликационные фильмы преобразовываются в сложные интерактивные приложения. Чего у нас еще нет, так это структуры, позволяющей создавать свои собственные сценарии. Кроме того, вы должны изучить правила языка. Без знания этих правил невозможно использовать ActionScript для правильного построения структуры сценария.

В этой главе рассматриваются такие вопросы:

- ▶ синтаксис языка ActionScript;
- ▶ наиболее важные элементы языка ActionScript;
- ▶ комментарии;
- ▶ типы данных;
- ▶ выполнение математических вычислений в ActionScript;
- ▶ действия ActionScript;
- ▶ задание условий в сценарии;
- ▶ предопределенные объекты.

## Синтаксис языка ActionScript

Подобно разговорным языкам, языки программирования (в том числе и ActionScript) имеют определенный набор правил использования слов языка, называемый синтаксис. Эти правила необходимо понять и запомнить, чтобы потом правильно пользоваться ими при написании сценариев.

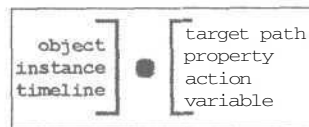
Знание всех терминов ActionScript, описанных в этой книге, несомненно полезно, если вы хотите стать опытным разработчиком сценариев. Но особенно важно знать, как используются правила совместно с элементами языка. Разобравшись во всем этом, вы сможете легко создавать сценарии любого уровня сложности и связывать их с Flash-фильмами.

## Пунктуация в ActionScript

В ActionScript применяется несколько знаков пунктуации. Вы можете использовать эти знаки, чтобы выполнить команду или указать путь к фильму или переменной. Они также должны помочь при написании сценариев, обеспечив в последних ввод описаний команд (комментарии). Основное назначение знаков пунктуации состоит в упорядочении сценариев и организации их содержимого так, чтобы программа Flash могла понять, как эти сценарии должны быть обработаны.

## Точечный синтаксис

Точечный синтаксис в ActionScript был введен во Flash 5. Он делает язык ActionScript подобным JavaScript. Если ранее вы работали с JavaScript, то в процессе изучения ActionScript встретите много знакомого для себя.



На рисунке видно, что слева от точки указывается объект или ссылка на временную шкалу клипа. Справа от точки может быть свойство, путь, переменная или действие, которое относится к элементу в левой части. Вот три примера:

```
myClip._visible=0;
menuBar.menu1.item3;
_root.gotoAndPlay(5);
```

В первом примере клип с именем `myClip` сделан невидимым путем установки свойства `_visible` равным `0`. Во втором примере указан путь к переменной `item3` клипа `menu1`, вложенного в клип с именем `menuBar`. В третьем примере ссылка `_root` используется для того, чтобы головка воспроизведения могла перейти к пятому кадру главной временной шкалы и начать воспроизведение фильма с этого места. В каждом примере слева от точки содержится имя объекта или ссылка на него, в то время как справа указывается некоторая команда, определяющая объект или манипулирующая им.

### ПРИМЕЧАНИЕ

Точка известна также как точечный оператор, поскольку она может быть использована для выполнения команд и изменения значений свойств. Более подробную информацию об этом вы можете найти в разделе «Операторы» далее в этой главе.

## Синтаксис с использованием символа /

Синтаксис с использованием символа `/` применялся во Flash 4, но впоследствии был заменен точечным. Хотя первый все еще поддерживается во Flash MX, его использование не приветствуется. Этот синтаксис применялся во Flash 4 прежде всего для того, чтобы разбить на фрагменты путь к объекту. Например, путь к клипу `leg`, являющемуся элементом клина `spider`, который, в свою очередь, вложен в клип `body`, указывался так:

```
"/spider/body/leg";
```

Теперь, применяя точечный синтаксис, тот же самый путь можно указать следующим образом:

```
spider.body.leg;
```

Синтаксис с использованием символа `/` применялся также при обращении к переменным, принадлежащим другим фильмам или клипам, а также при наличии в сценарии действия `tellTarget()`. Теперь, когда действие `tellTarget()` применять не рекомендуется, лучше использовать точечный синтаксис и действие `with()`.

## Комментарии

Хороший стиль программирования подразумевает включение в код сценариев примечаний. Они могут быть полезны тому, кто будет редактировать ваши сценарии. Они пригодятся также вам самому, если вы будете вынуждены отложить выполнение проекта на некоторое время. При возобновлении работы примечания напомнят вам о назначении различных частей сценария.

Примечания в сценарий на языке ActionScript необходимо вставлять как комментарии. Чтобы добавить комментарии в режиме Normal, дважды щелкните на действии comment или просто перетащите его из подкатегории Actions ▸ Miscellaneous Actions и введите примечания в текстовом поле Comment области отображения параметров. Чтобы добавить комментарии в режиме Expert, введите оператор // и добавьте примечания после него. Например:

```
//Проверка того, все ли кадры загружены
if (_framesLoaded>=_totalFrames) {
// Если да (true), то начать воспроизведение анимации с кадра 6
gotoAndPlay (6);
} else {
// Если нет (false), то воспроизводить анимацию, начиная с кадра 1
gotoAndPlay (1);
}
```

В ActionScript все данные в строке, которые следуют справа от оператора //, будут игнорироваться. Комментарии также позволяют отключать целые блоки сценария. Если есть одна или более строк кода, выполнение которых необходимо заблокировать, вы можете преобразовать их в комментарий и выполнить сценарий без этих строк. В сценарии все преобразованные в комментарий строки будут игнорироваться. Несколько больше об этой технике можно узнать, ознакомившись с материалами главы 18.

### СОВЕТ

Чтобы вставить в сценарий длинные описания или большие текстовые блоки, вы можете использовать оператор многострочных комментариев (/\*). Этот удобный инструмент облегчает написание длинных сообщений и пояснений в теле сценария. Просто начните описание открывающим оператором (/\*) и закончите закрывающим (\*/). Все слова и выражения, находящиеся между этими операторами, будут проигнорированы приложением Flash Player.

По умолчанию комментарии выделяются в области отображения сценариев шрифтом серого цвета, что позволяет легко отличить их от других блоков сценария. Для изменения этих установок выберите команду Edit ▸ Preferences или воспользуйтесь командой Preferences меню Options панели Actions. Перейдите к вкладке ActionScript Editor и выберите **новый** цвет шрифта в палитре цветов Comments.

## Фигурные скобки

Сценарий в ActionScript состоит из блоков; инструкции, которые должны быть выполнены в пределах блока, помещаются в фигурные скобки ({}). В следующем сценарии все выражения, заключенные в фигурные скобки, будут выполнены после щелчка мышью на кнопке, с которой связан этот сценарий.

```
on(press){
with(fishClip){
```



```

 gotoAndStop(50);
 }
}

```

В этом примере фигурные скобки разбивают сценарий на две части. Обработчик события `on(press)` выполняет действие `with()`, а действие `gotoAndStop()` указывает головке воспроизведения перейти к кадру 50 клипа `fishClip`; после чего воспроизведение фильма прекращается.

## Круглые скобки

Круглые скобки используются в ActionScript для указания аргументов функций или установки порядка выполнения математических операций. Например, действие `setProperty()` имеет три аргумента: имя клипа, свойство и значение (см. приложение Б). Круглые скобки используются, чтобы перечислить аргументы действия: `setProperty("testClip", _alpha, 50)`;

Этот сценарий устанавливает для клипа `testClip` значение свойства `_alpha` (прозрачность) равным 50 %.

Вы можете также использовать круглые скобки, чтобы изменить порядок выполнения математических операций в выражении. Результат выполнения выражения  $2 + 3 \times 4$  равен 14, тогда как для выражения  $(2 + 3) \times 4$  он равен 20, потому что круглые скобки требуют сначала произвести сложение.

## Точка с запятой

В ActionScript точка с запятой (;) используется, чтобы указать конец инструкции. Например:

```

on(release){
 introClip._visible=0;
}

```

Здесь строка со свойством `_visible`, которая делает экземпляр `introClip` невидимым, оканчивается точкой с запятой. Точка с запятой в конце инструкции не является обязательной, и если ее не поставить, после компиляции не будет получено сообщение об ошибке. Однако все-таки желательно следовать правилам синтаксиса и всегда завершать инструкции точкой с запятой.

Точка с запятой также используются для того, чтобы отделить один от другого аргументы в структуре цикла `for`, но об этом мы поговорим позже (в этой же главе).

## Другие правила синтаксиса ActionScript

Кроме следования установленным правилам языка ActionScript, необходимо еще хорошо знать и некоторые термины. Некоторые из них являются зарезервированными словами языка. Также существуют термины, которые должны быть написаны с использованием символов верхнего или нижнего регистра, чтобы Flash знал, как правильно их интерпретировать.

## Константы

Константа — это элемент языка, значение которого не изменяется. В коде сценария имена констант записываются буквами верхнего регистра. Константы содержатся в трех объектах ActionScript: `Key`, `Math` и `Number`.

Следующий сценарий использует объект `Key`, чтобы проверить, была ли нажата клавиша пробела:

```
if(Key.isdown(Key.SPACE)){
 laser.shoot(1);
}
```

Свойство `Key.SPACE` объекта `Key` является константой (ее значение равно 32). После нажатия клавиши пробела выполняются инструкции, расположенные после действия `if`.

В следующем примере объект `Math` использует константу `PI`, чтобы определить площадь круга:

```
area=Math.PI*(radius*radius);
```

А в этом примере константа `MAX_VALUE` применяется, чтобы присвоить переменной `duration` максимально возможное в ActionScript значение числа ( $1.79E + 308$ ):

```
duration=Number.MAX_VALUE;
```

## Ключевые слова

Ключевые слова — термины ActionScript, которые являются зарезервированными словами и имеют особое назначение. Эти термины используются только в определенном контексте, их нельзя употреблять в качестве имен переменных, функций, объектов или ссылок. Список ключевых слов ActionScript приведен в табл. 15.1.

**Таблица 15.1.** Ключевые слова языка ActionScript

|                       |                       |                         |                     |
|-----------------------|-----------------------|-------------------------|---------------------|
| <code>break</code>    | <code>else</code>     | <code>instanceof</code> | <code>typeof</code> |
| <code>case</code>     | <code>for</code>      | <code>new</code>        | <code>var</code>    |
| <code>continue</code> | <code>function</code> | <code>return</code>     | <code>void</code>   |
| <code>default</code>  | <code>if</code>       | <code>switch</code>     | <code>while</code>  |
| <code>delete</code>   | <code>in</code>       | <code>this</code>       | <code>with</code>   |
| <code>do</code>       |                       |                         |                     |

## Чувствительность к регистру в ActionScript

Как правило, ActionScript «прощает» ошибочное использование символов верхнего или нижнего регистра. Если вы уже создавали сценарии в режиме Expert, то, наверное, обратили внимание на следующее: цвет шрифта имени действия изменяется, если используются символы регистра, отличного от указанного в документации по ActionScript. Например, действие `stop` (записано символами нижнего регистра) обычно отображается в области отображения сценариев синим цветом. Если записать его как `Stop()` (обратите внимание на прописную букву S), цвет изменится на черный. В большинстве случаев Flash может безошибочно интерпретировать сценарий. Однако, чтобы полностью застраховаться от них, лучше всегда следовать установленным правилам: набирать прописными буквами все выражения, которые следует указывать символами верхнего регистра, и строчными буквами все выражения, которые следует указывать символами нижнего регистра.

## Элементы языка ActionScript

Итак, вам *уже* знакомы соглашения, принятые в ActionScript, и вы знаете, как различные элементы языка используются для создания сценария. Теперь пришло время более подробно рассмотреть отдельные элементы. Большинство из них напоминают слова английского языка, которые приблизительно соответствуют назначению элементов в ActionScript, что само по себе упрощает их запоминание. Самое важное при изучении ActionScript — разобраться в том, какую роль тот или иной элемент играет в языке. Впоследствии принципы работы этих элементов станут вам более понятными и вы научитесь правильно их использовать при решении той или иной проблемы.

### Типы данных

В языке ActionScript данные могут быть представлены одним из шести типов: строка, число, булев, объект, клип и неопределенный. Рассмотрим каждый тип в отдельности.

#### Строковый тип данных

Строки — последовательности, состоящие из буквенных и цифровых символов, а также знаков пунктуации. Строковые значения в коде сценария заключаются в кавычки, двойные ("" ) или одинарные ( ' ' ). Поскольку строки могут содержать буквенные символы, они чувствительны к регистру, так что, например, строки one и One различны. В следующем примере Joe — строковое значение переменной myName:

```
myName="Joe";
```

Строки могут быть объединены (это называется — конкатенация строк). Например:

```
fullName=myName+" Smith";
```

Оператор сложения (+) используется, чтобы объединить строковые значения. Обратите внимание на то, что в предыдущем примере строка Smith содержит пробел перед символом S. Таким образом предотвращается слияние двух строковых значений при объединении. Результат **объединения** строк следующий:

```
fullName="Joe Smith";
```

Строки можно также упорядочить в алфавитном порядке, используя операторы сравнения: <, >, = < и > =. Во Flash применяется кодировка символов Latin1. При использовании операторов сравнения для упорядочивания строк в алфавитном порядке, помните, что код символа z имеет самое большое значение среди кодов, а код символа A — самое маленькое (см. приложение Б). Обратите внимание на следующее выражение:

```
"alligator"<"zebra"=true
"alligator">"Zebra"=true
```

Из этого примера видно, что коды символов нижнего регистра имеют более высокие значения, чем коды символов верхнего регистра. Для получения более подробной информации относительно операторов сравнения обратитесь к материалам раздела «Операторы» этой главы.

## Числовой тип данных

В ActionScript со значениями этого типа данных можно производить различные математические операции: сложение (+), вычитание (-), умножение (\*), деление (/), деление по модулю (%), инкрементирование (++) и декрементирование (--). Также при работе с числами вы можете использовать предопределенный объект `Math`. Более подробную информацию о данных этого типа вы найдете в разделе «Операторы» этой главы и в приложении Б.

## Булев тип данных

Переменные булева (логического) типа могут содержать только одну из двух констант: `true` или `false`. Булевы переменные используются в условных инструкциях, которые в зависимости от результата проверки условия выполняют соответствующие строки кода. Следующая конструкция проверяет, действительно ли все кадры фильма были загружены в память:

```
_framesloaded=loaded;
if(loaded==true){
introMovie.gotoAndPlay(2);
}
```

В большинстве условных конструкций булево значение подразумевается по умолчанию и не устанавливается явно в действии `if`.

## Объекты

Объект -- структура данных с определенными свойствами. Каждое из этих свойств имеет имя и значение, к которым можно обратиться во Flash-фильме. Объектный тип данных позволяет управлять свойствами, назначать их определенному объекту. В следующей строке кода объект `money` имеет свойство `myAccount`, которому присваивается значение 5000:

```
money.myAccount=5000;
```

Вы можете создавать собственные объекты или использовать один из встроенных во Flash объектов, такой как `Date`, `Array` и `Color`.

## Клипы

Клипы — отдельные анимации, которые воспроизводятся во Flash-фильме независимо, потому что имеют свою *собственную* временную шкалу. Клипы имеют свойства `_alpha` и `_rotation`, которым присваиваются значения. В приведенной ниже строке экземпляр клипа `spinClip` поворачивается на 180 градусов:

```
spinClip._rotation=180;
```

Клипы также имеют методы, которые вы можете использовать, чтобы управлять воспроизведением клипов. В следующем примере головка воспроизведения переходит к первому кадру клипа `audioClip`, после чего воспроизведение останавливается:

```
audioClip.gotoAndStop(1);
```

## Неопределенный тип данных

Неопределенный тип используется для того, чтобы указать на отсутствие данных. Переменная, которая не имеет *определенного* значения, возвращает значение

`undefined`. Переменную без значения нельзя считать «пустой» (имеющей значение `null`), она не определена, а значит, не существует.

Если вы хотите создать пустую переменную, то ее значение необходимо установить равным `null`. В этом случае переменная существует, но она пуста. Например, следующая конструкция возвращает значение `undefined`, так как переменная объявлена, но не имеет никакого значения:

```
var one;
```

А такая конструкция возвращает `null`:

```
var two=null;
```

Более детальную информацию о переменных вы найдете в следующем разделе.

## Переменные

В *переменных* хранятся значения, которые по ходу выполнения сценария меняются. Переменные подобны карманам, в которые можно поместить что-нибудь, хранить некоторое время, а затем, когда это потребуется в фильме, извлечь. Прежде чем использовать переменную, ее необходимо объявить. Затем ее нужно инициализировать, чтобы сообщить Flash, данные какого типа в ней хранятся. После инициализации переменная будет содержать полученное значение, пока вам не понадобится его изменить. Чтобы изменить значение переменной, нужно его увеличить, уменьшить или повторно инициализировать переменную.

Переменные могут использоваться для хранения данных любого типа: строкового, числового, булевого, объекта или клипа. Например, переменная может быть инициализирована как число или как набор строковых данных:

```
x=24;
y="myName";
```

Переменная может также использоваться в качестве счетчика цикла. Это можно сделать, например, так:

```
j=2
while(j>0){
duplicateMovieClip(_root.ship,"ship"+j,j);
setProperty("ship"+j,_y,(_root.ship._y+50*j));
trace("Копия"+j);
j=j-1;
}
```

В этом примере переменной `j` присваивается значение 2. Каждый раз, когда она проходит через цикл `while`, ее значение уменьшается на 1. Переменная `j` также используется в аргументах метода `duplicateMovieClip()` для указания имен копий клипов и уровней. Наконец, в действии `setProperty()` переменная `j` применяется в аргументе, с помощью которого копии размещаются на расстоянии `50*j` пикселей вверх по оси `Y` от точки регистрации первоначального клипа. Эти примеры показывают, что одну переменную можно неоднократно использовать в сценарии для выполнения разнообразных задач.

Переменной необходимо присвоить соответствующее имя. В имени переменной могут содержаться комбинации буквенных символов, чисел, символа подчеркивания (`_`), и символа доллара (`$`). Просмотрите еще раз главу 14, в которой рассказано о спецификациях переменных. Однако имя переменной не может быть ключевым словом языка ActionScript и должно быть уникальным в рамках фильма. Неплохо присвоить переменной какое-нибудь значащее имя, напоминающее о роли, которую она будет исполнять в вашем фильме. Например, если посетители вводят свои имена в текстовое поле, ему (соответствующей этому полю переменной) дают имя `siteVisitor`. Как только вам понадобится имя посетителя, вы можете вызвать эту переменную и сразу получить имя. Для того чтобы вывести прощальное сообщение в текстовом поле `goodbye`, можно написать следующий код:

```
goodbye="Спасибо за посещение "+siteVisitor;
```

В предыдущем примере сценария вы, возможно, заметили строку `trace ("Копия"+j);`. Специальная функция `trace` языка ActionScript позволяет контролировать различные элементы сценария. В этой строке слово `Копия` связывается с текущим значением переменной `j`. Функция `trace` может оказаться очень полезной, потому что позволяет отслеживать значения переменных, изменяющихся в процессе выполнения сценария.

Все что вы должны сделать для использования функции `trace` — это перечислить в круглых скобках после ее имени переменные, значения которых вы хотите проконтролировать. При тестировании фильма в окне `Output` будут выводиться значения этих переменных. Все символы, указанные в кавычках, будут обработаны как строка и отобразятся в окне `Output` в том виде, в каком они были введены. Это может оказаться полезным для указания переменных, значения которых необходимо проследить. Для нашего примера функцией `trace ("Копия"+j);` в окне `Output` будет выведено:

```
Копия2
Копия1
```

Этот результат отражает изменение значения переменной `j`, поскольку она проходит цикл `while` дважды, каждый раз уменьшая свое значение на единицу.

## Область действия переменных

Одна из самых важных характеристик переменных — их доступность из различных элементов фильма, то есть область их действия. Область действия переменной указывается при ее объявлении. Поскольку вы используете переменные для хранения изменяющихся значений, необходимо, чтобы они были доступны. В зависимости от области действия переменные могут быть либо глобальными, либо локальными.

Если переменная глобальная, то она доступна из любого элемента фильма. В любом сценарии вашего фильма вы можете обратиться к глобальной переменной или изменить ее значение. Для того чтобы объявить глобальную переменную, используется ссылка `_global`. Например, в следующей строке кода создается глобальная переменная `store1` и ей присваивается значение `1`:

```
_global.store1=1;
```

Значение переменной `store1` может быть считано в любой части фильма с помощью функции `trace()`:

```
trace(store1);
```

Поскольку переменная глобальная, для доступа к ней достаточно указать ее имя.

Переменная, определенная в клипе, который является элементом главного фильма, также доступна из любого элемента фильма. Однако при работе с такой переменной требуется всегда указывать путь к этому клипу, иначе Flash не будет знать, где найти значение, которое содержит переменная. Например, введя приведенную ниже строку сценария, вы объявите переменную, дав ей имя и присвоив значение:

```
store2=2;
```

Если вы создали эту переменную в фильме главной временной шкалы, ее значение может быть получено в любой точке фильма с помощью следующей ссылки:

```
_root.store2;
```

Если бы вы создали `store2` в клипе (например, в клипе `sprocket`), то обратиться к ней можно, указав следующий путь:

```
_root.sprocket.store2
```

Локальные переменные используются иначе. Они «привязаны» к функциям, в которых объявлены, и могут быть изменены только в пределах тела функции, в которой они находятся. Это может быть полезно при работе со значением, которое должно существовать в течение короткого периода времени. Для объявления локальной переменной используется действие `var`. Например:

```
function init3C(){
var store3=3;
}
```

В этом фрагменте сценария объявляется локальная переменная `store3` и ей присваивается значение 3. Будучи локальной, переменная `store3` доступна только в пределах тела функции (внутри фигурных скобок), в которой она постоянно находится, и содержит значение только во время выполнения функции.

Чтобы создать глобальную переменную

- ▶ в режиме **Normal**, выберите ссылку **Objects** ▶ **Core** ▶ **\_global** и укажите значение переменной в поле **Expression**;
- ▶ в режиме **Expert**, наберите имя переменной (не забудьте ввести перед переменной ссылку `_global`) и задайте значение переменной после оператора присваивания.

Чтобы создать переменную, определенную в клипе,

- ▶ в режиме **Normal** выберите действие **Actions** > **Variables** ▶ **Set Variable**, введите имя переменной в поле **Variable** и задайте ее значение в поле **Value**;
- ▶ в режиме **Expert** наберите имя переменной и воспользуйтесь оператором присваивания (`=`), чтобы присвоить ей значение.

Чтобы создать локальную переменную

- ▶ в режиме Normal, выберите действие Actions ▶ Variables ▶ var и укажите имя переменной в поле Variables; чтобы присвоить ей значение, следуйте инструкции по созданию переменной, определенной в клипе;
- ▶ в режиме Expert, введите с клавиатуры действие var, затем имя переменной и воспользуйтесь оператором присваивания (=), чтобы присвоить ей нужное значение.

## Операторы

Операторы используются для того, чтобы манипулировать значениями переменных и констант (объединять, удалять, сравнивать их и т. д.). С обеих сторон оператора находятся значения, называемые *операндами*. Оператор выполняет над ними определенную операцию и возвращает результирующее значение выражения.

Если в выражении используется более одного оператора, они выполняются в зависимости от *приоритета* каждого из них. Сначала выполняются операторы с самым высоким приоритетом, затем оставшиеся в порядке убывания приоритета. В случае равенства приоритетов операции выполняются в порядке следования операторов, то есть в направлении слева направо. Например:

```
total=3+4*5
total=23
```

Согласно правилам математики, сначала будут перемножены 4 и 5, а затем к результату прибавлено 3. В табл. 15.2–15.5 приведен список некоторых операторов ActionScript, а также указаны математические операции, которые они выполняют.

### Арифметические операторы

Арифметические операторы используются для сложения, вычитания, умножения и деления операндов выражения. В табл. 15.2 приведен список арифметических операторов ActionScript. Они расположены по мере убывания приоритета.

**Таблица 15.2.** Арифметические операторы

| Оператор | Действие          |
|----------|-------------------|
| ++       | Декрементирование |
| --       | Инкрементирование |
| *        | Умножение         |
| /        | Деление           |
| %        | Деление по модулю |
| +        | Сложение          |
| -        | Вычитание         |

### Операторы сравнения

Эти операторы используются для сравнения значений двух операндов и возвращают значение true или false исходя из результата сравнения. В табл. 15.3 приведен список операторов сравнения ActionScript. Они имеют одинаковый приоритет и обрабатываются в порядке слева направо.



Таблица 15.3. Операторы сравнения

| Оператор | Действие         |
|----------|------------------|
| <        | Меньше           |
| <=       | Меньше или равно |
| >        | Больше           |
| >=       | Больше или равно |

## Логические операторы

Логические операторы используются, чтобы сравнить два значения типа `Boolean` и вернуть третье значение: `true` или `false` (тоже типа `Boolean`). Оператор `&&` (логическое И) возвращает `true`, если проверка обоих условий возвращает `true`. Оператор `||` (логического ИЛИ) возвращает `true`, если проверка одного из условий возвращает `true`, и `false`, если проверка двух условий возвращает `false`. Оператор `!` (логическое НЕ) инвертирует значение выражения (например, `!false = true`). В табл. 15.4 приведен список логических операторов ActionScript. Операторы расположены по мере убывания приоритета.

Таблица 15.4. Логические операторы

| Оператор | Действие       |
|----------|----------------|
| !        | Логическое НЕ  |
| &&       | Логическое И   |
|          | Логическое ИЛИ |

## Операторы равенства и присваивания

Операторы равенства используются для проверки равенства двух операндов, они возвращают значения `true` или `false`. Операторы присваивания применяются для задания значений переменных и их инициализации. В табл. 15.5 приведен список операторов равенства и присваивания языка ActionScript. Операторы равенства расположены по мере убывания приоритета. Операторы присваивания (`+ =`, `- =`, `* =` и т. д.) имеют одинаковый приоритет и обрабатываются в порядке следования (слева направо).

Таблица 15.5. Операторы равенства и присваивания

| Оператор          | Действие                                  |
|-------------------|-------------------------------------------|
| <code>= =</code>  | Равенство                                 |
| <code>== =</code> | Строгое равенство                         |
| <code>!=</code>   | Неравенство                               |
| <code>!= =</code> | Строгое неравенство                       |
| <code>=</code>    | Присваивание                              |
| <code>+ =</code>  | Присваивание результата сложения          |
| <code>- =</code>  | Присваивание результата вычитания         |
| <code>* =</code>  | Присваивание результата умножения         |
| <code>/ =</code>  | Присваивание результата деления           |
| <code>% =</code>  | Присваивание результата деления по модулю |

**ПРИМЕЧАНИЕ**

Операторы побитовой обработки значений не рассматриваются в этой книге. Для того чтобы побольше узнать о них, изучите документацию, поставляемую вместе с Flash MX.

**Действия**

Действия — встроенные команды ActionScript, применяющиеся по отношению к фильму или его компонентам. Например, действие

```
gotoAndPlay(5);
```

указывает головке воспроизведения фильма перейти к кадру 5 и продолжить воспроизведение фильма.

Вот другой пример:

```
duplicateMovieClip("myClip"."myOtherClip".1);
```

Это действие создает копию клипа `myClip`, называет ее `myOtherClip` и размещает на уровне, номер которого на единицу больше уровня исходного клипа.

Списки доступных действий находятся в подкатегориях категории Actions панели Actions.

**Условные конструкции**

Условные конструкции — это блоки сценария, содержащие условия, которые должны быть проверены. Проверка возвращает значение `true` или `false`. В зависимости от полученного результата в сценарии выполняются соответствующие выражения. Вот основные условные конструкции языка ActionScript — `if`, `if ... else`, `switch`, `while`, `do ... while` и `for`. Они реализуются с помощью соответствующих действий.

**Ветвления**

Действие `if` проверяет условие и возвращает значение `true` или `false`. Проверка условия производится в первой строке данной условной конструкции, а затем, в зависимости от полученного результата, выполняются соответствующие инструкции. Например:

```
if(y<_currentFrame){
 move=Math.PI*y;
 saucerClip._x=saucerClip._x+move;
}
```

Если при проверке условия `y<_currentFrame` возвращается значение `true`, следующие инструкции, заключенные в фигурные скобки, выполняются. Если возвращается значение `false`, код, содержащийся в фигурных скобках, пропускается и выполнение сценария продолжается после закрывающей фигурной скобки. Приведенный пример может быть расширен следующим образом:

```
if(y<_currentFrame){
 move=Math.PI*y;
 saucerClip._x=saucerClip._x+move;
}else{
```

```

 gotoAndPlay(1);
 y++;
}

```

Здесь присутствует компонент `else`. Это означает, что если проверка условия `y < _currentFrame` возвращает `false`, то выполняются инструкции, следующие после ключевого слова `else`.

**MX** Во Flash MX вы можете использовать новую условную конструкцию, которая является альтернативой конструкции `if ... else`. Эта конструкция, названная `switch`, включает ряд ветвей, код каждой из которых будет выполнен в зависимости от значения проверяемого условия. Рассмотрите следующие инструкции:

```

num=1;
switch(num){
 case 1:
 trace("В результате проверки условия в ветви case 1 возвращается значение true ");
 break;
 case 2:
 trace("В результате проверки условия в ветви case 2 возвращается значение true ");
 break;
 case 3:
 trace("В результате проверки условия в ветви case 3 возвращается значение true ");
 break;
 default:
 trace("Поскольку в результате проверки предыдущих условий не было возвращено
 значение true, возвращается значение false")
}

```

При обработке конструкцией `switch` значения `num` проверяется несколько случаев (ветвей `case`). Если возвращается `true`, это говорит о том, что значение `num` равно значению, указанному в соответствующей ветви `case`, и что после проверки условия инструкции, находящиеся в данной ветви, будут выполнены. В этом примере в окне Output была бы выведена фраза «В результате проверки условия в ветви `case 1` возвращается значение `true`», потому что значение `num` равно 1. Если бы в первой строке было `num = 3`, вы увидели бы в окне Output следующее: «В результате проверки условия в ветви `case 3` возвращается значение `true`». Если бы не было верного варианта, то было бы выведено сообщение «Поскольку в результате проверки предыдущих условий не было возвращено значение `true`, возвращается значение `false`».

Обратите внимание на то, что в каждой ветви имеется действие `break`. Это гарантирует, что будут выполнены только инструкции, находящиеся в пределах этой ветви. Если выполнить этот сценарий, но удалить перед этим действия `break`, то будет выведено следующее:

```

В результате проверки условия в ветви case 2 возвращается значение true
В результате проверки условия в ветви case 3 возвращается значение true
Поскольку в результате проверки предыдущих условий не было возвращено значение true,
возвращается значение false

```

Действие `break` прекращает выполнение инструкций в текущем блоке, ограниченном фигурными скобками (`{}`), в котором указаны ветви `case`. Без инструкции `break` оставшиеся условия в теле инструкции `switch` были бы оценены как `true`.

## Циклы

Проверка условия может также использоваться в циклах, которые выполняются определенное число раз. Такой цикл может быть создан с помощью действия `while()`, `do ... while()` или `for()`. Каждый из этих циклов имеет счетчик, который подсчитывает количество выполнений цикла. Обычно в начале цикла счетчик инициализируется как переменная, и ее значение уменьшается или увеличивается при каждом его повторении. Функции перечисленных циклов практически идентичны, но способы, которыми каждый из них реализуется, уникальны.

Цикл `while()` устанавливает условие. Если в результате проверки условия возвращается значение `true`, выполняются инструкции, указанные в теле цикла (в фигурных скобках). В следующем примере переменная `k` используется как счетчик, и ее значение уменьшается каждый раз при прохождении цикла. Этот цикл выполняется три раза:

```
k=3;
while(k>0){
 duplicatMovieClip("spotClip","spot"+k,k+1);
 k=k-1;
}
```

В цикле `do...while()` сначала выполняются инструкции, а уже затем проверяется условие, чтобы определить, продолжать ли выполнение цикла. Если условие возвращает значение `true`, выполнение цикла продолжается. В этом цикле инструкции всегда выполняются не менее одного раза, даже если условие возвращает `false`. Например:

```
k=3;
do{
 duplicatMovieClip("spotClip","spot"+k,k+1);
 k=k-1;
} while(k>0);
```

В цикле `for()` вся необходимая информация помещается в одной строке: инициализация счетчика, условие, модификация значения счетчика. Например:

```
for(k=3;k>0;k=k-1){
 duplicateMovieClip("spotClip","spot"+k,k+1);
}
```

Из приведенного фрагмента кода вы можете видеть, что счетчик и условие цикла здесь те же, что и в примерах с циклами `while()` и `do ... while()`. Однако вся информация, необходимая для управления циклом, экономно разместилась в одной строке.

Действие `for( ... in)` работает, используя свойства или вложенные объекты самого объекта. Это действие может изменить свойства объекта либо клипа, с его помощью можно также управлять объектами и клипами. Дополнительную информацию о действии `for( ... in)` можно найти в приложении Б.

## Функции

В предыдущей главе функции были представлены как подпрограммы обработки информации. Вы устанавливаете для них набор аргументов, и они, используя эти аргументы, выполняют определенную задачу. Применение функций может оказаться очень эффективным, если есть ряд задач или операций, которые должны

быть выполнены в фильме много раз. Например, если требуется часто изменять размеры области отображения кадров клипа на 50 % от его исходных размеров, можно реализовать это с помощью функции. Вместо того чтобы каждый раз использовать свойства `_xscale` и `_yscale`, можно вызвать соответствующую функцию.

Сначала необходимо ее объявить. Синтаксис объявления функции следующий:

```
function имя_функции(аргументы){
 выражение(я);
}
```

В следующем примере приведена функция `Half`, которая уменьшает горизонтальный и вертикальный размеры области отображения кадров клипа на 50 % и перемещает его на 100 пикселей влево и вниз:

```
function Half(myClip){
 var x=myClip._x;
 var y=myClip._y;
 myClip._x=x-100;
 myClip._y=y-100;
 myClip._xscale=50;
 myClip._yscale=50;
}
```

В этой функции клип `myClip` используется как аргумент. Когда данная функция вызывается, размеры области отображения кадров клипа, указанного в круглых скобках, будут изменяться. Сценарий, содержащий такую функцию, лучше связывать с начальными кадрами фильма, в этом случае функция будет обрабатываться сразу при загрузке фильма.

#### СОВЕТ

При создании функции очень важно дать ей уникальное имя, чтобы впоследствии избежать конфликта между функциями, переменными и ссылками. Если функция и переменная названы одинаково, к примеру `Half`, то будьте готовы к возможным ошибкам. Если вы все же не можете использовать другое имя, то примените префикс, например `funHalf` для функции и `varHalf` для переменной.

Если вы создали и объявили функцию, ее можно вызвать из любого элемента фильма — в любом сценарии, в любом обработчике событий — а также из любого фильма. Вот пример, в котором функцию `Half` вызывают, когда кнопка, с которой связан данный сценарий, нажата и отпущена:

```
on(release){
 _root.Half(invaders.rogue_1);
}
```

В этом сценарии клип `rogue_1` (вложенный в клип `Invaders`) перемещен и размеры его области отображения кадров уменьшены на 50 % за счет использования функции `Half`. Путь к клипу `rogue_1` передается функции как аргумент. Поскольку функция `Half` была создана в фильме главной временной шкалы, при вызове ее из фильма другой временной шкалы необходимо использовать ссылку `_root`.

## Объекты

Объект — один из типов данных Flash. Как и данные всех остальных типов, он может быть использован в сценариях ActionScript. Объекты являются очень важными компонентами языка, потому что содержат информацию, касающуюся различных элементов фильма. Объекты имеют свойства, значения которых могут быть установлены и изменены. Они также имеют методы, являющиеся, по сути, встроенными функциями, которые могут использоваться для того, чтобы задавать необходимые значения или выполнять некоторые действия.

Ниже представлен список предопределенных объектов ActionScript. Каждый объект имеет соответствующий набор методов.

- ▶ Объект Arguments. Массив, который используется для управления аргументами, передаваемыми в функцию.
- ▶ Объект Array. Применяется для получения и изменения значений и позиций элементов массива.
- ▶ Объект Boolean. Используется для возврата логического или строкового значения true либо false.
- ▶ Объект Button. Кнопками теперь можно управлять как объектами, используя их собственные наборы методов и свойств.
- ▶ Объект Capabilities. Применяется для определения технических характеристик системы и версии проигрывателя, который воспроизводит фильм.
- ▶ Объект Color. Позволяет преобразовывать и задавать RGB-цвета клипа.
- ▶ Объект Date. Позволяет получить текущую информацию о дате и времени.
- ▶ Объект Key. Используется для выявления нажатий клавиш во время воспроизведения фильма.
- ▶ Объект LoadVars. Применяется для передачи переменных между Flash и сервером.
- ▶ Объект Math. Позволяет обращаться к математическим константам и функциям для выполнения математических расчетов в процессе воспроизведения фильма.
- ▶ Объект Mouse. Позволяет отображать или не отображать указатель мыши, а также выполнить для него сценарий во время воспроизведения фильма.
- ▶ Объект MovieClip. Управляет свойствами клипа и его воспроизведением.
- ▶ Объект Number. Используется при проведении вычислений.
- ▶ Объект Object. Универсальный объект, способный использовать методы других объектов; применяется для создания пользовательских объектов, свойств и т. и.
- ▶ Объект Selection. Управляет доступными для редактирования текстовыми полями.
- ▶ Объект Sound. Применяется для управления воспроизведением звука в фильме.
- ▶ Объект Stage. Используется для изменения размера и положения области действия Flash-фильма.
- ▶ Объект String. Применяется при работе со строковыми значениями.
- ▶ Объект TextField. Используется для управления характеристиками и свойствами экземпляра текстового поля.

- ▶ Объект `TextFormat`. Применяется для форматирования текста в пределах текстовых полей.
- ▶ Объект `Video`. Используется для передачи видео непосредственно из видеокамеры во Flash-фильм.
- ▶ Объект XML. Обеспечивает связь Flash-фильма с XML-документами.
- ▶ Объект XML Socket. Позволяет с компьютера пользователя, на котором запущено приложение Flash Player, связываться непосредственно с сервером.

## Разработки

Пример интерактивного приложения, которое можно создать с помощью Flash, можно найти на Web-узле `Destroy Everything`, разработанном Никола Стумпо, по адресу [www.abnormalbehaviorchild.com/nuovo/default.html](http://www.abnormalbehaviorchild.com/nuovo/default.html) (рис. 15.1). Этот сайт демонстрирует довольно интересную смесь музыки, текста, изображений, анимации и интерактивности. Небольшие «уловки» ActionScript, как их называет Никола, оживляют проект. Наибольшее впечатление производит система навигации, в которой используются интерактивные анимации и уникальный интерфейс.

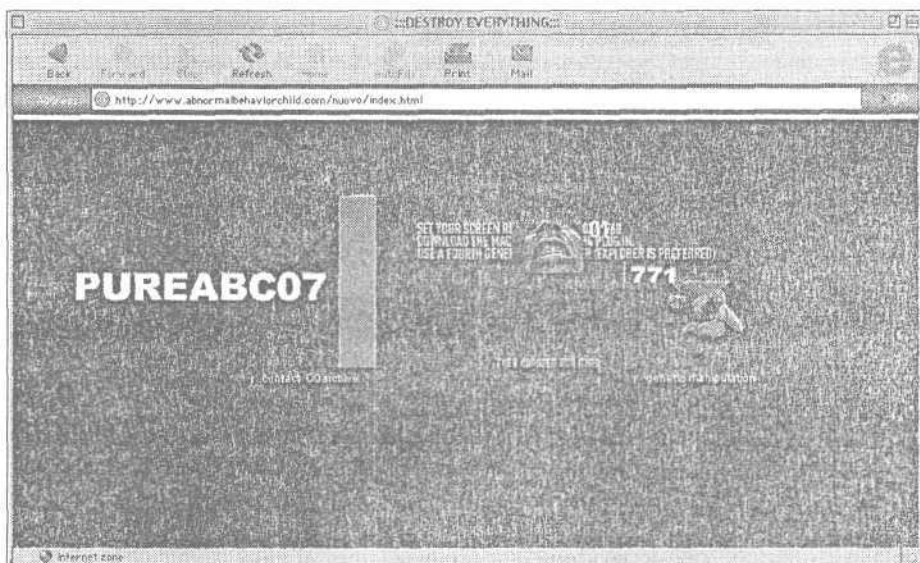


Рис. 15.1. Web-узел Destroy Everything

## Резюме

В этой главе описаны правила синтаксиса ActionScript и элементы языка. Вы узнали, как в инструкциях ActionScript используются знаки пунктуации и символы верхнего и нижнего регистров, познакомились с типами информации, которые поддерживает ActionScript, а также со способами ее хранения и проверки.

В следующей главе мы продолжим изучение этого языка и расскажем, как с помощью ActionScript управлять различными элементами фильма.

## Глава 16

# Повышение интерактивности фильмов с помощью ActionScript

Видимо, вы уже поняли, что изучение всех технических аспектов ActionScript является достаточно сложной задачей. Если перед вами не стоит цель стать узким специалистом в этой области, наверное, не стоит вникать во все тонкости. Но сразу заметим, что если вы будете терпеливы, то по мере изучения языка увидите и приятные моменты при работе с ActionScript, которых намного больше, чем неудачных.

Вы уже изучили основы этого языка и готовы применять приобретенные навыки на практике. Возможности, предоставляемые ActionScript, фактически безграничны: иногда сложнее придумать, что же нужно сделать в фильме, чем реализовать эти идеи. При разработке фильмов думайте о том, *что* вы хотите создать, а не *как* это сделать, то есть сначала ставьте перед собой художественные и дизайнерские цели, а потом выясняйте, какими средствами их можно достичь.

Поскольку охватить все возможности Flash и ActionScript в одной книге невозможно, цель данной главы — дать вам толчок для начала работы. Здесь вы найдете примеры, которые раскрывают основы ActionScript и Flash, и сможете использовать их в качестве образца. Как всегда, помните, что создавать фильм нужно по фрагментам, часто их сохранять и каждый раз проверять фильм при добавлении новой функции или изменении сценария. Если вы будете следовать этим рекомендациям, то без труда сможете определять причины возможных ошибок.

В этой главе рассмотрены следующие темы:

- ▶ создание сложных сценариев для кнопки;
- ▶ разработка регулятора управления элементами интерфейса фильма;
- ▶ программная обработка нажатий клавиш;
- ▶ управление несколькими временными шкалами и слоями;
- ▶ свойства ActionScript для управления клипами;
- ▶ компоненты Flash MX;
- ▶ программное изменение графических элементов фильма.

## Программная обработка действий пользователя

Flash — великолепное приложение для создания векторных анимаций, а благодаря использованию языка ActionScript в нем можно разрабатывать интерактивные игры, интерфейсы, презентации, музыкальные проигрыватели и многое другое.

Основными устройствами взаимодействия пользователя с компьютерной программой являются мышь и клавиатура. Когда-нибудь (скорее всего, раньше, чем



мы думаем) мышь и клавиатуру выставляют в музеях всего мира, а посетителей будет удивлять, что люди действительно использовали эти предметы. Но пока этот день не настал, мы с вами находимся в каменном веке, где необходимо применение этих основных устройств ввода данных. ActionScript предоставляет ряд действий, свойств и методов, которые предназначены для использования мыши и клавиатуры.

Хотя описанные в главе манипуляции можно выполнить, когда панель Actions находится в режиме Normal, все рекомендации приводятся для режима Expert. Режим Normal описан в главе 12, а режим Expert — в главе 14.

## Сложный сценарий для кнопки

Кнопка является одним из наиболее часто используемых и интуитивно понятных элементов интерфейса. Роль кнопок в интерактивном фильме проста: зритель нажимает на нее, после чего происходит некоторое действие или действия. Для кнопки, как и для клипа, можно присоединить действия языка ActionScript. При нажатии кнопки такие присоединенные действия выполняются в соответствии с программой-обработчиком этого события. Подробная информация о кнопках и обработчиках приведена в главе 13.

В самом простом случае для кнопки задается единственный обработчик с единственным действием. Например:

```
on(release){
 gotoAndPlay(15);
}
```

После отпускания кнопки мыши выполняется действие, которое перемещает головку воспроизведения к кадру в позиции 15, и продолжается воспроизведение фильма. Это пример обработчика события «отпускание кнопки мыши» (*release*). Такие обработчики позволяют организовать простую навигацию в фильме. Однако сценарии для кнопок позволяют сделать гораздо больше.

В рамках одного обработчика события мыши может выполняться много действий. Это означает, что, нажав на кнопку, вы можете заставить ее делать множество вещей: управлять основной временной шкалой и другими шкалами, воспроизводить и останавливать воспроизведение звуков, вызывать математические функции и выполнять вычисления, задавать или изменять значения переменных.

Кроме того, с одной кнопкой можно связать несколько обработчиков, что позволяет увеличить функциональность кнопки. Например:

```
on(rollOver){
 swish.start();
}
on(press){
 click.start();
}
on(release){
 ring.start();
}
```

Этот пример сценария воспроизводит три различных звука (*swish*, *click* и *ring*) в случае наступления различных событий. Звук *swish* воспроизводится тогда,

когда указатель мыши находится над кнопкой, `click` — когда кнопка мыши нажата и `ring` — когда кнопка мыши отпущена. Эти сценарии создают очень музыкальную и интерактивную кнопку!

Методика использования нескольких обработчиков и действий позволяет повысить функциональные возможности кнопок в фильме Flash.

## Использование объекта Button

**MX.** Одним из новшеств Flash MX является объект Button. Благодаря этому объекту появилась возможность назначать уникальное имя кнопке и управлять ее поведением и свойствами программно — в сценариях ActionScript.

Подобно другим объектам, объект Button имеет коллекцию связанных методов, свойств и событий. Каждый метод выполняет свою определенную задачу. Свойства задают атрибуты объекта Button. События наступают при определенных действиях пользователя.

С объектом Button связаны следующие события.

- ▶ `Button.onDragOut`. Происходит, когда указатель мыши перемещен за пределы кнопки при нажатой клавише мыши.
- ▶ `Button.onDragOver`. Происходит, когда указатель мыши перемещен за пределы кнопки, а затем обратно при нажатой клавише мыши.
- ▶ `Button.onKillFocus`. Происходит, когда кнопка теряет фокус.
- ▶ `Button.onPress`. Происходит, когда указатель мыши находится в пределах кнопки и выполняется нажатие кнопки мыши.
- ▶ `Button.onRelease`. Происходит, когда указатель мыши находится в пределах кнопки и кнопка мыши сначала нажимается, а затем отпускается.
- ▶ `Button.onReleaseOutside`. Происходит, когда кнопка мыши была нажата, пока указатель мыши находился в пределах кнопки, а отпущена после выведения указателя за границы кнопки.
- ▶ `Button.onRollOut`. Происходит, когда указатель мыши перемещается за пределы кнопки.
- ▶ `Button.onRollOver`. Происходит, когда указатель мыши перемещается в пределах границ кнопки.
- ▶ `Button.onSetFocus`. Происходит, когда кнопка получает фокус.

После знакомства с событиями объекта Button рассмотрим, как связать функцию с этим объектом. Откройте новый документ Flash и создайте символ кнопки. Подробная информация о разработке кнопок приведена в главе 13. Для того чтобы связать действия с объектом Button, выполните следующие операции.

1. Выберите кнопку в области действия и активизируйте команду Window ▶ Properties (Окно ▶ Свойства) или нажмите клавиши `Cmd/Ctrl+F3`. Откроется панель свойств.

- В поле имени экземпляра введите имя кнопки. В этом примере использовано имя `clicker`.



- Вставьте новый слой во временную шкалу и назовите его `actions`. Выберите первый ключевой кадр этого слоя и активизируйте команду **Window** ▶ **Actions** или нажмите клавишу **F9**, чтобы открыть панель **Actions** (Действия). Введите следующие операторы:

```
clicker.onRelease=function(){
 trace ("clicked");
}
```

Этот сценарий задает функцию для экземпляра кнопки `clicker`. Функция выводит в окне **Output** слово *diced* и выполняется, когда кнопка мыши нажата и отпущена на экземпляре кнопки `clicker`.

- Выберите команду **Control** ▶ **Test Movie**, чтобы проверить работу новой функции. При щелчке па кнопке в окне **Output** должно появиться сообщение *diced*.

Преимущество этой новой методики создания сценария состоит в том, что вместо связывания действия с каждой кнопкой индивидуально, вы можете назначить их для каждой кнопки в одном месте фильма. Если в фильме имеется много кнопок и сценариев для них, можно централизовать задание сценариев для всех этих элементов. Убедитесь, что для каждого сценария объекта `Button` есть соответствующий экземпляр этого объекта, находящийся на временной шкале в том же самом кадре. Если сценарий находится в кадре 1, а соответствующая кнопка в кадре 2, Flash будет выдавать сообщение об ошибке.

Пока все, что вы узнали о новом объекте `Button`, не сильно отличается от способов работы с кнопками в предыдущих версиях Flash. В главе 13 были рассмотрены способы совместного использования кнопок и графических символов для разработки схемы навигации по фильму с возможностью создания закладок. Чтобы реализовать эту схему, мы работали с кнопкой, графическим символом и несколькими сценами фильма. Но если использовать новый объект `Button` языка ActionScript, решение этой задачи потребует меньше времени и сил. А самое главное заключается в том, что применение объекта `Button` делает программы фильма легко масштабируемыми.



Посмотрите файл `navBarButtons.fla`, находящийся в каталоге **Chapter 16** на прилагаемом компакт-диске. Этот файл содержит законченные сценарии.

Чтобы создать область навигации, используя объект `Button` ActionScript, выполните следующие действия.

- Откройте файл `navBarLesson.fla` из каталога **Chapter 16** прилагаемого компакт-диска. Этот файл позволит получить некоторые элементы фильма.
- Откройте клип `navMC` в режиме редактирования символа. Во временной шкале клипа `navMC` выберите первый ключевой кадр слоя `actions` и откройте панель

- Actions (Действия). Введите в этой панели действие `stop()`. Поскольку мы используем этот клип в качестве кнопки, он не должен запускаться сразу же.
3. Выберите команду **Window** ▶ **Properties**, чтобы открыть панель свойств. Задайте имя кадра `_up`. Таким образом вы задаете поведение клипа, соответствующее состоянию `Up` обычной кнопки.
  4. Для остальных ключевых кадров (в позициях 4 и 7) введите имена `_over` и `_down` соответственно (рис. 16.1).

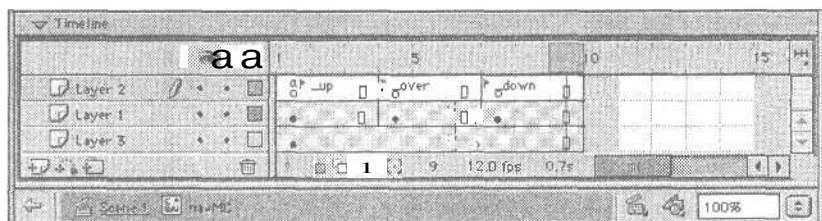


Рис. 16.1. Окончательный вид временной шкалы клипа `navMC`

5. Возвратитесь к главной временной шкале. Обратите внимание, что в нижнем левом углу окна шкалы имеются два экземпляра `navMC`. Для клипа, находящегося слева и подписанного «Section 1», зададим имя `button_one` в панели свойств, а для клипа, подписанного «Section 2», — имя `button_two`. Теперь, когда оба экземпляра клипа имеют свои имена, ими можно управлять из сценария **ActionScript**.
6. Выделите первый ключевой кадр слоя `actions` в главной временной шкале и откройте панель **Actions**. Введите ниже строки `stop` следующий код:

```
button_one.onRollOver=function(){
 button_one.gotoAndPlay("_over");
}
button_one.onPress=function(){
 marker.gotoAndStop(1);
 button_one.gotoAndStop("_down");
 button_two.gotoAndStop("_up")
}
button_one.onRollOut=function(){
 button_one.gotoAndStop("_over");
}
```

Эти инструкции обеспечивают функционирование первой кнопки:

- > если указатель мыши находится на кнопке, выполняется переход к кадру `_over`;
- > при щелчке на кнопке осуществляется переход к первому кадру экземпляра клипа `marker`, а также к кадру с меткой `_down` первой кнопки и кадру с меткой `_up` второй кнопки;
- > когда указатель мыши выходит за границы кнопки, выполняется переход к кадру с меткой `_over` первой кнопки.

7. Скопируйте введенный код и поместите его в нижней части области действия панели Actions. В скопированных инструкциях поменяйте местами имена экземпляров `button_one` и `button_two`. Этим вы зададите функционирование второй кнопки. Полный текст сценария должен быть таким, как указано на рис. 16.2.



Рис. 16.2. Завершенный сценарий работы кнопок

8. Выберите команду `Control ▶ Test Movie`. Проведите указателем мыши над кнопками и пощелкайте на них, чтобы увидеть навигационную область в действии. При активизации кнопки Section 2 выполняется переход к части Section 2 клипа, при щелчке на кнопке Section 1 — к части Section 1 клипа. После щелчка кнопка остается нажатой, чтобы было видно к какой части клипа выполнен переход.

Итак, почему же этот клип действует подобно кнопке? В приведенном примере описан только один из путей работы с объектами `Button`. Вы можете писать сценарии для кнопок непосредственно или же задавать сценарии для клипов и заставлять их вести себя подобно кнопкам. Единственное различие между этим клипом и обычной кнопкой заключается в том, что клип не имеет состояния `Hit`. Это действительно так, но такое состояние для клипа можно «смоделировать». Внимательно посмотрите на временную шкалу клипа `navMC`. Обратите внимание на то, что слой `Layer 3` содержит скрытый графический символ. Этот символ моделирует состояние `Hit`.

В данном примере мы рассмотрели один из простейших вариантов сценария для объекта `Button`. Поскольку клип является независимой анимацией, то подобным образом можно создавать очень сложные сценарии поведения кнопок. Попробуйте использовать временную шкалу клипа, чтобы создать сложные анимированные кнопки и другие динамические графические элементы интерфейса фильма.

## Создание сценария для заказного контроллера

*Заказным контроллером* называется некоторый элемент фильма, отличающийся от кнопки и предназначенный для управления ходом воспроизведения фильма и его отображения на экране. Подобно другим элементам фильма, которые создаются во Flash, для реализуемых контроллеров не существует фактических ограничений: это могут быть поворачиваемые кнопки, вращающиеся меню, виртуальные джойстики и т. д. Основным свойством таких контроллеров является то, что пользователь управляет ими с помощью мыши так, как будто касается контроллера физически. Контроллеры создают иллюзию наличия в интерфейсе фильма некоторых физических объектов.

Для создания такой иллюзии необходимо иметь возможность перетаскивать или перемещать элементы интерфейса фильма. В ActionScript для этого имеются два специальных действия — `startDrag()` и `stopDrag()`. Первое разрешает перемещать клип, а второе отключает эту возможность и фиксирует текущее положение клипа. Синтаксис действия `stopDrag()` приведен ниже:

```
startDrag("target", lock, left, top, right, bottom)
```

Аргументы этого действия следующие:

- ▶ `target` задает экземпляр объекта, перемещение которого необходимо разрешить. Если значение параметра не указано (задана пустая строка ""), разрешается перемещение клипа, с которым связано это действие.
- ▶ `lock` является дополнительным аргументом и имеет булево значение `true`, если точка регистрации клипа связывается с позицией указателя мыши, или `false`, если точка регистрации клипа устанавливается в месте первого щелчка мыши.
- ▶ `left`, `top`, `right` и `bottom` — дополнительные аргументы, которые задают координаты границ области перемещения. Значения этих координат указываются относительно координат временной шкалы, где воспроизводится клип.

## Создание регулятора для прокрутки временной шкалы

Одним из наиболее часто используемых и полезных видов заказных контроллеров является регулятор. *Регулятор* — это ползунок, который может перемещаться по горизонтали или вертикали. В зависимости от положения регулятора изменяется значение некоторого параметра или вид какого-то графического элемента фильма. Регуляторы наиболее часто используются, чтобы установить громкость звука, но могут также применяться для изменения цвета, позиционирования и масштабирования объектов в области действия фильма, загрузки дополнительных клипов, навигации по фильму и т. д. При разработке регулятора сначала создается его ручка, затем задаются границы ее перемещения, а потом готовый регулятор помещается в область действия. После этого для реализации функций регулятора с ним можно связать сценарии ActionScript.

Далее рассмотрим пример создания регулятора, который выполняет прокрутку временной шкалы.



Готовый регулятор вы найдете в файле `slider fla` из каталога Chapter 16 на прилагаемом компакт-диске.

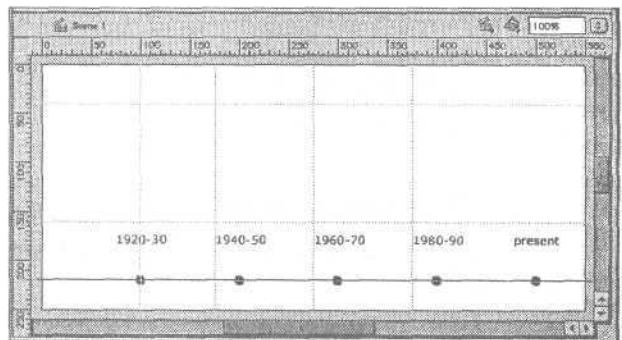
Для создания регулятора выполните следующие действия.

1. Создайте кнопку, которую мы будем использовать в качестве рукоятки регулятора. Это не должна быть кнопка какого-то специального типа, подойдет любой элемент, который можно перетаскивать. После создания рукоятки выберите команду **Edit ▸ Edit Document** или щелкните на кнопке **Back** (Назад), чтобы возвратиться к главной временной шкале.
2. Выберите команду **Insert ▸ New Symbol** или нажмите клавиши **Cmd/Ctrl+F8**, чтобы отобразить окно **Create New Symbol** (Создание нового символа). Введите имя символа `slider knob` и выберите переключатель **Movie Clip**. Нажмите кнопку **OK**, и Flash перейдет в режим редактирования символа.
3. Перетащите кнопку, предназначенную для реализации ручки регулятора, в область действия клипа `slider knob`. Выберите эту кнопку и активизируйте команду **Window ▸ Actions** (Окно ▸ Действия) или нажмите клавишу **F9**, чтобы отобразить панель **Actions** (Действия). Теперь необходимо связать сценарии ActionScript с кнопкой в клипе `slider knob`. Благодаря этому сценарии можно будет использовать для каждого экземпляра кнопки регулятора.
4. В панели **Actions** введите следующие строки:

```
on(press){
 startDrag("", false, leftBounds, topBounds, rightBounds, bottomBounds);
 drag=true;
}
on(release){
 stopDrag();
 drag=false;
}
```

Эти действия включают и выключают возможность перетаскивания кнопки регулятора, когда кнопка нажата и отпущена соответственно. Обратите внимание, что в действии `startDrag()` не заданы конкретные значения для фаниц области перетаскивания. Вместо этого используется набор переменных, которые будут инициализированы позже.

5. Закройте панель **Actions** и возвратитесь к главной временной шкале фильма. Обычно кроме рукоятки для регулятора задается некоторый графический элемент, вдоль которого она перемещается. Используйте какой-нибудь инструмент рисования, чтобы изобразить горизонтальную прямую линию в нижней части области действия. Не забудьте отступить от края области так, чтобы поместилась рукоять регулятора. Затем инструментом **Oval** (Овал) создайте на этой линии пять точек со значениями горизонтальной координаты 100, 200, 300, 400 и 500.

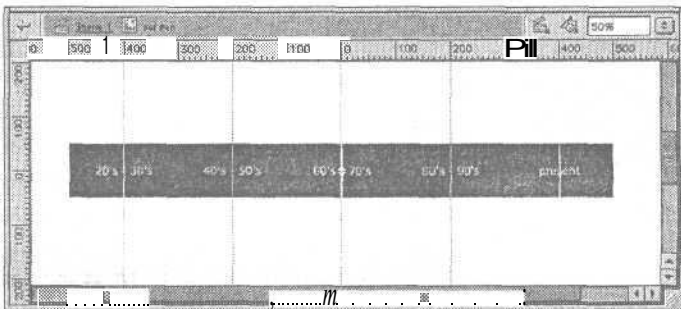


6. Перетащите экземпляр рукоятки регулятора в область действия, расположите его непосредственно поверх первой точки. При необходимости выберите команду View > Rulers, чтобы выполнить точное позиционирование регистрационной точки по линейкам. Задайте имя `slider` для этого экземпляра.
- 7: Теперь, когда клип с рукояткой помещен в фильм, зададим действия ActionScript, необходимые для работы регулятора. Щелкните на экземпляре `slider knob` и выберите команду Window ▶ Actions (F9). Введите следующие операторы в панель Actions:

```
onClipEvent(load){
 topBounds=_y;
 leftBounds=_x;
 rightBounds=_x+400;
 bottomBounds=_y;
}
```

Эти действия выполняются при первом отображении клипа в области действия фильма. Они инициализируют переменные, которые задают границы области перемещения рукоятки регулятора. Согласно указанным значениям рукоятку можно перемещать только по горизонтали от текущего горизонтального значения `_x` вправо на 400 пикселей. Поскольку эти переменные инициализируются только однократно, в начале клипа фиксируется позиция ручки регулятора при запуске клипа, и границы перетаскивания задаются относительно этой позиции. Сохраните фильм и выберите команду Control ▶ Test Movie, чтобы проверить работу регулятора.

8. Теперь добавим в фильм данные, которыми будет управлять регулятор. Создайте новый символ типа клип и назовите его `screen`. Задайте размеры прямоугольной области действия этого клипа 1000 на 100 пикселей. Убедитесь, что прямоугольник позиционирован точно в центре области действия фильма. Проще всего добиться этого с помощью панели Align (Выравнивание), которая открывается после выбора команды Window ▶ Align или нажатия клавиш `Cmd/Ctrl+K`.
9. В клипе `screen` создайте графический символ в виде прямоугольника, имеющего центр в точке регистрации клипа и размеры, совпадающие с размерами его области действия (1000x100 пикселей). Кроме того, поместите в клип деления временной шкалы с историческими датами через 200 пикселей.





10. После завершения работы с клипом `screen` вернитесь к главной временной шкале. Поместите экземпляр клипа `screen` в область действия фильма выше регулятора так, чтобы горизонтальная координата его левого края составила 175 пикселей. При этом точка регистрации клипа попадет в точку с горизонтальной координатой 675 пикселей. Для точного позиционирования используйте панель Info.

Выберите команду Window ▶ Properties (клавиши Cmd/Ctrl+F3), чтобы открыть панель свойств, и введите имя `info` для создания уникального экземпляра клипа `screen`.

11. Теперь необходимо задать сценарий для того, чтобы регулятор управлял данными. Щелкните на рукоятке регулятора и откройте панель Actions. Введите в эту панель следующие строки кода:

```
onClipEvent:centerFrame){
 if(drag==true){
 root.info._x=(675-((root.slider._x-100)*2));
 }
}
```

Эти действия задают позицию по горизонтали `_x` экземпляра клипа `info` относительно позиции `_x` экземпляра клипа `slider` следующим образом. Число 675 — расстояние от левого края области действия до точки регистрации экземпляра `info`. Если рукоять регулятора перемещается вправо на 100 пикселей, клип `info` должен переместиться влево на 200 пикселей, охватывая вдвое большее пространство, чем регулятор. Выражение `_root.slider._x-100` задает нулевое значение начальной координаты X рукоятки регулятора, поскольку ее начальная позиция `_x` находится в точке с горизонтальной координатой 100. Таким образом, для клипа `info` используются значения горизонтальной координаты от 0 до 675. При перемещении клипа `slider` в направлении от точки с координатой 0 вправо клип `info` перемещается от точки с координатой 675 в обратном направлении с удвоенной скоростью. Для сравнения значений координат см. табл. 16.1.

**Таблица 16.1.** Значения горизонтальной координаты для клипов фильма `slider.fla`

| <code>slider._x</code> | <code>slider._x-100</code> | <code>info._x</code> |
|------------------------|----------------------------|----------------------|
| 100                    | 0                          | 675                  |
| 200                    | 100                        | 475                  |
| 300                    | 200                        | 275                  |
| 400                    | 300                        | 75                   |
| 500                    | 400                        | -125                 |

Возникает вопрос, почему перемещение этих клипов происходит с разной скоростью. Клипы `slider` и `info` имеют пять отображаемых фрагментов данных. Но если клип `slider` для отображения его нового фрагмента данных нужно переместить только на 100 пикселей, то клип `info` необходимо переместить на 200 пикселей.

12. Регулятор управления временной шкалой готов (рис. 16.3). Выберите команду Control ▶ Test Movie, чтобы проверить его работу.

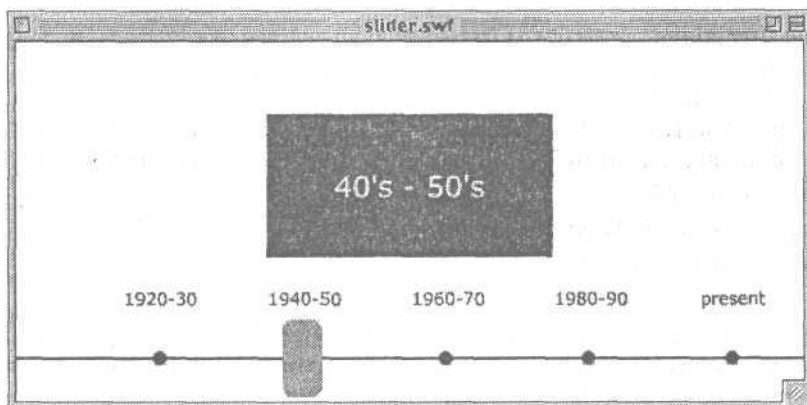


Рис. 16.3. Заключительный вид регулятора

Фильм `slider.fla`, находящийся на прилагаемом компакт-диске, имеет некоторые дополнительные функции, не рассмотренные в этом примере. Он содержит несколько текстовых элементов и слой-маску, задающую окно, через которое пользователь видит названия временных периодов. Создание слоя-маски подробно рассмотрено в главе 8.

## Обработка нажатий клавиш

С помощью сценариев `ActionScript` можно обрабатывать не только события мыши, но и нажатия клавиш. Эти функциональные возможности позволяют использовать клавиатуру в интерфейсе фильма. Такой вид взаимодействия особенно полезен в играх, интерактивных анимациях, контрольных опросах и при проведении презентаций.

Для обработки нажатия клавиш применяется один из predefined объектов `ActionScript` — `Key`. Ниже приведены некоторые методы этого объекта:

- ▶ `isDown(keyCode)`. Возвращает булево значение `true`, если клавиша с кодом `keyCode` нажата, и значение `false`, если это не так.
- ▶ `getCode()`. Возвращает код последней нажатой клавиши во время воспроизведения фильма.
- ▶ `getAscii()`. Подобен методу `getCode()`, но возвращает значение ASCII кода последней нажатой клавиши.

Коды клавиш являются константами языка `ActionScript`. Например, код клавиши для символа `A` — `65`. Полный перечень кодов клавиш приведен в приложении Б.

### ПРИМЕЧАНИЕ

Следует помнить, что обработчик `on(keyPress)`, связанный с кнопками, отслеживает возможные нажатия клавиш во время воспроизведения фильма. Подробнее этот обработчик рассмотрен в главе 12.

## Использование клавиш для выяснения ответов в опросе

Клавиатура может быть очень полезной альтернативой мыши при взаимодействии пользователя с фильмом. Более 90 различных кодов клавиш предоставляют большое количество вариантов выбора действий. Посредством объекта `Key` можно фиксировать клавиши, которые были нажаты во время воспроизведения фильма, а также назначать определенные горячие клавиши и их комбинации для выполнения определенных действий.

### СОВЕТ

Одним из важных фактов, который следует учитывать при проектировании интерфейса интерактивного фильма, является то, что некоторые клавиши имеют специальное назначение во время воспроизведения фильма в среде Flash. Это такие клавиши, как `Return/Enter`, запятая (,) и точка (.). По возможности избегайте их использования в интерфейсе своего фильма. Если все же эти клавиши нужно применить в вашем интерфейсе, отключите их предопределенное назначение посредством выбора команды `Disable Keyboard Shortcuts` (Отключить комбинации клавиш) из меню `Control` (Управление).



В следующем примере мы изучим создание интерфейса клавиатуры, которая будет служить устройством ввода данных при ответе на вопросы анкеты. Откройте файл `keyPress.fla`, находящийся в каталоге `Chapter 16` на прилагаемом компакт-диске, чтобы увидеть тексты нужных сценариев. Обратите внимание, что эти сценарии достаточно просты.

Для получения кодов нажатых клавиш при воспроизведении фильма выполните следующие действия.

1. Поместите клип в область действия фильма. Выберите команду `Window` ▶ `Actions (F9)`, чтобы отобразить панель `Actions`, а затем введите следующие строки кода:

```
onClipEvent(keyDown){
 trace("the key was: "+Key.getCode());
}
```

2. Активизируйте команду `Control` ▶ `Test Movie` и понажимайте клавиши. При нажатии клавиши возникает событие `keyDown` и выполняются действия этого сценария. Функция `trace()` выведет в окне `Output` заданный текст и код нажатой клавиши. Код клавиши предоставляется методом `getCode()` объекта `Key`.
3. Возвратитесь в режим редактирования фильма и откройте панель `Actions`, чтобы отобразить сценарий, который вы только что ввели. Удалите или закомментируйте (//) строку, которая начинается со слова `trace`, и введите следующие строки в тело обработчика `onClipEvent`:

```
if(Key.isDown(32)){
 _root.gotoAndPlay(1);
}
if(Key.isDown(65)){
 _root.gotoAndPlay(3);
}
if(Key.isDown(66)){
 _root.gotoAndPlay(2);
}
if(Key.isDown(67)){
```

```

 _root.gotoAndPlay(3);
}

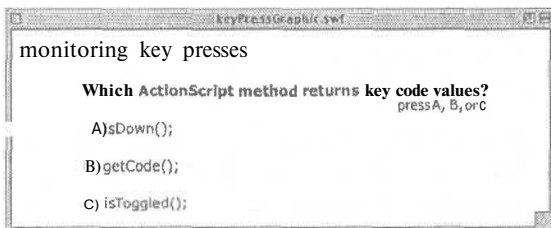
```

В зависимости от нажатой клавиши эти строки сценария переводят головку воспроизведения фильма к определенному кадру. Для нашего примера сценарий проверяет вариант ответа на первый вопрос опроса и перемещает головку воспроизведения. Если нажата клавиша пробела (код клавиши 32), осуществляется возврат к первому кадру. При нажатии клавиши А (код 65) осуществляется переход к кадру 3, клавиши В (код 66) — к кадру 2, а клавиши С (код 67) — также к кадру 3.

#### ПРИМЕЧАНИЕ

Объект Key не чувствителен к регистру символов. То есть при его использовании отсутствует возможность отличить символ «а» от «А».

4. Нажмите клавишу F5, чтобы добавить два дополнительных кадра в слой, который содержит клип. Создайте новый слой и вставьте три ключевых кадра в первые три позиции. Прикрепите действие `stopO` к каждому из этих ключевых кадров. Затем посредством инструмента Text введите слово **CORRECT!!** во второй кадр и слова **SORRY, NO** в третий кадр.
5. Теперь необходимо записать текст вопроса. Не имеет значения какой, но он должен иметь не менее трех возможных вариантов ответа: А, В и С. Для нашего сценария правильным должен быть вариант В. Используйте инструмент Text, чтобы поместить текст вопроса и варианты ответа в первый кадр еще одного нового слоя (третьего по счету, если вы создавали полностью новый фильм).
6. Выберите команду **Control ▶ Test Movie** для проверки фильма. Если все сделано правильно, то при нажатии клавиши В должно появиться сообщение «CORRECT!!», а при нажатии клавиш А и С — сообщение «SORRY, NO». Если сообщения выдаются не так, то либо измените сценарий, либо переупорядочите варианты ответа на вопрос.



## Сценарии ActionScript для фильмов с несколькими временными шкалами

Для оптимального использования ActionScript при создании фильмов Flash нужно понимать, как в композиционном смысле устроен Flash-фильм. Важно знать, из каких элементов состоит такой фильм и каким образом они объединены.

Как средство анимации Flash создает последовательности движения, используя изображения и текст. Если бы этим исчерпывались все его возможности, то по сегодняшним стандартам такое приложение было бы слишком простым. Однако Flash имеет возможность работать со слоями и комбинировать различные анимации. Кроме того, сценарии ActionScript позволяют объединять эти анимации и организовывать взаимодействие между ними. Простые сценарии поведения и функции отдельных анимаций могут использоваться вместе в одном и том же фильме, а поскольку они могут взаимодействовать друг с другом через ActionScript, в фильме выполняются сложные действия.

Для реального увеличения интерактивных возможностей фильма вы должны понять то, как использовать Flash в качестве контейнера или единой арены для анимированного содержимого, и сколько простых фильмов нужно объединить, чтобы создать интерактивный продукт с большой глубиной и потенциалом.

## Как Flash управляет многими временными шкалами фильма

Когда Flash воспроизводит фильм, все его элементы появляются (или слышатся) в области действия фильма. Это создает иллюзию, что все компоненты фильма существуют в единой области и совместно используют одно и то же пространство в среде Flash. Хотя они действительно совместно используют общее пространство области действия, существует определенная иерархия в этом пространстве и у элементов, находящихся там.

Каждый фильм Flash имеет специфическую организацию анимаций, которые этот фильм содержит. Это относится как к анимациям в виде клипов, так и к дополнительным Flash-фильмам, или SWF-файлам, загруженным в главный фильм Flash. Когда дополнительный фильм загружается в основной фильм, он помещается на новый уровень иерархии. Главная временная шкала находится в основании (корне) иерархии и образует уровень 0. Следующий загруженный фильм помещается в стек поверх предыдущего последовательно: уровень 1, уровень 2, уровень 3 и т. д. Поскольку каждый фильм загружается на свой собственный уровень иерархии, это поддерживает его автономию и позволяет воспроизводить фильм в соответствии с его временной шкалой независимо от временных шкал других уровней.

Клипы также имеют похожие иерархические отношения. Один клип может содержать другой клип или клипы. Они, в свою очередь, могут включать в себя дополнительные клипы. Такие клипы называются вложенными. Так же, как фильм может содержать множество независимых временных шкал на различных уровнях иерархии, клипы способны включать другие клипы, и каждый клип способен вести себя независимо. Иерархические отношения в клипе описываются другими терминами. Вложенный клип называется *потомком* или *наследником*, а клип, включающий потомка, называется *родителем* или *родительским*. Клип может быть и родителем и потомком одновременно. Если клип А содержит клип В, а клип В, в свою очередь, содержит клип С, то клип В является одновременно родителем для клипа С и потомком для клипа А. В конечном счете, все клипы являются потомками временной шкалы, в которой они воспроизводятся. Если

упомянутые клипы А, В и С находятся в фильме второго уровня иерархии, то этот фильм для них является родительским, а все эти клипы — наследниками фильма. Сказанное иллюстрирует рис. 16.4.



Рис. 16.4. Иерархические отношения между фильмами и клипами

Так как же все это применяется во Flash? Возьмем, например, фильм, который содержит анимированный персонаж по имени BubbleBody (дутый человечек, человечек, состоящий из пузырьков), или ВВ для краткости. Когда ВВ готовится к прыжку, он приседает, его голова немного наклоняется, а руки разводятся в стороны. При прыжке вверх он выпрямляется, поднимая руки над головой, в это же время его ноги отрываются от земли и сдвигаются.

Эта анимация содержит массу элементов и самая большая трудность — заставить их взаимодействовать так, чтобы анимация выглядела максимально естественной. Обеспечить синхронизацию элементов можно, создавая каждую часть тела ВВ в отдельном клипе и вкладывая эти клипы в один. Затем, если для основного клипа ВВ задать анимацию, при которой он выпрямляется, все остальные клипы (рук, ног, кистей рук), являясь его потомками, выпрямляются вместе с ним. Если для клипов рук ВВ задать анимацию, при которой они будут вытягиваться вверх, то клипы с ладонями также выполнят эти действия. Это достигается благодаря тому, что вложенный клип находится на временной шкале родительского клипа. Если родительский клип изменяется или перемещается, вложенный клип изменяется вместе с ним.

Но при вложении клипов сохраняется возможность управлять каждым клипом независимо. Чтобы создать приветственный взмах рукой ВВ, необходимо приподнять предплечье на 120 градусов и наклонять кисть руки влево и вправо на небольшой угол. В этом случае, поскольку клип, отвечающий за движение кисти, является потомком клипа руки, задать приветственный взмах довольно просто. Для родительского клипа задаем поворот на 120 градусов, при этом кисть также

поворачивается на этот угол, так как соответствующий клип является наследником клипа руки. А затем для клипа, отвечающего за движение кисти, задаем отдельный поворот из стороны в сторону. При этом родительский клип руки не меняет свое положение.

## Абсолютный путь к объекту

Для создания анимации, описанной в предыдущем примере, необходимо в рамках одной временной шкалы выполнять действия ActionScript над объектами, расположенными в другой временной шкале. В нашем примере кнопка или другой элемент управления главного фильма должны выдать команды для клипов руки и кисти.

В ActionScript можно использовать как абсолютные, так и относительные пути к объектам.

*Абсолютный* путь задает точное местоположение экземпляра клипа относительно главной временной шкалы. Для анимации, описанной ранее, абсолютный путь к клипу левой ладони ВВ может быть таким:

```
_root.BB.upperBody.armL.handL
```

В этом случае фильм с анимацией ВВ находится на главной временной шкале, поэтому указывается ссылка `_root`. Если бы фильм с анимацией ВВ находился на уровне 2, абсолютный путь к клипу был бы таким:

```
_level2.BB.upperBody.armL.handL
```

Вы можете использовать абсолютные пути, чтобы послать команды, переменные и любые данные экземплярам клипов и временным шкалам из любого места фильма. Когда иерархия фильма усложняется и возможно возникновение путаницы, надежнее использовать абсолютные пути. Чтобы понять структуру иерархии клипов, составляющих фильм ВВ, обратитесь к рис. 16.5.

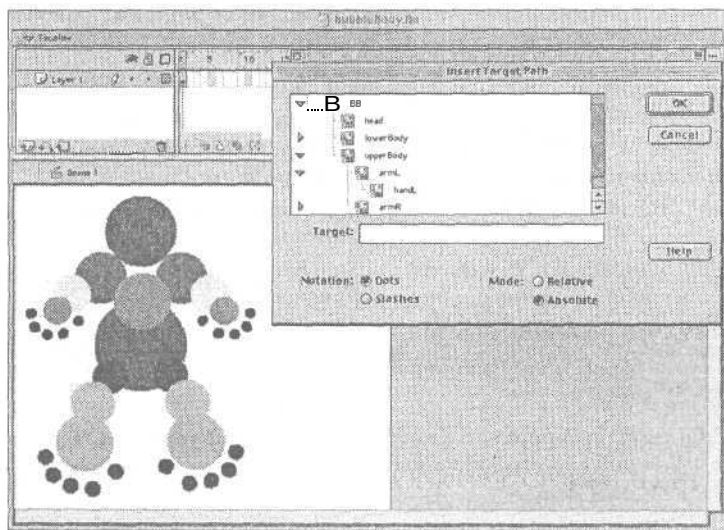


Рис. 16.5. Фильм BubbleBody состоит из девяти отдельных клипов, связанных отношениями наследования

## Относительные пути к объекту

Также возможно указание относительных путей к клипам. Эти пути не содержат всех элементов иерархии, начиная от главной временной шкалы, а указывают местоположение клипа относительно позиции сценария. Возвращаясь к примеру ВВ: если в клипе `handL` необходимо задать действие для клипа `armL`, то можно указать как абсолютный путь

```
_root.BB.upperBody.armL._rotation=0;
```

так и относительный

```
_parent._rotation=0;
```

Как видно из примера, относительный путь намного короче. В таком пути используется ссылка `_parent`, что означает «подняться на один уровень иерархии». Клип `handL` является потомком клипа `armL`, поэтому чтобы задать действия для клипа `armL`, нужно подняться на один уровень в иерархии. Кроме того, в относительном пути может быть применена ссылка `this`, которая относится к текущей временной шкале. Эта ссылка `this` удобна тем, что позволяет обращаться к элементам клипа непосредственно из его собственной временной шкалы, а не вводить длинный путь. Чтобы задать действие для клипа `armR` непосредственно из временной шкалы клипа `upperBody`, можно использовать ссылку `this`:

```
this.armR._rotation=-120;
```

## Управление временными шкалами фильма с помощью ActionScript

Теперь, когда вы знаете, как правильно ссылаться на различные временные шкалы в фильме Flash, нужно научиться правильно применять эти знания на практике для управления воспроизведением и параметрами клипа. В языке ActionScript имеются следующие элементы, необходимые для управления клипами.

- ▶ **Действия**, или команды ActionScript, которые используются, чтобы управлять воспроизведением. Они включают операторы типа `gotoAndPlay()`, `stop()` или `nextFrame()`.
- ▶ **Методы**, определенные в объекте `MovieClip`. Некоторые из этих методов не имеют аналогов среди действий языка, например `getBounds()` и `hitTest()`.
- ▶ **Свойства**, или атрибуты клипа. Многие свойства клипа рассматриваются в этой главе.

Полный перечень этих элементов языка приведен в приложении Б.

В зависимости от объекта, которым нужно управлять с помощью ActionScript, следует использовать различные команды языка, чтобы достигнуть желательного эффекта. Приведенные ниже примеры демонстрируют синтаксис команд, которые обычно используются для загрузки дополнительных фильмов и доступа к экземпляру фильма.

### Загрузка фильма

Для загрузки дополнительного фильма используется действие `loadMovie()`, имеющее следующий синтаксис:

```
loadMovie("URL", "уровень",метод_передачи)
```



Это действие имеет три аргумента.

- ▶ *URL* — определяет местоположение и имя загружаемого фильма (SWF-файла). Этот URL может быть или абсолютным или относительным и задается в виде строки (в кавычках).
- ▶ *Уровень* — строковый аргумент задает место фильма, куда будет загружен новый фильм, например, на уровень `_level1`. Указав целевой путь к загруженному ранее клипу, вы можете заменить его экземпляром нового.
- ▶ *Метод передачи* — дополнительный аргумент, который передает информацию новому клипу. Подробности смотрите в приложении Б.

Следующая строка загружает фильм `levelOne.swf` на уровень 2:

```
loadMovie("levelOne.swf", "_level2");
```

Строка, приведенная ниже, загружает фильм `begin.swf` в клип `intro`, расположенный на главной временной шкале:

```
loadMovie("begin.swf", "_root.intro");
```

Для выгрузки фильма применяется действие `unloadMovie()`, которое имеет только один аргумент, задающий выгружаемый объект:

```
unloadMovie("_level1");
```

Этот сценарий удаляет клип, который в настоящее время воспроизводится на уровне 1.

---

#### ПРИМЕЧАНИЕ

С действиями `loadMovie` и `unloadMovie` связаны действия `loadMovieNum` и `unloadMovieNum`. Они предназначены для решения тех же задач, но имеют несколько другой синтаксис. В зависимости от приложения вы можете использовать первую или вторую пару действий. Подробнее эти действия описаны в главе 12 и приложении Б.

---

## Выбор уровней фильма

Для доступа к клипу в фильме можно использовать относительный или абсолютный путь. После пути через оператор точка указывается действие ActionScript или параметр, значение которого нужно изменить. Например:

```
_root.upperBody.armL.handL._rotation=200;
```

Эта строка, начиная от главной временной шкалы, через цепочку вложенных экземпляров задает значение угла поворота для экземпляра клипа `handL`.

Следующий пример останавливает воспроизведение родительского клипа в кадре 1. Клип-потомок, содержащий эту строку сценария, продолжает выполняться.

```
_parent.gotoAndStop(1);
```

## Добавление элементов интерфейса из компонентов

**MX** Компоненты — одна из наиболее мощных новых функций Flash MX. Они представляют собой дополнительные элементы интерфейса, которые можно перетаскивать из специальной панели в фильм. Компоненты позволяют

формировать полностью настраиваемые элементы интерфейса, многократно используемые разными разработчиками. Их можно представить как «мгновенно получаемые интерфейсы», которые помещаются в фильм путем перетаскивания в область действия.

Если вы использовали клипы типа Smart в предыдущих версиях Flash, то вам понравится и работа с компонентами. Подобно клипам Smart, компоненты предназначены для упрощения процесса разработки интерфейса; но благодаря модульному характеру они позволяют повышать настраиваемость интерфейса на новый уровень.

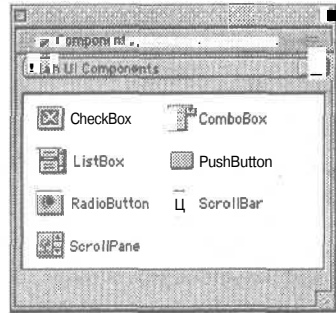
## ПРИМЕЧАНИЕ

Полное рассмотрение возможностей компонентов и их применения выходит за рамки этой книги. Материал данного раздела главы даст вам общее понятие о компонентах и методах их использования в проектах Flash. Для получения дополнительных сведений о компонентах и способах их создания во Flash обратитесь к справочной подсистеме (панель Reference). Кроме того, информацию о компонентах можно найти по адресу [www.macromedia.com/desdev/mx/flash](http://www.macromedia.com/desdev/mx/flash).


Выберите команду Window > Components (Окно > Компоненты) или нажмите клавиши Cmd/Ctrl+F7, чтобы открыть панель Components.

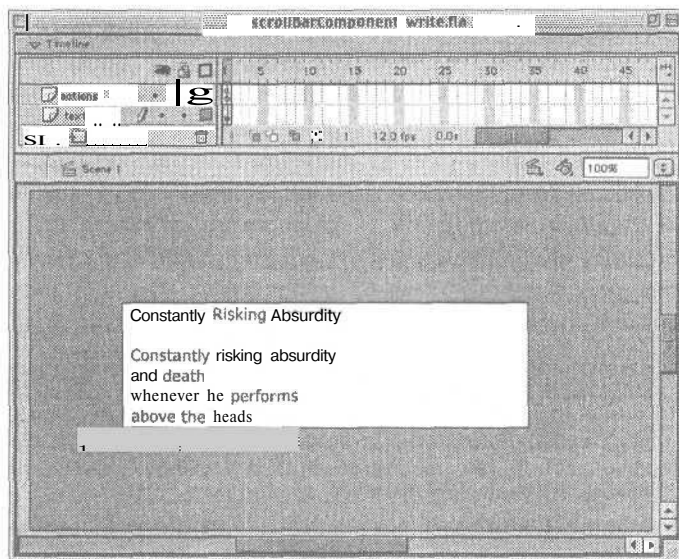
Компоненты имеют семь разновидностей.

- ▶ **CheckBox** — флажок для независимого выбора параметра.
- ▶ **ComboBox** — раскрывающийся список для выбора одного элемента.
- ▶ **ListBox** — поле списка с полосой прокрутки, в котором можно выбрать один или несколько элементов.
- ▶ **RadioButton** — стандартная прямоугольная кнопка.
- ▶ **PushButton** — переключатель. Отдельно не используется, применяется только в группах. В каждой группе в определенный момент может быть активизирован лишь один из переключателей.
- ▶ **ScrollBar** — полоса прокрутки для текстовых полей.
- ▶ **ScrollPane** — панель прокрутки для больших изображений и клипов.



Эти семь компонентов позволяют задать любой современный интерфейс. Основная идея использования этих стандартизованных элементов заключается в том, что любой пользователь, впервые работая с интерфейсом приложения или фильма, интуитивно поймет, как использовать тот или иной элемент.

 Помещение компонентов в интерфейс фильма является простым и понятным процессом. Ниже мы рассмотрим, как добавить компонент ScrollBar к текстовому полю. Откройте файл `scrollBarComponent.fla`, который находится в каталоге Chapter 16 на прилагаемом компакт-диске. В области действия вы увидите, что динамическое текстовое поле содержит текст поэмы «Constantly Risking Absurdity» Лоренса Ферлингетти (рис. 16.6).

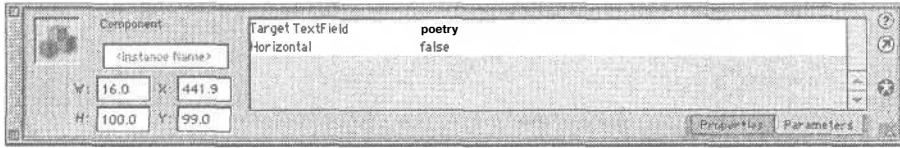


**Рис. 16.6.** Текстовое поле не вмещает весь текст поэмы

Поэма слишком длинная, чтобы поместиться в текстовое поле. Добавив к этому полю полосу прокрутки, можно прокручивать текст и читать его полностью. Для добавления полосы прокрутки к текстовому полю выполните следующие действия.

1. Выберите текстовое поле, к которому нужно добавить компонент. Активизируйте команду **Window** ▶ **Properties** (**Cmd/Ctrl+F3**), чтобы открыть панель свойств. Компонент можно добавлять к динамическому текстовому полю или полю ввода. (Информация о видах текстовых полей приведена в главах 5 и 17.) На панели свойств выберите тип поля и задайте имя `poetry`.
2. Теперь, когда текстовое поле готово, добавьте полосу прокрутки. Выберите команду **Window** > **Components** (**Cmd/Ctrl+F7**), чтобы открыть панель **Components** (Компоненты).
3. Перетащите компонент **ScrollBar** из панели **Components** в область действия максимально близко к правому краю текстового поля. После того как вы отпустите кнопку мыши, Flash должен выровнять полосу прокрутки и текстовое поле автоматически, хотя это срабатывает не всегда. Используйте клавиши управления курсором или мышь, чтобы переместить полосу прокрутки в нужную позицию.
4. Выбрав компонент **ScrollBar** в области действия, вы увидите на панели свойств, что этот компонент имеет два параметра.
  - > **Target TextField** (целевое текстовое поле) — указывает имя текстового поля `poetry`, с которым связана полоса прокрутки. Если значение этого параметра не задано, щелкните на имени параметра и введите имя поля.

- > **Horizontal** — задает положение полосы прокрутки. Если этот параметр имеет значение `true`, полоса расположена горизонтально, а если `false` — вертикально. Убедитесь, что для нашего примера этот параметр имеет значение `false`.



5. Вы, наверное, заметили, что полоса прокрутки имеет меньшую высоту, чем текстовое поле. Высота поля `poetry` составляет 110 пикселей, а высота полосы прокрутки только 100. Это не сложно исправить. Выберите компонент `ScrollBar` и на панели свойств введите в поле `H` значение 110, чтобы сделать полосу прокрутки на 10 пикселей выше.
6. Выберите команду `Control` ▶ `Test Movie` для проверки функционирования полосы прокрутки. Пошелкайте на кнопках со стрелками этого компонента: содержимое текстового поля должно прокручиваться вверх и вниз постранично; при перетаскивании регулятора полосы прокрутки содержимое поля должно прокручиваться плавно.

Одна из самых интересных особенностей компонентов — возможность изменения их внешнего вида. Каждый компонент содержит несколько клипов, называемых оболочками (`skins`). Эти клипы-оболочки буквально являются компонентами компонентов. Изменяя цвет, форму и другие параметры оболочек, можно преобразовать вид любого компонента. Редактирование оболочек является довольно хитроумным процессом, но ниже мы приводим методику, которую можно использовать, чтобы выполнять некоторые основные модификации. Для редактирования оболочки компонента в библиотеке выполните следующие действия.

1. Откройте библиотеку фильма `scrollBarComponent.fla`, выбрав команду `Window` ▶ `Library` (`F11`).
2. Откройте следующую папку библиотеки `Flash UI Components` ▶ `Component Skins` ▶ `FScrollBar Skins`.
3. Выполните двойной щелчок на значке редактируемой оболочки в библиотеке (рис. 16.7).
4. Измените клип по своему усмотрению. Затем вернитесь к главной временной шкале, воспользовавшись командой `Edit` ▶ `Document`. Выполненные изменения станут видны при воспроизведении фильма.

Для получения подробной информации о редактировании компонентов и их оболочек обратитесь к справочной подсистеме `Flash MX` или панели `References`.

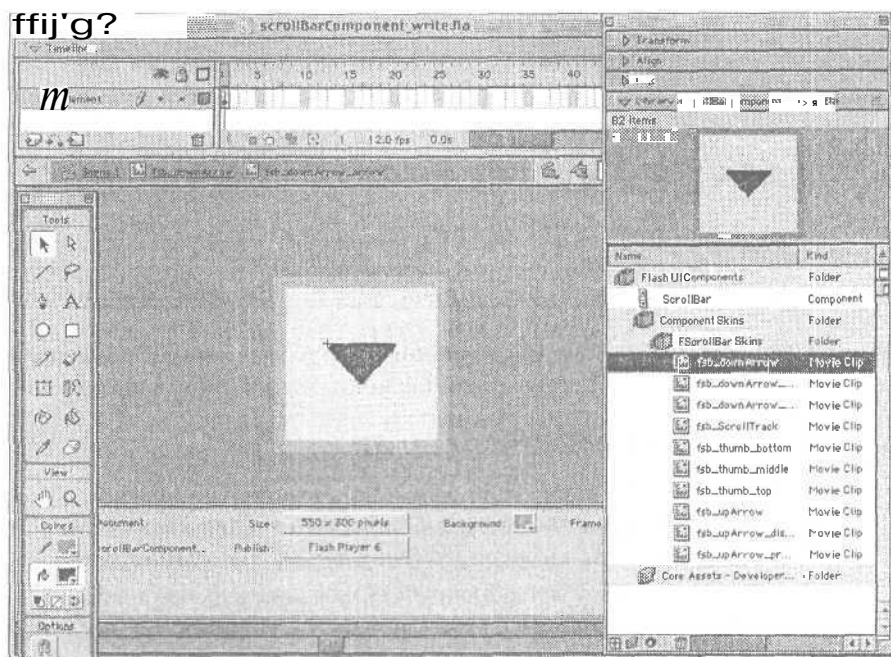


Рис. 16.7. Один из элементов полосы прокрутки подготовлен для редактирования

## Программное управление графическими элементами

Одно из самых важных и уникальных свойств клипов то, что они являются единственными элементами библиотеки Flash, вид которых может быть изменен посредством ActionScript. Это чрезвычайно важный фактор, дающий возможность создавать интерактивные фильмы. Кроме обеспечения навигации по фильму, программно можно изменять любой элемент интерфейса: перемещать, масштабировать, растягивать, скрывать, вращать или выполнять любую комбинацию этих действий в клипе. В конечном счете, вы получаете интерактивный фильм.

ActionScript предоставляет достаточное количество команд управления визуальным содержимым. Ниже перечислены параметры, определяющие внешний вид клипа. Вы можете найти более подробное описание этих параметров в приложении Б.

- ▶ `_alpha` задает прозрачность экземпляра клипа.
- ▶ `_height` задает высоту экземпляра клипа.
- ▶ `_rotation` задает поворот клипа в градусах.
- ▶ `_visible` определяет активность и видимость клипа. Если этот параметр имеет значение `true`, то клип активен и отображается на экране, а при значении `false` клип скрывается, но остается активным в фильме.

- ▶ `_width` задает ширину экземпляра клипа.
- ▶ `_x` и `_y` указывают координаты X и Y регистрационной точки клипа. Если клип находится на главной временной шкале, то координаты отсчитываются относительно левой верхней точки области действия. Если же клип вложен в другой клип, то координаты измеряются относительно точки регистрации родительского клипа (`_parent`).
- ▶ `_xscale` и `_yscale` содержат коэффициент масштабирования клипа по горизонтали и вертикали в процентах.

Кроме этого, изменять внешний вид клипов позволяют методы *предопределенного* объекта `Color`. Используя эти методы, можно манипулировать цветами клипа. Чтобы узнать больше об этих методах, обратитесь к разделу «Программирование изменения цвета» ниже в этой главе или описанию объекта `Color` в приложении Б.

## Создание сценария анимации

Многие из параметров клипа определяют его положение на экране. Быстро изменяя позицию клипа можно создать иллюзию плавного движения, или анимацию. Кроме того, `ActionScript` позволяет изменять размеры или масштаб клипа, что также реализует эффект анимации.

Последующие примеры иллюстрируют возможности `ActionScript` по управлению клипами.


## Анимирование графических элементов

`ActionScript` предоставляет следующие свойства, позволяющие следить за позицией указателя мыши в области действия фильма.

- ▶ `_xmouse` содержит горизонтальную координату позиции указателя мыши.
- ▶ `_ymouse` содержит вертикальную координату позиции указателя мыши.

Координаты измеряются относительно точки начала координат фильма, которая расположена в левом верхнем углу области действия. При использовании этих свойств следует указывать имя соответствующей временной шкалы. Например, получить координаты указателя мыши из главной временной шкалы позволяют следующие строки:

```
hPos=_root._xmouse;
vPos=_root._ymouse;
```

 Рассматриваемый пример находится в файле `flyCursor.fla`, который расположен в каталоге `Chapter 16` на прилагаемом компакт-диске.

Для прикрепления анимации к заказному указателю и ее воспроизведения посредством `ActionScript` выполните следующие действия.

1. Создайте клип с нужной анимацией. В данном примере используется клип с летающим объектом.
2. Перетащите этот клип в область действия фильма и назовите полученный экземпляр `fly`. Выберите этот экземпляр и активизируйте команду `Window ▸ Actions`, чтобы открыть панель `Actions`.

3. Введите в эту панель следующие строки кода:

```
onClipEvent(load){
 Mouse.hide();
}
onClipEvent(mouseMove){
 _x=_root._xmouse;
 _y=_root._ymouse;
}
```

Закройте панель Actions и проверьте фильм. Результаты просто удивительны! Первые действия превращают экземпляр клипа в заказной указатель, скрывая стандартный указатель мыши посредством метода объекта Mouse. Затем обработчик `onClipEvent(mouseMove)` привязывает позицию клипа к положению указателя мыши, таким образом создается эффект, что клип является указателем. Привязка осуществляется за счет приравнивания горизонтальной и вертикальной координат позиции клипа (свойств `_x` и `_y`) к координатам позиции указателя мыши относительно координат главной временной шкалы (`_root`).

4. Вернитесь в режим редактирования фильма и повторно откройте панель Actions для клипа. Введите дополнительные строки:

```
onClipEvent(enterFrame){
 v=_root.fly._x;
 h=_root.fly._y;
 posNeg=Math.round(Math.random()+1);
 if(posNeg>1){
 posNeg=1;
 }else{
 posNeg=-1;
 }
 moveV=Math.round(Math.random()*4*posNeg);
 moveH=Math.round(Math.random()*4*posNeg);
 _root.fly._x=v+moveV;
 _root.fly._y=h+moveH;
}
```

Этот сценарий сохраняет текущую позицию экземпляра клипа, генерирует случайное число в диапазоне от -4 до 4 и прибавляет его к первоначальным координатам экземпляра. Полученный результат используется в качестве координат новой позиции клипа.


5. Выберите команду Control ▶ Test Movie, чтобы проверить работу сценария.

Попробуйте самостоятельно доработать этот пример, изменив его следующим образом:

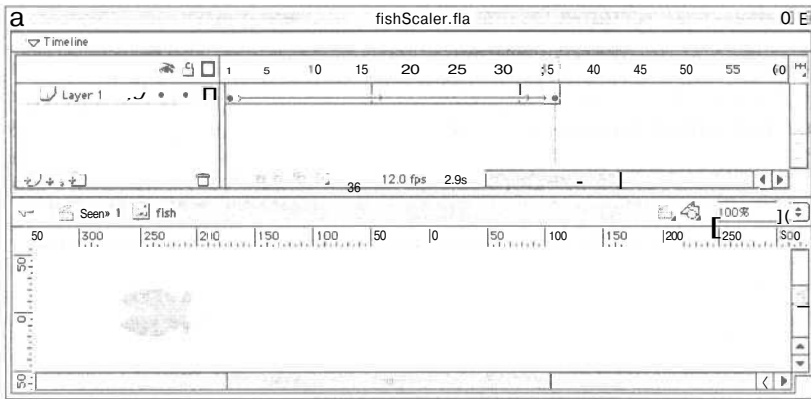
- ▶ вместо одной переменной `posNeg` используйте две различные переменные для формирования горизонтальной и вертикальной координаты. В этом случае полет объекта будет более плавным;
- ▶ измените диапазон изменения координаты (в этом примере равен 4). Таким образом вы можете имитировать более плавный или более неравномерный полет.

## Масштабирование графических элементов

Каждый знает высказывание «всегда есть рыба больших размеров», то есть все относительно в этом мире. Во Flash свойства `_xscale` и `_yscale` применяются для создания «рыбы больших размеров». Этому посвящен следующий пример.

 Свойства `_xscale` и `_yscale` используются, чтобы задать масштаб клипа по горизонтали и вертикали в процентах. Сценарии примера находятся в файле `fishScaler.fla`, который расположен в каталоге Chapter 16 на прилагаемом компакт-диске.

1. Откройте новый файл и выберите команду `Modify > Document`, чтобы задать размеры области действия 400x400 пикселей и частоту смены кадров не менее 12 кадров в секунду. Нажмите кнопку ОК.
2. Создайте новый символ типа `MovieClip` с изображением рыбы. Задайте движение изображения рыбы справа налево (рис. 16.8).



**Рис. 16.8.** Изображение рыбы перемещается от точки `_x=240` до точки `_x=-235`

3. Вернитесь к главной временной шкале и перетащите клип с изображением рыбы в центр области действия; поместите его точку регистрации в точку (200, 200). Задайте в панели свойств имя `fishy` для этого экземпляра клипа. Протестируйте фильм, чтобы увидеть экземпляр `fishy` в действии.
4. Вновь перейдите в режим редактирования клипа, выберите команду `Insert > New Symbol` и создайте новый символ кнопки. Как эта кнопка будет выглядеть, зависит от вашей фантазии. После создания кнопки возвратитесь к главной временной шкале и перетащите вашу кнопку в область действия. Расположите ее в любой точке выше или ниже клипа с рыбой.
5. Теперь все готово для ввода сценария `ActionScript`, который будет масштабировать экземпляр клипа `fishy`. Выберите экземпляр и активизируйте команду `Window > Actions (F9)`, чтобы отобразить панель `Actions`. Введите следующие строки:

```
onClipEvent(load){
 count=0;
}
```



Затем выделите кнопку и откройте панель Actions еще раз. Введите такие строки:

```
on(release){
 count+=1;
 duplicateMovieClip(_root.fishy,"fishy"+count,count);
 setProperty("fishy"+count,_xscale,count*200);
 setProperty("fishy"+count,_yscale,count*200);
}
```

6. Закройте панель Actions и выберите команду Test ▶ Movie. Вы увидите, что при нажатии кнопки кроме существующих рыб на экране появляется новая рыба больших размеров.

Свойства `_xscale` и `_yscale` используются для масштабирования нового клипа-наследника на `count*200` процентов. Поскольку значение переменной `count` увеличивается с каждым щелчком кнопки, новые клипы масштабируются, создавая изображение все большей и большей рыбы.

#### ПРИМЕЧАНИЕ-

Вы могли заметить, что новые экземпляры рыбы движутся быстрее предыдущих. Это связано с тем, что все размеры клипа увеличиваются, в том числе и расстояние, которое проплывает рыба. Однако клип должен проделать увеличенный путь за то же число кадров анимации. Поэтому клип, который проходит 400 кадров за 3 секунды, движется медленнее клипа, которому необходимо пройти 1600 кадров за то же время.

Размеры клипов можно также изменять посредством указания новых значений для свойств `_height` и `_width`. Подробная информация об этих свойствах приведена в приложении Б.

## Программирование изменения цвета

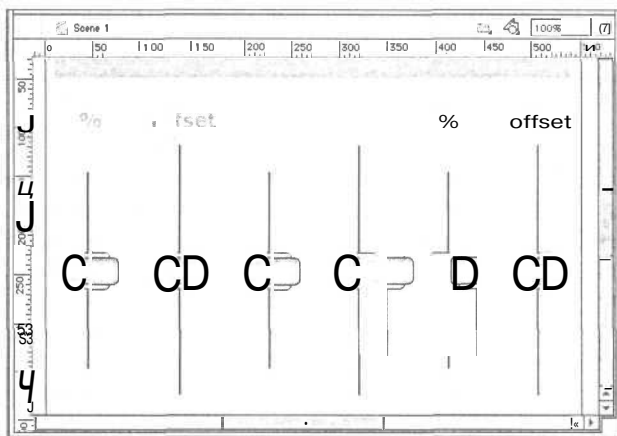
В главе 9 вы изучили использование панели Effects (Эффекты) для задания эффекта динамического изменения цветов. Теперь рассмотрим изменение цвета посредством сценариев ActionScript. Используя один из предопределенных объектов Flash, объект `Color`, и его методы, можно динамически изменять цвета в фильме.

Экземпляр объекта `Color` создается функцией-конструктором. После этого появляется возможность обрабатывать информацию о цвете посредством следующих методов этого объекта.

- ▶ `getRGBO` возвращает числовое значение цвета, заданное последней командой `setRGBO`.
- ▶ `setRGBO` задает шестнадцатеричное значение цвета для указанного объекта `Color`.
- ▶ `getTransformO` возвращает параметры изменения цвета, которые были заданы последней командой `setTransformO`.
- ▶ `setTransformO` задает параметры изменения цвета для объекта `Color`.

## Создание микшера цветов

Этот пример иллюстрирует применение объекта `Color` для динамического изменения цвета. Чтобы задать такие изменения, мы используем несколько регуляторов для увеличения и уменьшения долей компонентов результирующего цвета.



 Файлы `colorMixer.fla` и `colorMixer.swf`, содержащие полностью завершенный пример, находятся в каталоге Chapter 16 на прилагаемом компакт-диске. Если вы хотите самостоятельно повторить все описанные ниже действия, откройте файл `colorMixerCD.fla`.

Создадим набор регуляторов, которые изменяют компоненты цвета.

1. Откройте файл `colorMixerCD.fla` и выберите команду `File ▸ Save As`, чтобы сохранить его на жестком диске.

Вы увидите, что часть черновой работы уже сделана, главным образом для экономии время и предоставления вам возможности сосредоточиться на важных аспектах этого урока. Все регуляторы уже готовы и размещены в области действия фильма.

В этом фильме используются регуляторы двух видов: одни изменяют значения в интервале от -100 до 100, а другие — от -255 до 255. Это вызвано тем, что при работе с методом `setTransform()` объекта `Color` нужно задавать значения параметров в этих пределах.

2. Задайте для каждого регулятора имя экземпляра. Выберите крайний левый регулятор и активизируйте команду `Window ▸ Properties (Cmd/Ctrl+F3)`, чтобы открыть панель свойств. Введите имя `Ra` в поле имени экземпляра. Остальные регуляторы нужно назвать `rb`, `ga`, `gb`, `ba` и `bb` (слева направо). Сохраните фильм.
3. Активизируйте экземпляр `ga` регулятора (крайне левый) и откройте панель `Actions`. Введите следующие строки:

```
onClipEvent(enterFrame){
 raColor=new Color(_root.meter);
 raTransform=new Object();
```

```

 raTransform.ra=((this.knob100._y)*-1);
 trace("ra:"+raTransform.ra);
 raColor.setTransform(raTransform);
}

```

Этот сценарий выполняется при воспроизведении каждого кадра фильма. Поскольку наш фильм содержит один кадр, Flash воспроизводит его снова и снова. Во второй строке сценария используется функция-конструктор для создания нового объекта `Color` с именем `raColor`. Следующая инструкция создает новый объект с именем `raTransform`. Он понадобится позже для метода `setTransform()`.

Метод `setTransform()` может принимать до восьми параметров, определенных посредством объекта `raTransform`. Для этого экземпляра регулятора мы будем использовать только один из них, `ra`, который представляет долю красного цвета в процентах. Значение этого параметра вычисляется выражением `((this.knob100._y)*-1)` в зависимости от позиции рукоятки регулятора. Если вы посмотрите на текст сценария, связанного с экземпляром объекта `knob100`, то увидите, что диапазон возможных значений этого объекта составляет от `-100` до `100`. В выражении эти значения умножаются на `-1`, чтобы при движении рукоятки регулятора вверх увеличивались положительные значения.

Функция `trace()` выводит вычисленное значение в окне `Output`. В последней строке сценария используется метод `setTransform()`, который получает в качестве аргумента объект `raTransform` и изменяет процентную долю красной составляющей цвета экземпляра `raColor` объекта `Color`.

4. Теперь, после окончания работы, вы можете скопировать этот сценарий для остальных регуляторов. Выделите действия, которые вы только что ввели, и скопируйте их в буфер обмена (клавиши `Cmd/Ctrl+C`). Затем активизируйте экземпляр `da`, откройте панель `Actions` и вставьте эти действия, используя клавиши `Cmd/Ctrl+V`. Замените в тексте сценария все вхождения `ra` на `da`. Произведите такие же действия для экземпляра `ba` регулятора.

## СОВЕТ

Для быстрого перехода от текста одного сценария к тексту другого используйте раскрывающийся список сценариев панели `Actions`.

Таким образом с каждым регулятором связывается один и тот же сценарий, но поскольку имена объектов изменены, каждый регулятор управляет своим параметром объекта в методе `setTransform()`. Чтобы узнать больше об этом методе, обратитесь к приложению Б.

5. Осталось создать сценарии для оставшихся трех регуляторов. Вы также можете скопировать имеющийся сценарий регулятора `ra`, но внести в него дополнительные модификации. Скопируйте этот сценарий в панель `Actions` для регулятора `rb`. Замените в тексте сценария все вхождения `ra` на `rb`. Затем измените строку

```
raTransform.ra=((this.knob100._y)*-1);
```

на строку

```
rbTransform.rb=((this.knob255._y)*-2);
```

В этом регуляторе используется другой объект рукоятки регулятора, knob255, для генерирования других числовых значений. В выражении значение вертикальной координаты  $_u$  экземпляра объекта knob255 умножается на -2. Делается это потому, что параметры ga, da и ba могут принять значение в интервале от -100 до 100, а параметры gb, gb и bb могут изменяться в диапазоне между -255 и 255. Для объекта же knob255 значения находятся в диапазоне от -127,5 до 127,5 и должны быть удвоены, чтобы получить диапазон от -255 до 255.

После внесения изменений скопируйте этот сценарий в буфер обмена и закройте панель Actions.

6. Выберите экземпляр gb регулятора, откройте панель Actions и вставьте скопированный сценарий. Измените все имена gb на gb. Сделайте то же самое для экземпляра bb регулятора.

Все необходимые сценарии созданы. Сохраните фильм, а затем выберите команду Control ▶ Test Movie для проверки его работы. При перетаскивании ручек регуляторов вверх и вниз вы увидите, что цвета изменяются.

## Разработки

Выставка African Voices была организована Смитсоновским институтом, чтобы представить широкой аудитории образ жизни обитателей Африки и ее богатства. Выставка «демонстрирует разнообразие, динамизм и глобальное влияние народов Африки и их культур» и содержит информацию о людях, семье и профессиональной жизни.



Рис. 16.9. Web-узел African Voices

Хотя выставка находится в национальном Музее естествознания Вашингтона, округ Колумбия, она также доступна через Интернет по адресу [www.mnh.si.edu/africanvoices/](http://www.mnh.si.edu/africanvoices/). Комплексный Web-узел African Voices, использующий Flash и ActionScript для динамического представления огромного количества информации в удобном виде, был разработан Смитсоновским институтом и компанией Terra Incognita.

## Резюме

В этой главе было представлено большинство наиболее полезных элементов языка ActionScript. Вы узнали, как увеличить интерактивность фильма с помощью альтернативных контроллеров и как применять клавиатуру для взаимодействия с пользователем. Мы рассказали также о глубине и многосторонности фильмов, содержащих много временных шкал и клипов.

Для управления клипами в анимации Flash или интерактивном фильме вы можете использовать много различных свойств и действий ActionScript. Некоторые из них применяются для непосредственного управления воспроизведением, другие для изменения внешнего вида клипа или фильма и создания интересных графических эффектов. В следующей главе мы рассмотрим дополнительные приемы манипулирования данными и управление некоторыми невидимыми компонентами Flash-фильма посредством сценариев ActionScript.

## Глава 17

# Управление вводом-выводом данных

Одна из наиболее мощных функций программы Flash — получение пользовательских данных от компьютера зрителя (Flash-фильм в браузере) и отображение динамического содержимого, которое передается с сервера. Например, можно получить от вашей аудитории информацию, вводимую в формы, или отображать на узле постоянно изменяющийся список новостей. В таких ситуациях Flash переключается из статического состояния в динамическое, что делает эту программу еще более универсальной и мощной.

Честно говоря, наибольшее внимание Flash уделяет динамическому содержимому (которое пользователь видит в Web-браузере) сетевых приложений. Внутренние операции (анализ, сбор и обработка информации, полученной из внешнего источника) производятся благодаря технологиям управления базами данных. Обсуждение операций ввода-вывода данных в среде Flash может легко выйти за рамки этой главы. Поэтому мы лишь слегка коснемся внутренних операций (ими нельзя полностью пренебречь) и сосредоточим внимание на динамическом содержимом. Этой информации будет достаточно для создания динамических сетевых приложений, использующих в работе базы данных.

В главе освещаются такие темы:

- ▶ приложения, использующие базы данных;
- ▶ создание динамического внешнего интерфейса;
- ▶ представление динамического содержимого;
- ▶ генерирование содержимого из текстовых файлов.

## Приложения, использующие базы данных

Еще не так давно Интернет был в значительной степени статичным. Пользователи работали со страницами, которые содержали предустановленный, неизменный объем информации. С появлением HTML все изменилось. Благодаря ему стала возможной разработка приложений, использующих в процессе работы базы данных. Посетители перестали быть пассивными: на узле, где происходит постоянное обращение к базе данных, они могут менять страницы (содержимое или дизайн) в режиме реального времени.

Что же представляет собой динамическое Web-приложение? Оно состоит из двух частей: внешнего и внутреннего интерфейсов. Внешний интерфейс — это часть приложения в окне браузера, с которым пользователь работает. Кнопки, флажки, текстовые поля, всплывающие меню — все это элементы внешнего интерфейса. Внутренний интерфейс состоит из трех компонентов: Web-сервера, базы данных и сценария, благодаря которому все компоненты работают как единое

целое. Web-сервер выполняет несколько функций: предоставляет компьютеру пользователя доступ к HTML-документам и обрабатывает код, управляющий работой приложения.

База данных (начиная с Microsoft Access и заканчивая SQL) содержит информацию, которая динамически поставляется пользователю в соответствии с его запросами. Например, при создании сайта, где пользователь должен зарегистрироваться перед получением доступа, следует сформировать базу данных для хранения данных о пароле и имени пользователя.

---

#### ПРИМЕЧАНИЕ

SQL — это язык форматирования структурных запросов к базам данных.

---

Сценарий — это «душа» приложения, использующего базу данных. Существует много различных способов создания сценария, запускающего приложение. Наиболее интересные рассмотрены в разделе «Представление динамического содержимого» этой главы.

## Создание динамического внешнего интерфейса

Среда Flash идеально подходит для формирования внешнего интерфейса сетевого приложения. Далее подробно описываются три этапа этого процесса. А пока охарактеризуем их вкратце. Первый этап — создание визуальных элементов интерфейса, которые будут использоваться для ввода или отображения данных. Второй и третий этапы взаимосвязаны. На них осуществляются передача данных серверу и динамическое отображение данных в Flash.

## Использование текстовых полей для сбора и отображения динамической информации

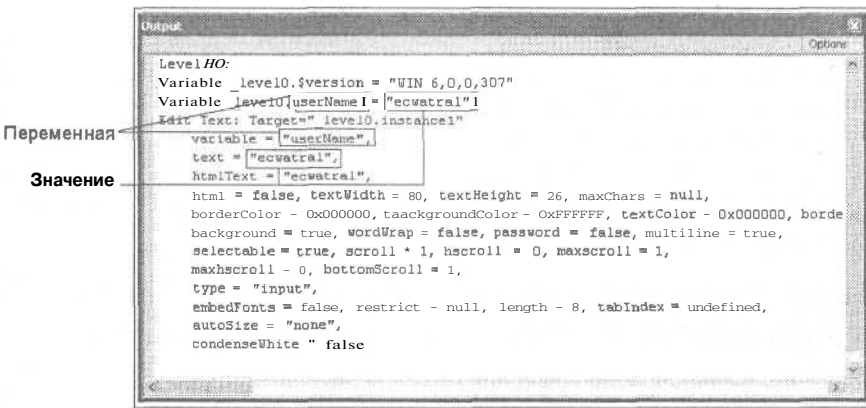
В главе 5, посвященной работе с текстом, говорилось о том, что текст в среде Flash создается в блоках, которые называются текстовыми полями. Существует три типа текстовых полей: расширяемые, фиксированные и текстовые поля ввода. Соответственно, текст также делят на три категории: статический, динамический и вводимый. Управление текстовыми объектами каждого типа осуществляется с помощью панели Properties (Window ▶ Properties). Статический текст используется во Flash по умолчанию, но неотъемлемой частью любого внешнего интерфейса динамического приложения является текст двух других типов — вводимый и динамический. Далее мы рассмотрим возможности интеграции каждого типа текста в Flash-приложение.

### Сбор данных с помощью полей ввода

Текстовое поле ввода можно сравнить с контейнером, в который пользователь добавляет информацию: имя, любимый цвет и т. д. Любой введенный текст присваивается как значение некоторой переменной, которую определяете вы. Затем это значение передается на сервер или обрабатывается во Flash. Например, если

с текстовым полем ввода связана переменная `userName`, то все, что пользователь введет в это поле (имя и другие данные), станет значением переменной `userName`. Любой тип текста во Flash создается с помощью инструмента Text. Давайте выполним упражнение.

1. Создайте новый документ.
2. Активизируйте инструмент Text на панели Tools.
3. Загрузите панель Properties, вызвав одноименную команду меню Window.
4. Выберите пункт Input Text в списке Text Type.
5. В области действия создайте поле для ввода текста.
6. Далее требуется установить ряд параметров. Наиболее интересный на данном этапе — Var. Если вы хотите, чтобы созданное текстовое поле ввода функционировало должным образом, укажите переменную в поле Var. Для того чтобы поле было видно пользователю, активизируйте флажок Show Border around Text.
7. Чтобы протестировать поле, сохраните документ. Затем вызовите команду Control ▶ Test Movie (или Control ▶ Test Scene). После этого введите в поле некоторое значение и активизируйте команду Debug ▶ List Variables. В окне Output отобразится такая информация:



Получение данных, введенных пользователем, и передача их серверу — это часть процесса (охватывающего действия `getUrl()` и `loadVariables()`), который рассматривается в разделе «Обмен данными с сервером».

## Отображение динамических данных

Динамический текст практически полностью подобен вводимому тексту. Единственное (и чрезвычайно важное) отличие заключается в том, что данные перемещаются в другом направлении — с сервера на компьютер пользователя.

Содержимое динамического поля генерируется на основе некоторого источника — базы данных или текстового файла, размещенного на сервере. Это подготавливает богатую почву для реализации всевозможных интересных идей. Например, можно создать постоянно изменяющийся список ваших любимых шуток или афоризмов.



Рассмотрим процесс создания динамического текста.

1. Создайте новый документ.
2. Активизируйте инструмент Text на панели Tools.
3. Загрузите панель Properties посредством одноименной команды меню Window.
4. В области действия создайте текстовое поле. Удостоверьтесь, что размер поля соответствует ожидаемому объему текста.
5. Выберите элемент Dynamic Text в списке Text Type на панели Properties.
6. В поле Var присвойте переменной уникальное имя. Если не назначить переменную, динамический текст не будет отображаться.

В следующем разделе вы узнаете, как использовать в фильме динамический текст, взятый из внешних файлов.

## Обмен данными с сервером

Для работы с данными программе Flash необходим сценарий ActionScript, процесс создания которого послужит следующей темой нашего разговора. Давайте познакомимся с действиями `getURL()`, `loadVariables()` и `loadMovie()`.

### Действие `getURL()`

В процессе создания динамических Flash-приложений действие `getURL()` позволяет пересылать данные в файл и отображать результаты в новом окне, отдельном фрейме или существующем окне. Отрицательный момент при использовании действия `getURL()` состоит в том, что оно не может возвращать информацию в текущее окно браузера, где воспроизводится Flash-фильм. Если вы укажете Flash вернуть информацию в текущем окне, Flash-фильм будет заменен ею. В качестве альтернативы можно задать возврат информации в новом окне, используя аргумент `blank` (далее вы узнаете, как это сделать), что является не лучшим вариантом, если вы хотите ограничить навигацию пользователя в пределах Flash-фильма. Однако это единственное решение проблемы в том случае, когда применяется действие `getURL()`.

В следующем примере пошагово рассматривается процесс использования действия `getURL()`. Переменная пересылается в вымышленный ASP-файл `name.asp`, находящийся в том же каталоге, что и фильм:

1. Создайте новый документ.
2. Активизируйте инструмент Text.
3. Откройте панель Properties, вызвав одноименную команду меню Window.
4. Создайте текстовое поле в области действия.
5. Выберите пункт Input Text в списке Text Type на панели Properties.
6. Чтобы сделать текстовое поле видимым, нажмите кнопку Show Borders Around Text.
7. Введите имя `userName` в поле Var.
8. Создайте кнопку и разместите ее в области действия рядом с текстовым полем ввода. (Если вы не помните, как создается данный символ, вернитесь к главе 13.)

9. Выберите символ **Button** в области действия и откройте панель **Actions** (**Window > Actions**).
10. Переведите панель **Actions** в режим **Normal** посредством команды **Normal Mode** меню **Options**.
11. Щелкните на категории **Actions** в области элементов **ActionScript** (левая часть панели **Actions**).
12. Выберите подкатегорию **Browser/Network**.
13. Выполните двойной щелчок на имени действия `getURL()`, чтобы добавить его в область отображения сценариев.
14. Укажите имя файла в поле **URL** — `name.asp`. В тех случаях, когда информация передается на Web-страницу, вводится полный URL-адрес этой страницы. Если файл, которому передается информация, находится в другом каталоге, необходимо указать полный путь к нему. Сразу после ввода имя файла или URL добавляется в область отображения сценариев.
15. Далее следует задать, в каком окне выводить информацию. Откройте список **Window** и выберите в нем один из элементов:
  - ▷ `_blank` — информация выводится в новом окне (поверх того, в котором находится Flash-фильм);
  - ▷ `_self` — информация выводится в текущем фрейме активного окна;
  - ▷ `_parent` — информация выводится в родительском фрейме текущего окна;
  - ▷ `_top` — информация выводится в верхнем фрейме текущей страницы.
16. Следующий шаг — выбор метода передачи информации. Список возможных вариантов — **Variables** — расположен непосредственно под списком **Window**:
  - ▷ **GET** — информация передается как строка запроса.
  - ▷ **POST** — информация передается с помощью буфера. Этот метод предпочтительнее при пересылке большого объема информации.
17. Работа над сценарием подошла к концу. Закройте панель **Actions**.

При нажатии кнопки, с которой связан созданный вами сценарий, Flash-фильм примет текст, введенный в текстовое поле ввода, и перешлет его в файл `name.asp`. Далее ASP-файл примет информацию (которая передается как комбинация переменная/значение) и обработает ее. Как именно — определяется непосредственно в ASP-файле (который необходимо создать).

---

#### ПРИМЕЧАНИЕ

В раскрывающемся списке **Window** выбирать элемент необходимо только тогда, когда создан ASP-файл, возвращающий данные для отображения в окне браузера.

---

Например, можно создать ASP-файл, который получает значение переменной `userName` и помещает его в базу данных. Существует еще один вариант. Вместо того чтобы помещать значение в базу данных, можно переслать информацию по электронной почте в ваш почтовый ящик.

## Действия `loadVariables()` и `loadVariablesNum()`

Действие `getUrl()` подходит в тех случаях, когда нужно отобразить информацию за пределами Flash-фильма. Но что делать, если она должна быть выведена непосредственно в фильме? Для решения таких задач надлежит применять действия `loadVariables()` и `loadVariablesNum()`. Эти действия возвращают информацию, которая добавляется либо в клип (`loadVariables()`) Flash-фильма, либо на уровень фильма главной временной шкалы (`loadVariablesNum()`). Они также могут обновлять значения переменных в фильме.

На данном этапе нет необходимости детально рассматривать процесс использования этих действий. После изучения внутреннего механизма работы динамического Web-приложения (будь то Flash или HTML) вы способны самостоятельно создать фильм, используя действие `loadVariablesNum()` для загрузки динамического содержимого из набора текстовых файлов.

## Использование действий `loadMovie()` и `loadMovieNum()` для динамической загрузки JPEG-файлов

Мы рассмотрели, как применять действия `getUrl()`, `loadVariables()` и `loadVariablesNum()` для получения и отображения текстовой информации. Однако неизбежны ситуации, когда наряду с текстовыми потребуются вывести данные других типов. Для загрузки изображений используются действия `loadMovie()` и `loadMovieNum()` (оба являются действиями объекта `MovieClip`). С их помощью можно динамически загрузить изображения в формате JPEG, которые находятся на Web-сервере, в клип (`loadMovie()`) или на уровень фильма главной временной шкалы (`loadMovieNum()`).

### ПРИМЕЧАНИЕ

В разделе сделан акцент на использовании действий `loadMovie()` и `loadMovieNum()` для динамического отображения содержимого JPEG-файлов в Flash-фильме. Следует отметить, что эти действия подходят и для загрузки SWF-файлов в фильм.

Ниже описан процесс динамической загрузки JPEG-изображения с использованием действия `loadMovie()` или `loadMovieNum()`.

1. Выберите кадр, кнопку или клип, с которыми вы хотите связать действие `loadMovie()` или `loadMovieNum()`.
2. Загрузите панель Actions, вызвав одноименную команду меню Window.
3. Отобразите содержимое категории Actions в области элементов ActionScript.
4. Перейдите в подкатегорию Browser/Network.
5. Выполните двойной щелчок на имени действия `loadMovie()`, чтобы добавить его в область отображения сценариев.
6. В поле URL, расположенном в области отображения параметров, укажите имя загружаемого JPEG-файла и путь доступа к нему. Помните: если изображение находится в том же каталоге, что и Flash-фильм, достаточно ввести только имя файла, а если в другом каталоге, надлежит указать полный путь, чтобы Flash-фильм мог определить его местоположение.

7. Если необходимо загрузить изображение на определенный уровень фильма, выберите элемент `Level` в списке `Location` и укажите номер уровня справа от списка `Location`. Чтобы загрузить изображение в клип, выберите элемент `Target` в списке `Location` и укажите имя клипа справа от списка `Location`.
8. По окончании закройте панель `Actions`.

Если загрузка осуществляется в главную временную шкалу, область отображения фильма будет автоматически *отмасштабирована*, чтобы соответствовать площади, занимаемой JPEG-изображением. Поэтому, вероятно, лучше загружать JPEG-изображение в клип. Тогда автоматическое масштабирование будет произведено для клипа, а главный фильм не изменится.

#### ПРИМЕЧАНИЕ

Если клип, в который загружается JPEG-изображение, не находится в фильме главной временной шкапы, необходимо указать путь, используя точечный синтаксис.

Существует два важных момента, которые нужно учитывать в процессе применения действий `loadMovie()` и `loadMovieNum()` для динамической загрузки JPEG-изображения. Во-первых, при использовании действия `loadMovie()` все содержимое указанного клипа заменяется загруженным JPEG-изображением. Во-вторых, на определенном уровне Flash-фильма может содержаться только одно JPEG-изображение. Таким образом, если загрузить второе JPEG-изображение на уровень `_level1`, первое будет автоматически заменено им.

Подробная информация об уровнях Flash-фильма приведена в главе 16.

## Представление динамического содержимого

Приложения, использующие в работе базу данных, состоят из внешнего и внутреннего интерфейсов. Вы уже знаете, как создать внешний интерфейс динамического приложения во Flash. Теперь мы приступим к изучению внутреннего интерфейса, основные компоненты которого — сервер, база данных и сценарий.

Вам известна процедура передачи данных на сервер. В случае применения действий `getUrl()` и `loadVariables()` в сценарии указывается местоположение получателя информации (в поле `URL` на панели `Actions`). Независимо от того, используется ASP-файл или CGI-сценарий, вы информируете Flash о том, куда следует переслать данные.

В следующих разделах дан обзор сценариев различных типов и серверных технологий. Этот материал должен помочь вам в выборе технологии, которая в наибольшей степени подходит для решения рабочих задач.

### ASP

ASP (Active Server Pages) — это среда разработки на стороне сервера сценариев, используемых для создания динамических Web-страниц или встроенных Web-приложений. Страницы ASP являются файлами, содержащими теги HTML,

текст и команды сценариев. Они позволяют вызывать компоненты модели COM для решения таких задач, как подключение к базе данных и выполнение деловых расчетов. Благодаря ASP перед пользователями открываются возможности добавления интерактивных элементов на Web-страницы и разработки Web-приложений, которые используют HTML-страницы как интерфейс.

Одно из значительных достоинств ASP заключается в возможности обеспечить доступ к базам данных для записи или выборки информации. Информация извлекается из базы данных с учетом того, какой пользователь зашел на узел. Как следствие, вы можете по-разному представлять страницу (варьировать шрифты, цвета и т. д.), изменяя ASP-, а не HTML-код. ASP, подобно многим серверным технологиям, позволяет извлекать информацию из базы данных для непрерывного заполнения Web-страницы или Flash-фильма динамическими данными.

Хотя ASP-код по-прежнему может быть написан в любом текстовом редакторе (например, в Notepad), процесс его создания стал довольно сложным. Это обусловило появление множества оболочек, призванных помогать в разработке ASP-кода. Наибольший интерес представляют Macromedia Dreamweaver и UltraDev. Благодаря им можно создать сложную ASP-программу с помощью визуальных инструментов разработки.



Демонстрационные версии Macromedia Dreamweaver и UltraDev находятся на прилагаемом компакт-диске.

Технология ASP может работать с любыми базами данных, соответствующими стандарту ODBC.

## ColdFusion

ColdFusion, разработка фирмы Allaire, которая в 2001 году была приобретена компанией Macromedia, представляет собой технологию построения динамических приложений, работающих с базами данных, которая использует уникальный язык сценариев CFML (ColdFusion Markup Language) вместе со специализированным сервером ColdFusion Server для создания мощных динамических сетевых приложений. CFML подобен HTML и довольно прост в изучении и применении.

Когда пользователь запрашивает документ ColdFusion (который называется шаблоном и имеет расширение CFM), Web-сервер передает файл сначала ColdFusion Server в Интернете, а затем браузеру пользователя. ColdFusion Server проверяет шаблон, интерпретируя любой внедренный CFML-код. После этого ColdFusion Server генерирует шаблон на основе CFML-кода и возвращает на Web-сервер файл, который будет отображен в браузере пользователя.

Рассмотрим небольшой пример. Предположим, вы создали шаблон ColdFusion с необходимым CFML-кодом для составления списка книг, которые читаете. Когда пользователь загрузит файл (помните, что он называется шаблоном) в браузере, сервер назначит ему расширение CFM и отправит запрос ColdFusion Server. Получив запрос, ColdFusion Server извлечет список книг из регулярно обновляемой базы данных, подставит данные в соответствующие части шаблона (определяются посредством CFML) и вернет файл Web-серверу, который перенаправит его браузеру пользователя.

Для корректной работы ColdFusion требуется ColdFusion Server, что отличает его от ASP, которому нужен только Web-сервер определенного типа.

#### ПРИМЕЧАНИЕ

Подробная информация о ColdFusion предоставляется по адресу [www.macromedia.com/coldfusion](http://www.macromedia.com/coldfusion).

Как и в случае с ASP, шаблон ColdFusion можно задать как URL-адрес для действия `getUrl()` или `loadVariables()`. Когда информация (фактические данные или запрос на предоставление данных) передается CFM-документу, ColdFusion обрабатывает запрос и возвращает данные Flash, которые будут отображены определенным способом (непосредственно в Flash-фильме или в другом окне).

Существует множество способов генерирования CFML-кода. Его можно написать с нуля, а также создать с помощью Macromedia Dreamweaver UltraDev или ColdFusion Studio.

## PHP

В 1994 Расмас Лердорф разработал Perl-сценарии (Practical Extraction and Reporting Language), позволившие контролировать пользователей, которые обращались к его Web-узлу. Мир узнал об интересном изобретении Лердорфа, и люди начали просить разрешения использовать его сценарии на своих узлах. Вскоре Лердорф выпустил программный пакет под названием РНРТ.

К 1995 РНР (полное название Personal Home Page Tools трансформировалось в Hypertext Preprocessor) получил широкое распространение. Популярность РНР стимулировала Расмаса разработать механизм создания сценариев и дополнительные сценарии для обработки данных, вводимых в формы.

Третья версия РНР включала уже намного более эффективный механизм создания сценариев. Синтаксис был изменен так, что приблизил РНР к C/C++, а не только к Perl, который не был объектно-ориентированным языком программирования. Чтобы сторонние разработчики имели возможность модифицировать РНР, создавая собственные модули, РНР3 был снабжен API (Application Programming Interface — прикладной программный интерфейс).

Результатом полной модернизации стал РНР4. У разработчиков появилось намного больше возможностей управления своими РНР-кодами. Повысилась эффективность интегрирования сторонних РНР-модулей посредством API.

Среди преимуществ РНР прежде всего следует выделить то, что РНР-код можно записать с помощью простого текстового редактора типа Notepad, SimpleText или Emacs. Подобно многим серверным технологиям создания сценариев (включая все рассмотренные в этой главе), РНР способен отображать информацию из любой базы данных на сервере, которая совместима с ODBC. Пакет РНР может быть установлен на большинстве Web-серверов (Windows или Unix). Откровенно говоря, РНР работает надежнее и быстрее, чем ASP и ColdFusion. Все эти достоинства существенны, но основным является свободное распространение.

Вы можете указать PHP-файл (который имеет расширение .PHP) в качестве URL для действий `getUrl()` и `loadVariables()`. Как только информация (фактические данные или запрос на их предоставление) поступит к PHP-файлу, сервер PHP обработает и возвратит данные Flash для отображения указанным вами способом (в Flash-фильме или отдельном окне).

---

#### СОВЕТ

Подробная информация о PHP предоставляется на Web-узлах, расположенных по адресам [www.php.net](http://www.php.net), [www.phpbuilder.com](http://www.phpbuilder.com) и [www.opensource.org](http://www.opensource.org).

---

## CGI

Язык сценариев CGI (Common Gateway Interface) появился прежде, чем у кого-либо возникла мысль о динамических приложениях, использующих в работе базы данных. По сути CGI является не языком программирования, а технологией развития серверных программ, что расширяет функциональные возможности HTML-страниц (и Flash).

Perl — один из самых популярных языков программирования для разработки CGI-кода. Существуют, конечно, и другие средства создания CGI-кода: C/C ++, Visual Basic, Applescript, Unix Shell и Tcl.

CGI поддерживает работу с действиями `getUrl()` и `loadVariables()`. Данные, поступающие из Flash, обрабатываются определенным сценарием CGI, возвращаются и отображаются указанным вами способом.

---

#### СОВЕТ

Для получения дополнительной информации о CGI посетите Web-узлы [www.script-search.com](http://www.script-search.com) и [www.cgi-resources.com](http://www.cgi-resources.com).

---

## Генерирование содержимого из текстовых файлов

После изучения внешнего и внутреннего интерфейсов приложения, использующего в работе базу данных, вы готовы к созданию Flash-фильма, в котором отображается динамическая информация. В следующем упражнении вы на практике проверите теоретические сведения, изложенные в предыдущих разделах. Ожидаемый результат — Flash-фильм, в котором отображается информация из текстовых файлов. Заметим, что в нем не будет динамики, потому что текстовые файлы содержат статическую информацию. Главное в упражнении то, что вы, используя ActionScript, создадите Flash-фильм, при воспроизведении которого демонстрируется информация из различных источников. А вот динамику обеспечила бы одна из описанных выше технологий.

В процессе выполнения упражнения будут созданы четыре кнопки, каждая из которых предназначена для загрузки в Flash-фильм содержимого соответствующего текстового файла.



Текстовые файлы, используемые в следующем примере, находятся в каталоге Chapter 17 на прилагаемом компакт-диске. Здесь же размещен файл `quotes.swf`, содержащий конечный результат. Конечно, вместо указанных можно применять любые текстовые файлы.

1. Создайте на жестком диске каталог `Flash Quote Project`. Скопируйте в него все текстовые файлы из каталога Chapter 17, находящегося на прилагаемом компакт-диске.
2. Теперь создайте новый файл, вызвав команду `New` меню `File`. Сохраните файл под именем `quotes` в каталоге `Flash Quote Project`.

### СОВЕТ

Если вы скопировали файл `quotes fla` из каталога Chapter 17, то сохраненный файл будет записан поверх скопированного. Чтобы избежать этого, присвойте файлу другое имя.

3. Переименуйте существующий слой в `Dynamic Text`.
4. Активизируйте инструмент `Text`.
5. Загрузите панель `Properties (Window ▶ Properties)`.
6. На панели `Properties` выберите элемент `Dynamic Text` в списке `Text Type`.
7. Удостоверьтесь, что в списке `Line Type` отмечен пункт `Multiline`.
8. Выберите первый кадр слоя `Dynamic Text` и с помощью инструмента `Text` создайте в области действия поле шириной 10 сантиметров (4 дюйма) и высотой 7 сантиметров (2.75 дюйма). Это поле будет служить «контейнером», в котором Flash отобразит содержимое каждого текстового файла. Очень важный момент: введите имя переменной `quoteField` в поле `Var` на панели `Properties`.  
В начале каждого текстового файла можно увидеть выражение `quoteField=`. Эта переменная связывает текстовый файл с полем динамического текста, которое вы только что создали.
9. Установите параметры шрифта. Выделите динамическое поле и на панели `Properties` выполните такие установки: шрифт — `Helvetica` или `Arial`, цвет — черный, размер — 12, стиль — полужирный курсив.
10. Следующий шаг — создание интерактивных элементов управления, которые будут вызывать содержимое из каждого текстового файла. Вы можете создать четыре кнопки (`Homer`, `Plato`, `Frost` и `Melville`) или воспользоваться готовыми кнопками из файла `quotes fla`, находящегося на прилагаемом компакт-диске. Во втором случае используйте команду `File > Open as Library`.
11. Создайте четыре новых уровня с названиями, как у кнопок (`Homer`, `Plato`, `Frost` и `Melville`). Перетащите каждую кнопку на соответствующий уровень. Удостоверьтесь, что кнопки размещены равномерно ниже поля динамического текста.
12. Свяжите сценарий `ActionScript` с каждой из кнопок так, чтобы при щелчке на кнопке в динамическое поле загружался соответствующий текст. Выберите кнопку `Homer` и откройте панель `Actions (Window ▶ Actions)`.
13. Убедитесь в том, что активен режим `Normal`. В области элементов `ActionScript` (левая часть панели `Actions`) щелкните на категории `Actions` для отображения ее содержимого.

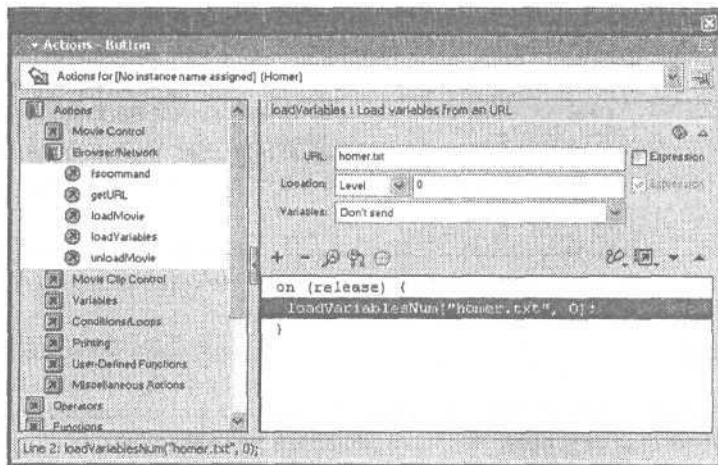


14. Перейдите в подкатегорию Browser/Network.
15. Выполните двойной щелчок на имени действия `loadVariables()`, чтобы добавить его в область отображения сценариев.
16. Укажите аргументы для действия `loadVariables()`. В поле URL введите имя текстового файла (в нашем случае — `homer.txt`), содержимое которого должно быть загружено в поле динамического текста, когда пользователь щелкнет на кнопке, с которой связан сценарий. Если текстовый файл находится в том же каталоге, что и Flash-фильм, достаточно задать имя файла, в противном случае надлежит указать путь к нему.
17. В списке Location выберите элемент Level. Введите значение 0 в поле справа, чтобы программа Flash загружала текстовый файл на нулевом уровне фильма. Обратите внимание на то, что при выборе элемента Level в списке Location действии `loadVariables()` автоматически заменяется действием `loadVariablesNum()`.

### ПРИМЕЧАНИЕ

Flash сохраняет информацию на различных иерархических слоях, или *уровнях*. В сценарии, связанном с клипом, можно обратиться к переменным, находящимся на том же уровне, что и клип. Если переменная содержится в клипе иного уровня, в сценарии следует указать ее местоположение.

Ниже показано, как должна выглядеть панель Actions на данном этапе.



18. Повторите шаги 12–17, чтобы связать с текстовым файлом каждую кнопку.
19. Проверьте фильм посредством команды Control ▶ Test Movie или Control ▶ Test Scene.

## Разработки

Популярность Flash во многом обусловлена возможностью создания динамических приложений. Среди самых интересных приложений — система сбора информации, поступающей от пользователей, в реальном времени. Один из признанных лидеров в данной сфере — Web-узел OpinionLab ([www.opinionlab.com](http://www.opinionlab.com)).

Система OpinionOnline включает два инструментальных средства. Первое, основанное на DHTML, обеспечивает непрерывный сбор пользовательской информации о Web-страницах. Второе средство работает совместно с Macromedia Generator и предназначено для статистической обработки информации, собранной первым средством, а также ее представления;

Работая в тандеме, эти два инструментальных средства образуют мощную систему, позволяющую производить мгновенный маркетинговый анализ (процесс, который обычно требует значительных временных затрат).



Рис. 17.1. Web-узел OpinionLab

## Резюме

Программа Flash может передавать данные серверным приложениям, а также получать информацию от серверных приложений и отображать ее.

Web-приложение состоит из внешнего и внутреннего интерфейсов. В главе подробно описаны приемы разработки каждого из них. Значительное внимание уделялось важнейшим действиям, используемым при создании Flash-приложений, которые в процессе работы обращаются к базе данных: `getUrl(0,1 loadVariables()` и `loadVariablesNum()`.

Ценность главы повышает обзор серверных технологий ASP, ColdFusion, PHP и CGI.

Теория была закреплена на практике: главу завершило упражнение, посвященное созданию Flash-фильма, при воспроизведении которого загружается содержимое внешних текстовых файлов.

## Глава 18

# Отладка сценариев ActionScript

Если вы внимательно изучили предыдущие главы, то обладаете достаточным багажом знаний об ActionScript, поскольку мы рассмотрели множество терминов, методов, синтаксических правил и команд этого языка. Скорее всего, вы уже пытались применять сценарии ActionScript в своих фильмах и при этом наверняка столкнулись с проблемами.

Причины появления ошибок в сценариях могут быть самыми разными — от неправильного употребления конструкций и элементов языка до обычных опечаток. Вам, видимо, пока было сложно определить, почему сценарий генерирует ошибку. Теперь мы расскажем, как избегать таких ситуаций и решать проблемы в случае их появления. В этой главе рассмотрены следующие темы:

- ▶ правила разработки сценариев ActionScript;
- ▶ методы поиска ошибок;
- ▶ использование отладчика Flash;
- ▶ работа с окном Output.

## Правила разработки сценариев

Даже самый опытный программист не застрахован от ошибок при выполнении рутинных операций. Однако профессионал отличается тем, что способен верно оценить ситуацию и найти решение проблемы. Если вы еще не профессионал, не расстраивайтесь! Как показывает практика, при надлежащей постановке процесса планирования и составления сценариев многих проблем можно избежать. Постарайтесь придерживаться следующих правил.

- ▶ Используйте соответствующие имена переменных. Переменные хранят значения, свойства, метки кадров, то есть всю различного рода информацию, какая есть в фильме. При инициализации выбирайте переменной имя, которое соответствует ее назначению. Это имя должно быть идентификатором. Более подробная информация о переменных содержится в главе 15.
- ▶ Следите за правильностью использования элементов ActionScript. Часть элементов ActionScript предназначена только для чтения. Это означает, что их можно проверить, но нельзя изменить. Для некоторых идентификаторов и конструкций важен регистр символов или наличие определенных символов. Если у вас возникли сомнения в правильности применения элемента, обратитесь к приложению Б этой книги либо командой Window ▶ Reference или нажатием клавиш Shift+F1 вызовите панель Reference (Руководство).
- ▶ Чаще тестируйте фильмы и сценарии. Разрабатывая фильм, очень легко увлечься идеей, позабыв о практической реализации задуманного. Не отказывайтесь от своего стиля работы, но продвигайтесь к цели последовательно. После завершения очередного этапа в создании сценария протестируйте

фильм и убедитесь, что каждый компонент по-прежнему работает должным образом.

- ▶ **Сохраняйте несколько версий и дополнительные копии файлов.** Создание резервных копий файла необходимо, чтобы застраховаться от различных сбоев в работе компьютера. Наличие нескольких копий файла позволяет избежать потери данных вследствие удаления или повреждения файла. Таким образом, при возникновении проблем в новой версии всегда можно вернуться к предыдущей и начать работу заново.
- ▶ **Используйте функцию трассировки для контроля значений переменных и свойств.** Функция `trace` — одна из наиболее полезных в ActionScript. Она позволяет установить обратную связь с Flash Player и увидеть, что происходит при воспроизведении фильма с переменными, циклами и другими элементами сценария. Более подробная информация об этом находится в разделе «Использование функции `trace`» далее в этой главе.
- ▶ **Применяйте комментарии, описывающие код.** Комментарии можно создать, указав два символа косой черты (`//`). Flash игнорирует текст, следующий после этих символов, до конца строки. С помощью комментариев можно указать, что делает тот ли иной фрагмент кода, а также упорядочить сценарий.

#### ПРИМЕЧАНИЕ

О том, как заблокировать большой фрагмент кода, рассказывается в описании многострочного оператора комментария (`/*` и `*/`) в приложении Б этой книги.

- ▶ **Составьте блок-схему сценария.** Если вы разрабатываете проект, для которого нужно написать сложный код на ActionScript, полезно начать с работы на бумаге. Сначала спланируйте структуру сценария, а уже затем приступайте к программированию. В частности, на бумаге удобно разрабатывать сложные циклы и ветвления, поскольку в этом случае вы можете сделать все вычисления и проследить логику обработки информации.
- ▶ **Формируйте фильм из небольших частей.** Большой проект рекомендуем разбить на части и заниматься ими по порядку. Доведите до рабочего состояния одну часть фильма и только потом приступайте к следующей.
- ▶ **Решайте проблемы по мере их возникновения.** Если возникла проблема, лучше решить ее сразу, не откладывая в долгий ящик. Однако при этом учитывайте, что исправления могут повлиять на другие компоненты фильма.

Конечно, эти правила не догма, поскольку у каждого свой стиль работы. Однако надеемся, что следование нашим рекомендациям поможет вам избежать ошибок.

## Методы устранения ошибок

Практика показывает, что полностью избежать проблем невозможно. Причем ошибки в сценариях могут остановить работу не только одного программиста, но и целой рабочей группы, занимающейся одним проектом. К счастью, технические ошибки обычно преодолимы — для этого нужно расставить все по своим местам и заставить работать должным образом. Возможно, такое утверждение

звучит слишком оптимистично, но вера в победу придаст вам уверенность в себе, а уверенность — это уже половина успеха.

## Помоги себе сам: вопросник для поиска ошибок

Попытайтесь ответить на приведенные ниже вопросы, что поможет вам легче добраться до сути проблемы, если она связана с использованным в фильме сценарием ActionScript.

**Правильны ли синтаксис?** Сценарий ActionScript содержит обращения к различным элементам: действиям, свойствам, методам, функциям и т. д. Убедитесь, что при ссылке на элемент вы указали его должным образом. Проверьте по буквам правильность записи каждого идентификатора и регистра символов, где это необходимо (например, если требуется запись `onClipEvent`, а не `onclipevent`).

Убедитесь в наличии точек с запятой в конце инструкций. Проверьте, правильно ли расставлены фигурные скобки, в которые заключено тело обработчика или цикла. В ActionScript допустим пропуск точки с запятой, но лишняя скобка или, наоборот, ее отсутствие приведут к остановке воспроизведения фильма в приложении Flash Player при попытке обработки сценария.

Обратите внимания на правильность использования операторов. В частности, не забывайте, что операторы присваивания (`=`) и равенства (`==`) имеют разное назначение. Оператор `=` служит для присваивания значения:

```
var i=0;
```

Оператор `==` используется для проверки на эквивалентность и возвращает булево значение:

```
if(this._x - bounds._x){
 // операторы выполняются, если оператор == в условии возвращает true
}
```

---

### СОВЕТ

В проверке синтаксиса помощь может оказать Flash. Чтобы проверить правильность введенных инструкций ActionScript, при работе с панелью Actions нажмите кнопку Check Syntax (Проверить синтаксис) или комбинацию клавиш `Cmd/Ctrl+T`.

---

**Правильно ли определены параметры?** Убедитесь, что все параметры действий и методов заданы должным образом. Если там, где нужна строка, вы указали переменную или идентификатор, сценарий не будет работать корректно. Например, следующая инструкция не выполняется правильно, поскольку для действия `duplicateMovieClip()` имя нового экземпляра должно быть определено как строка:

```
duplicateMovieClip(bar, bar + i, 1);
```

Правильный синтаксис, где переменная `bar` определена как строка, приведен ниже:

```
duplicateMovieClip(bar, "bar" + i, 1);
```

**Выполняется ли обработчик?** Каждый сценарий запускается некоторым обработчиком. Но если обработчик не выполняется, сценарий, соответственно, не

вызывается. Выявить такую ошибку позволяет функция `trace`. Например, проверить, отвечает ли клип на нажатие мыши, можно следующим образом:

```
onClipEvent(release){
 trace("clicked");
 // другие операторы
}
```

**Правильно ли указан путь?** Если фильм содержит большое число клипов и временных шкал, легко запутаться и забыть, где какие клипы находятся. Проверьте, правильно ли указаны все пути к функциям, объектам, клипам и их свойствам. Для получения детальной информации обратитесь к разделу «Контроль сценариев с помощью окна Output» этой главы.

**Какова область видимости переменных?** Доступными из любой части фильма и в любой момент времени являются только глобальные (`_global`) переменные. Поэтому при обращении к переменным, определенным в рамках некоторой временной шкалы, необходимо указывать путь к этой временной шкале. Более подробно эта тема освещена в главе 15.

**Правильно ли выполняются математические вычисления?** Проверьте правильность математических выражений, которые содержатся в сценарии. В том месте кода, где выполняются сложные расчеты, используйте для контроля функцию `trace`.

**Правильно ли работают циклы и условные выражения?** В цикле или условном операторе вы задаете проверку некоторых параметров и затем в зависимости от результатов этой проверки выполняете какие-то действия. Если сценарий выдает неправильный результат, проверьте, все ли циклы и условные операторы функционируют в соответствии с вашим планом.

**Все ли нужные экземпляры клипов и кнопок созданы?** Управлять клипом или кнопкой из ActionScript можно только в том случае, если создан экземпляр этого объекта. Используя панель свойств (выберите команду Window ▸ Properties или нажмите клавиши `Cmd/Ctrl+F3`), проверьте, имеет ли каждый объект, к которому вы хотите обратиться, уникальное имя экземпляра.

**Все ли переменные и функции имеют уникальные имена?** Имена для переменных и функций не должны совпадать с именами других компонентов фильма.

## Средства поиска ошибок

Для решения многих проблем, возникающих из-за появления ошибок в сценарии, хорошо было бы иметь подручные средства, а не полагаться только на собственный интеллект. Например, очень пригодилась бы возможность просматривать сценарий, проверяя его шаг за шагом, и раскрывать инструкции, являющиеся источником проблем, или возможность создавать фильм по частям, проверяя работоспособность каждой из них. К счастью, Flash предоставляет набор средств, упрощающих процесс отладки сценариев. Используя их как часть стратегии поиска ошибок, можно выявить проблему, изолируя части сценария и проверяя их по отдельности. Эти средства описаны ниже.

**Комментарии.** Они упоминались ранее как метод создания заметок в тексте сценария. Вы уже знаете, что Flash игнорирует строку текста, следующую после

символов `//`. Это позволяет с помощью комментариев включать и исключать из обработки определенные части сценария. Рассмотрим пример, приведенный ниже. В нем часть условной конструкции закомментирована, поэтому при выполнении сценария инструкции в теле условного оператора не выполняются.

```
onClipEvent(enterFrame){
 _root.Bbi._y-= j;
 i+=1;
 if(i>20){
 // _root.gotoAndPlay(1);
 }
}
```

**Отладчик.** Был реализован во Flash 5 и усовершенствован во Flash MX. Он позволяет контролировать клипы, переменные, свойства и вообще все технические нюансы фильма. Отладчик подробно рассмотрен ниже.

**Окно Output.** Оно вызывается всякий раз, когда вы обращаетесь к функции `trace`. Все запрашиваемые с помощью этой функции данные выводятся в окно Output. Кроме того, это окно можно использовать для вывода информации о любых активных в фильме объектах и переменных. Более подробно окно Output описано далее в этой главе.

## Использование отладчика

Отладчик — это окно, с помощью которого можно отслеживать все, что происходит при воспроизведении фильма. Он отображает все экземпляры клипов и слои в фильме, а также свойства каждого из них. Кроме того, отладчик позволяет контролировать значения всех переменных, которые активизируются в пределах данной временной шкалы. Этот инструмент очень удобен и эффективен, поскольку предоставляет необходимые средства контроля, сосредоточенные в одном месте.

Для работы отладчика требуется дополнительное программное обеспечение — Flash Debug Player. Это специальная версия Flash Player, автоматически устанавливаемая вместе с главным приложением Flash. Flash Debug Player позволяет Flash во время проверки загружать информацию о фильме в окно Debugger.

## Активизация отладчика

Окно Debugger появляется после выбора команды `Window ▸ Debugger` или нажатия клавиш `Shift+F4`. Изначально оно не активно, поскольку это, собственно говоря, оболочка, ожидающая поступления информации. Для инициализации отладки фильма нужно активизировать отладчик. Только после этого запустится компонент Flash Debug Player и окно Debugger сможет получать информацию.

Чтобы активизировать отладку фильма, выберите команду `Control ▸ Debug Movie`. После этого Flash автоматически перейдет в режим тестирования фильма (Test Movie) и в окне Debugger отобразится содержимое фильма. По умолчанию отладчик неактивен. Для инициализации отладки следует нажать зеленую кнопку Continue в меню Code View.

## Окно отладчика

Если отладчик активизирован и работает в режиме Test Movie, он выглядит так, как показано на рис. 18.1. Его окно Debugger предоставляет средства, использование которых поможет вам понять, как работает фильм.

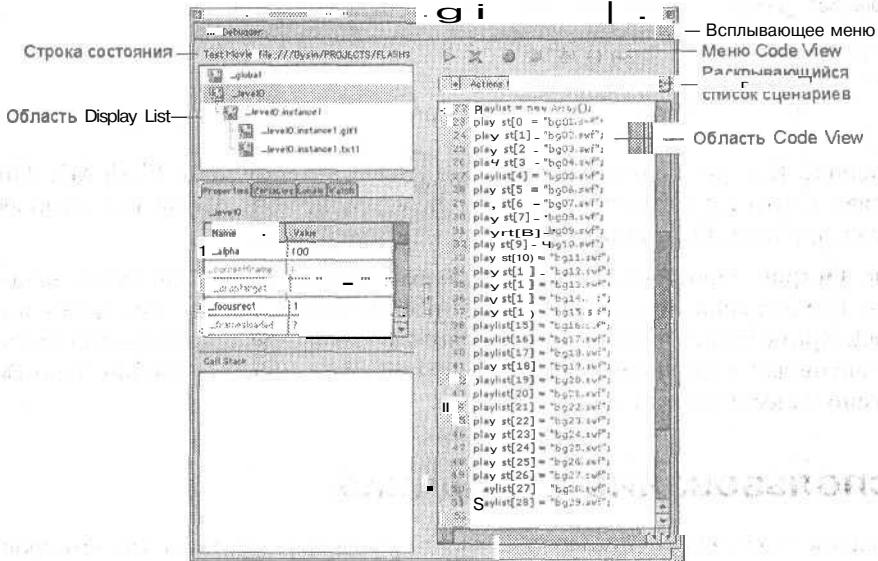


Рис. 18.1. Окно Debugger с информацией о компонентах фильма

## Строка состояния

В верхней части окна отладчика находится строка состояния, с помощью которой можно определить, на каком компьютере вы выполняете отладку (на своем или удаленном) и какой файл отлаживаете (указан путь к локальному файлу или URL).

## Область Display List

Если отладчик активизирован, в области Display List (Список) представлена структура каждого слоя фильма и каждого экземпляра клипа, включая отдельный слой для глобальных переменных (`_global`). Структура обновляется при выполнении операций дублирования, удаления, загрузки или выгрузки, производимых на основной временной шкале. Компоненты в области Display List представлены так же, как в панели Movie Explorer, поэтому вы можете определить иерархию отношений между клипами. Если фильм содержит несколько клипов, измените размеры области Display List так, чтобы в ней помещалось больше информации.

## Вкладка Properties

Вкладка Properties (Свойства) используется для просмотра и изменения свойств экземпляра клипа. Вы можете просматривать свойства клипа в процессе тестирования фильма и устанавливать их непосредственно в окне Debugger. Изменения, выполненные на вкладке, остаются в силе только па время тестирования фильма.

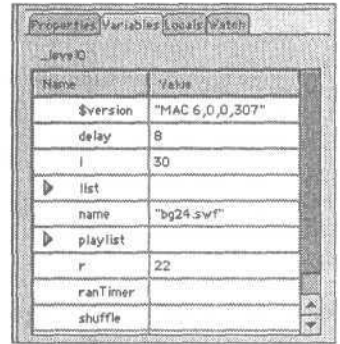


Чтобы отобразить свойства клипа, выберите его в области Display List и откройте вкладку свойств. В зависимости от структуры клипа и способа его использования в фильме некоторые свойства в области Display List могут постоянно обновляться.

Если вам нужно изменить значения свойств клипа, выберите его в области Display List, дважды щелкните на свойстве и введите новое значение в поле Value. После нажатия клавиши Return/Enter в фильме будут произведены соответствующие изменения.

## Вкладка Variables

Эта вкладка позволяет получить доступ ко всем переменным, которые доступны для данной временной шкалы. Выберите клип в области Display List и перейдите на вкладку Variables (Переменные), после этого вы увидите текущие значения переменных, которые содержит клип. Если в области Display List выбран элемент `_global`, то на вкладке Variables отображаются значения всех глобальных переменных, созданных в фильме.



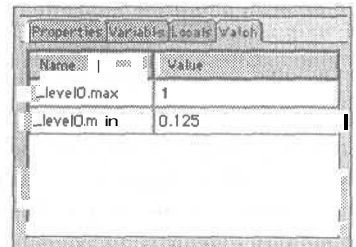
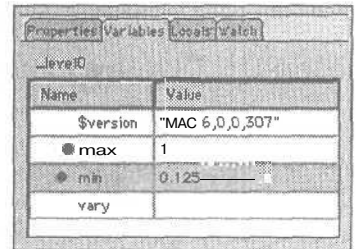
В процессе тестирования фильма значения переменных можно изменять. Выберите переменную в списке, дважды щелкните на ее значении и введите новое значение. Нажмите клавишу Return/Enter и в фильме будут произведены соответствующие изменения.

## Вкладка Watch List

Вкладка Watch List (Список контроля) позволяет одновременно видеть значения ряда переменных. Например, при просмотре значений переменных, связанных с разными клипами, вам придется каждый раз выбирать клип в области Display List и обращаться к вкладке Variables. А вот если добавить переменные в список на вкладке Watch List, то отследить их значения можно будет в одном окне.

Создать список контроля можно двумя способами:

- ▶ выбрать переменную на вкладке Variables, щелкнуть кнопкой мыши при нажатой клавише Control (Macintosh) или правой кнопкой мыши (Windows) и вызвать команду Add из контекстного меню (либо выбрать переменную и активизировать команду Add Watch из всплывающего меню окна отладчика). Выбранные для списка переменные в окне Variables будут отмечены синей точкой;
- ▶ щелкнуть кнопкой мыши при нажатой клавише Control (Macintosh) или правой кнопкой мыши (Windows) в окне Watch List, затем дважды щелкнуть на новой строке и ввести путь к переменной, которую нужно контролировать.



Учтите, однако, что свойство вкладки Properties (например, `_alpha` или `_currentframe`) нельзя непосредственно указать на вкладке Watch List, поскольку Flash позволяет наблюдать только за переменными. Чтобы обойти это ограничение, инициализируйте переменную, связанную со свойством, а затем используйте отладчик для проверки. Например, если необходимо постоянно контролировать позицию клипа по горизонтали, сначала создайте переменную для свойства `_x` (например, `hPos`)

```
onClipEvent (enterFrame){
hPos =this._x;}
```

а затем используйте вкладку *Variables*, чтобы включить переменную `hPos` в список контроля.

Для удаления элемента из списка наблюдения щелкните кнопкой мыши при нажатой клавише Control (Macintosh) или щелкните правой кнопкой мыши на элементе (Windows) и выберите команду Remove в контекстном меню. Можете также выделить элемент и вызвать команду Remove Watch из всплывающего меню отладчика.

## Контрольные точки

**MX** Во Flash MX появилась возможность устанавливать и удалять в сценарии контрольные точки, которые позволяют приостановить выполнение сценария в определенном месте кода и продолжить его выполнение в пошаговом режиме.

На панели Actions установка или удаление контрольных точек выполняется следующим образом.

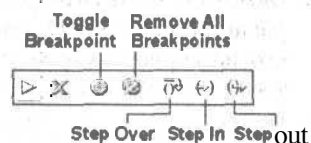
1. В области отображения сценария панели Actions отметьте мышью строку, в которой хотите установить контрольную точку.
2. Прделайте одно из следующих действий:
  - ▷ выберите команду Set Breakpoint (или Remove Breakpoint) из меню кнопки Debug Options;
  - ▷ щелкните кнопкой мыши при нажатой клавише Control (Macintosh) или нажмите правую кнопку мыши (Windows) и выберите команду Set Breakpoint (или Remove Breakpoint) из контекстного меню;
  - ▷ нажмите комбинацию клавиш `Cmd/Ctrl+Shift+B`.

В отладчике установка и удаление контрольных точек осуществляется в таком порядке.

1. В области Code View щелкните на строке кода, где необходимо установить контрольную точку.
2. Прделайте одну из следующих операций:
  - ▷ выберите команду Toggle Breakpoint или Remove All Breakpoints из меню Code View;
  - ▷ щелкните кнопкой мыши при нажатой клавише Control (Macintosh) или нажмите правую кнопку мыши (Windows) и выберите команду Set Breakpoint, Remove Breakpoint или Remove All Breakpoints из контекстного меню;
  - ▷ нажмите комбинацию клавиш `Cmd/Ctrl+Shift+B`.

После установки контрольных точек можно осуществлять построчное выполнение сценария с помощью кнопок Step In, Step Over и Step Out. Но при этом вы должны учитывать, что данные кнопки действуют только в функциях, созданных пользователем.

- ▶ Step Over — заставляет сценарий пропустить строку в функции.
- ▶ Step In — вызывает переход внутрь функции.
- ▶ Step Out — предписывает отладчику выйти из определяемой пользователем функции.



Рекомендуем не игнорировать эти средства. Со временем вы поймете, что только после их использования можно переходить к дальнейшему расследованию причин неполадок. Для того чтобы получить более подробную информацию о том, как контролировать работу сценариев, выберите команду меню Help ▶ Using Flash. Затем в открывшемся окне со справочной информацией выберите ссылку Using Flash ▶ Testing a Movie ▶ Using the Debugger ▶ Stepping through lines of code.

## Удаленная отладка

В некоторых случаях целесообразно протестировать сценарии и фильм в реальной среде. Например, если при воспроизведении фильма осуществляется интенсивный обмен данными с сервером, важно знать, правильно ли проходит этот обмен. Чтобы активизировать удаленную отладку фильма, необходимо выполнить два действия: включить удаленную отладку и активизировать отладчик из удаленного местоположения.

### Включение удаленной отладки

Для включения удаленной отладки фильма выполните следующие операции.

1. Активизируйте команду File ▶ Publish Settings.
2. В окне Publish Settings перейдите на вкладку Flash и установите опцию Debugging Permitted.

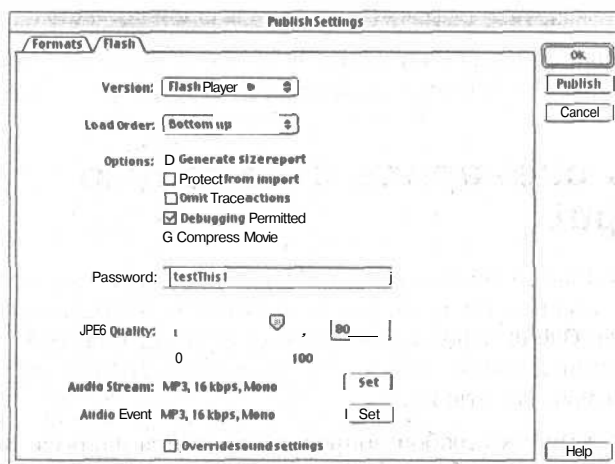


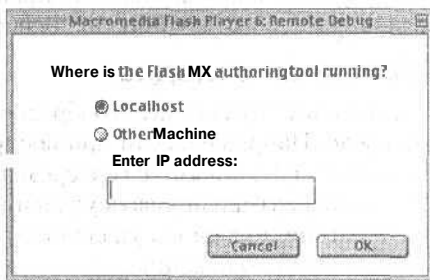
Рис. 18.2. Диалоговое окно Publish Settings

3. Если вы хотите защитить фильм паролем, чтобы проводить отладку могли только избранные пользователи, введите этот пароль в поле Password. Если пароль не задан, для отладки он не потребуется.
4. Затем опубликуйте фильм. В дополнение к SWF-файлу Flash создаст файл с расширением SWD, который необходим для удаленной отладки. Поместите его на сервер вместе с SWF-файлом и другими нужными файлами. Убедитесь, что SWD- и SWF-файлы находятся на сервере в одной папке.
5. Чтобы подготовить фильм к удаленной отладке, откройте окно отладчика с помощью команды Window ▸ Debugger (Shift+F4). Затем откройте всплывающее меню отладчика и выберите команду Enable Remote Debugging.

### Активизация удаленной отладки

Когда все необходимые параметры настройки во Flash установлены, можно производить удаленную отладку. Для активизации удаленного отладчика выполните следующие действия.

1. Запустите приложение Flash MX, если оно еще не выполняется.
2. С помощью браузера или автономного приложения Flash Player, введя URL, откройте фильм, находящийся на удаленном компьютере, после чего появится диалоговое окно Remote Debug (Удаленная отладка).
3. Выберите переключатель Localhost, если приложения Debug Player и Flash MX запущены на одном и том же компьютере, и переключатель Other Machine, если они работают на разных компьютерах. Во втором случае введите IP-адрес компьютера, на котором выполняется Flash MX. Нажмите кнопку ОК.
4. Если при публикации фильма задан пароль для отладки, введите его в специальном окне и нажмите кнопку ОК.



Самую свежую информацию об отладке фильмов во Flash MX можно найти на Web-узле компании Macromedia по адресу [www.macromedia.com/support/flash](http://www.macromedia.com/support/flash).

## Контроль сценариев с помощью окна Output

После того как во Flash 5 появилось окно Output (Вывод), можно смело говорить, что в программе имеются все средства, необходимые для работы со сценариями ActionScript. Окно Output используется в процессе создания любого фильма, содержащего сценарии данного языка. Оно позволяет задать приложению Flash Player вопросы и увидеть ответы.

Задать можно практически любой вопрос, касающийся фильма. Конечно, ответ на вопрос «Почему это не работает?» получить нельзя, но зато легко можно узнать значения переменных и Свойств, местоположение объектов на сцене, путь к

клипу или имя последней метки кадра. Кроме того, окно Output отображает сообщения об ошибках в процессе работы сценария.

Иногда можно заставить окно Output выводить информацию, не задавая ему запрос непосредственно. Например, команды List Objects и List Variables позволяют получать сведения об объектах и переменных в фильме.

В режиме редактирования фильма окно Output обычно скрыто, поскольку оно требуется, скорее, при проверке, чем при создании фильма. Однако если вы пишете сценарий на ActionScript и проверяете его синтаксис, можно вызвать данное окно с информацией об ошибках.

В режиме тестирования фильма окно Output открывается при необходимости отобразить информацию. Это может быть сообщение об ошибке или ответ на вопрос, который был задан непосредственно с помощью функции trace.

Чтобы открыть окно Output принудительно, выберите команду Window ▸ Output или нажмите клавиши Shift+F9. Окно Output появится, но не будет содержать информации.

В меню Options окна Output имеются следующие команды:

- ▶ Copy (Cmd/Ctrl+C) — копирует содержимое окна Output в буфер обмена компьютера;
- ▶ Clear — удаляет содержимое окна;
- ▶ Find (Cmd/Ctrl+F) — осуществляет поиск в окне;
- ▶ Find Again (F3) — повторяет поиск по заданному ранее критерию;
- ▶ Save to File — сохраняет всю информацию из окна в текстовом файле;
- ▶ Print — печатает все содержимое окна;
- ▶ Debug Level — устанавливает уровень детализации для информации, выводимой в окне.

## Получение списка объектов (функция List Objects)

Если вы работаете в режиме Test Movie, то можете задать отображение в окне Output определенной информации обо всех объектах фильма. Эта информация выводится упорядоченно в виде иерархической структуры. Список объектов содержит номер слоя, номер фрейма, тип объекта (клип, кнопка или форма) и путь к каждому клипу. Такая возможность особенно полезна, когда необходимо изучить поведение фильма или найти ошибки при указании путей.

Чтобы вывести список объектов, использованных в фильме Flash, выполните следующие действия.

1. Сохраните фильм и вызовите команду Control > Test Movie, чтобы перейти в режим тестирования. (Функция List Objects в обычном режиме редактирования фильма недоступна.)
2. Выберите команду Debug ▸ List Objects (Cmd/Ctrl+L), после чего в окне Output появится информация обо всех объектах фильма.

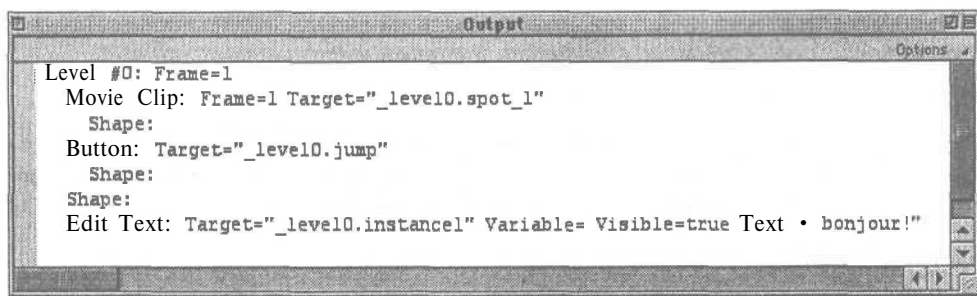


Рис. 18.3. Окно Output с информацией об объектах фильма

## Получение списка переменных (функция List Variables)

Вторая функция, которая доступна только при работе в режиме Test Movie, — вывод списка переменных. С ее помощью можно получить в окне Output список всех переменных фильма с указанием пути и значений переменных.

Чтобы вывести список переменных, которые используются в фильме Flash, выполните следующие действия.

1. Сохраните фильм и выберите команду Control ▶ Test Movie для перехода в режим тестирования. (Функция List Variables в обычном режиме редактирования фильма недоступна.)
2. Выберите команду Debug > List Variables (Cmd+Option+V/Ctrl+Alt+V), после чего в окне Output появится список всех переменных фильма. Для каждой переменной в нем указаны путь и значение в момент вывода.

Учтите, что при использовании команды List Variables или List Objects наблюдение за переменными и объектами в динамическом режиме не осуществляется. Информация в окне Output отвечает состоянию на момент получения команды.

### СОВЕТ

Для динамического контроля за переменными или объектами фильма лучше использовать область Display List и вкладку Variables окна отладчика либо функцию trace.

## Использование функции trace

Функция trace — одна из самых удобных для программиста на языке ActionScript. Она обеспечивает возможность направить Flash запрос о выдаче определенной информации, касающейся фильма. В ответ Flash выведет эту информацию в окно Output. Функция trace поможет вам собрать всю необходимую информацию о фильме, выявить неработающие обработчики, узнать значения параметров клипа и переменных, пути, результаты, возвращаемые функциями, и многое другое.

Поскольку `trace` — это действие, оно вводится на панели Actions наряду с другими инструкциями сценария. Синтаксис функции имеет следующий вид:

```
trace ("символьная информация в кавычках"+variablesOrProperties);
```

Как видите, функция `trace` способна принимать информацию разного типа. Если вы отслеживаете большое количество параметров, используйте символьные строки, в которых поясняется назначение параметров. Например:

```
trace ("Счетчик цикла: "+i);
trace ("Позиция по горизонтали: "+_root.ball.x);
trace ("Позиция по вертикали: "+_root.ball.y);
trace ("Экземпляр: "+_root.ball.newBall+i);
```

Заклученный в кавычки текст появляется в окне Output в таком же виде, как он указан в действии `trace`. Flash интерпретирует его как символьную строку, что позволяет создавать описания выводимых в окно данных. Иллюстрацией этому служит приведенный выше фрагмент сценария: в окне Output появятся не только значения заданных переменных, но и текст, поясняющий эти значения.

Обратите внимание на строковый оператор `+`, предназначенный для конкатенации (соединения) строк. Его действие легко понять на следующем примере. Введите инструкцию

```
trace (": "+"- "+"");
```

и в окне Output будет выведен смайлик `: -)`.

Функцию `trace` можно вызвать в любом месте сценария. Вы должны поместить ее там, где с ее помощью действительно можно осуществлять контроль. В частности, если значение переменной увеличивается в цикле, то инструкция `trace` должна находиться в таком его месте, чтобы можно было видеть ее значение при каждом выполнении цикла.

## Разработки

Мы стремились в каждой главе дать пример, ярко иллюстрирующий изложенный в ней материал. Данная глава является исключением, поскольку вряд ли кто-нибудь захочет, чтобы его работа была представлена как пример для отладки и поиска ошибок!

К сожалению, это задачи, с которыми рано или поздно приходится сталкиваться каждому разработчику. Поэтому здесь мы обращаем ваше внимание на Web-узел Macromedia, точнее, на ресурсы Flash по адресу [www.macromedia.com/software/flash/](http://www.macromedia.com/software/flash/).

Кроме того, хорошим источником информации являются пользовательские форумы, призванные помочь членам сообщества Flash. Так что присоединяйтесь, но не забывайте о сетевом этикете и читайте список FAQ перед отправкой своих сообщений.

## Резюме

Итак, мы заканчиваем знакомство с языком ActionScript. Вы узнали о методах и приемах, позволяющих упростить процесс создания кода на ActionScript, и мы надеемся, что это поможет вам предотвратить связанные со сценарием проблемы. Если же они все-таки появились, обращайтесь к вопроснику и наверняка сможете обнаружить место, где возникла ошибка. Учтите, ключом к успеху в деле отладки сценариев является не столько глубокое знание языка, сколько способность понять взаимосвязи компонентов сценария.

Кроме того, в этой главе вы познакомились с двумя основными средствами, применяемыми для поиска ошибок: отладчиком и окном Output. Конечно, они не исправят ошибки вместо вас, но сослужат хорошую службу, сообщая всю необходимую информацию о сценарии и фильме.



## Практикум 5

# Использование ActionScript для создания интерактивной, многослойной временной шкалы

С помощью ActionScript можно создавать интерактивные Flash-фильмы, которые будут реагировать на команды зрителей. ActionScript позволяет Flash выйти за рамки анимационной программы и превратиться в мощный инструмент для разработки всеобъемлющих мультимедийных приложений.

Данный практикум поможет вам более подробно ознакомиться с принципами создания перемещаемого бегунка, изложенными в главе 16. Используя эти принципы, мы создадим более сложный бегунок, связанный с определенной точкой на временной шкале. В данной точке осуществляется загрузка внешнего Flash-фильма (SWF-файла) в главный фильм. Далее в этом практикуме будут представлены функции, управляющие загруженными фильмами из главного фильма.

Прежде чем приступить к работе повторите особенности создания простого бегунка, просмотрев еще раз главу 16.



В первую очередь необходимо найти каталоге Hands On 5 на прилагаемом компакт-диске. При желании можете также взглянуть на конечный файл, advSlider.fla, который находится в том же каталоге.

### ПРИМЕЧАНИЕ

Если вы хотите просмотреть этот фильм, нужно либо открыть его непосредственно с компакт-диска, либо скопировать каталог Hands On 5 на жесткий диск. Данный фильм использует внешние SWF-файлы и, если отделить их от основного фильма, он будет работать некорректно.

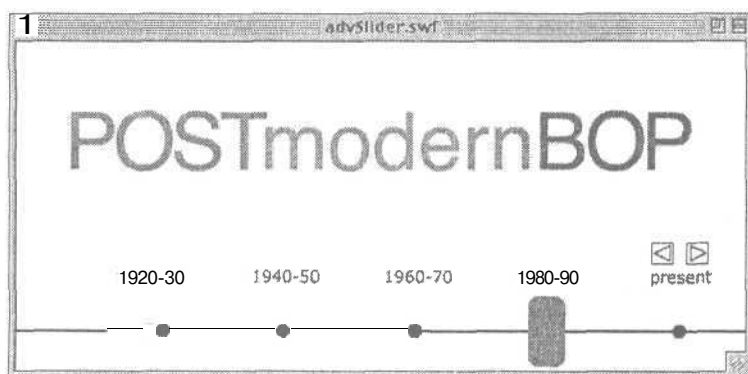
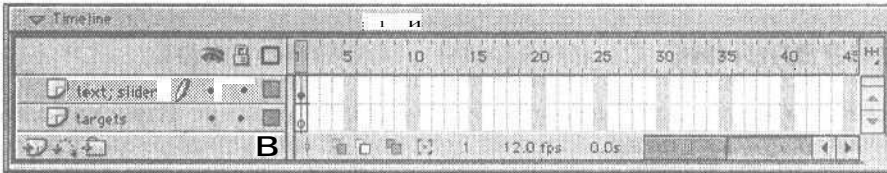


Рис. П5.1. Конечная версия создаваемого фильма

## Подготовка фильма

Готовы начать? Прекрасно! Запускайте Flash, копируйте каталог Hands On 5 на свой компьютер, открывайте файл `handsOn5.fla` — и приступайте к работе.

1. Открыв файл, вы увидите временную шкалу с датами и зеленым бегунком в левом углу области действия. Все сценарии для бегунка написаны таким образом, чтобы можно было сосредоточиться на создании его расширенных возможностей. Если хотите, выберите команду `Control ▶ Test Movie` и протестируйте фильм.
2. Когда бегунок достигает определенной точки (даты) на временной шкале, необходимо связать его с этой позицией. Для создания такого эффекта нужно добавить некоторые компоненты к фильму. На временной шкале главного фильма нажмите кнопку `Insert Layer`, чтобы создать новый слой в фильме. Призовите ему имя `targets`. Перетащите этот слой под слой с именем `text:slider`.



3. Откройте окно библиотеки (`Window ▶ Library`), в которой содержатся символы, используемые в фильме. Перетащите клип с именем `dropTarget` в первый кадр слоя `targets` (он автоматически станет ключевым). Расположите экземпляры клипа таким образом, чтобы он оказался ниже надписи `1920-30`, под первой черной точкой.

Выберите экземпляр клипа (если он еще не выбран) и откройте панель `Properties` (`Window ▶ Properties`). В поле `Instance name` укажите имя этого экземпляра — `drop1`.



Повторите этот процесс для оставшихся точек. Имена экземпляров должны быть следующими: `drop2`, `drop3`, `drop4` и `drop5` (рис. П5.2).

4. Экземпляры клипа `dropTarget` практически готовы. Однако, данный цвет экземпляров не соответствует дизайну фильма. Выберите экземпляр `drop1` на области действия, а затем в раскрывающемся списке `Color` на панели `Properties` — элемент `Alpha`. В поле `Alpha Amount` установите значение прозрачности `0`. Благодаря этому экземпляр `drop1` станет невидимым на области действия (но не менее функциональным). Повторите данную процедуру для оставшихся экземпляров клипа `dropTarget`.

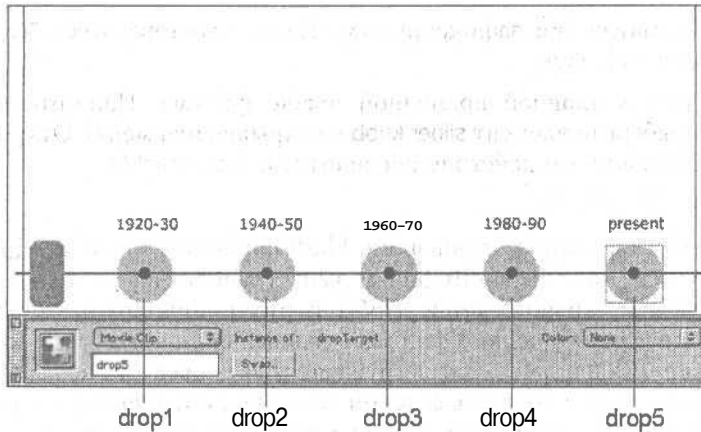


Рис. П5.2. Экземпляры клипа dropTarget, размещенные на области действия

## Добавление сценария ActionScript

5. Сейчас всё готово для добавления сценария ActionScript. Итак, необходимо чтобы бегунок останавливался на достигнутой временной точке. Щелкните на кнопке Edit Symbols и выберите Элемент slider knob в открывшемся меню.

Flash переключается в режим редактирования данного символа, и бегунок располагается посередине экрана. Выберите символ и откройте панель Actions (Window ▶ Actions). Вы увидите уже существующий сценарий, с помощью которого бегунок выполняет свои основные функции. После действия `stopDrag()`, но перед закрывающей фигурной скобкой, добавьте следующий код:

```
if(eval(_root.slider._dropTarget)==_root.drop1){
 root.slider._x=100;
 lock.start();
}
```

### ПРИМЕЧАНИЕ

Все действия должны быть помещены в фигурные скобки ({}), для обработчика событий `on (release, releaseOutside)`. А что же это за двойное действие? Хороший вопрос! Данное действие задает перемещение бегунка, даже если указатель мыши после щелчка на нем находится не над бегунком. Это превентивное действие во избежание ошибок.

Это инструкции привязки бегунка. Действие `if` проверяет условие, использующее свойство `_dropTarget`. Если бегунок находится над экземпляром `drop1`, то возвращается `true`, и выполняются следующие два выражения. Первое выражение устанавливает бегунок в позицию, имеющую по оси X координату 100, а второе воспроизводит звук `lock`. Более подробно о методе `start()` объекта `Sound` будет рассказано в главе 22.

6. Закройте панель Actions и вернитесь к главной временной шкале фильма. Выполните тестирование фильма на данном этапе, выбрав команду Window ▶ Test Movie. Поместите бегунок над первой точкой временной шкалы и убедитесь,

что он автоматически зафиксирован. Также удостоверьтесь, что при этом воспроизводится звук.

7. Возвратитесь к главной временной шкале фильма. Нажмите кнопку Edit Symbols и выберите элемент slider knob в открывшемся меню. Откройте панель Actions и добавьте это действие после строки `lock.start()`:

```
loadMovieNum ("20.swf", 1);
```

Данное действие загружает внешние Flash-фильмы в главный. Другими словами, оно загружает внешний фильм выше главного на один уровень. Действие `loadMovieNum()` используется, чтобы загрузить фильмы на уровни, находящиеся выше уровня главного фильма (главный фильм размещен на уровне 0). В нашем случае фильм `20.swf` загружается на уровень 1. Когда вы добавите это действие, то при воспроизведении всего фильма новый фильм должен загрузиться и начать воспроизводиться с левого верхнего угла области отображения кадров главного фильма. Фильм `20.swf` был создан таким образом, чтобы его содержимое заполнило верхнюю часть области отображения кадров главного фильма.

Вы, конечно, заметили, просмотрев содержимое скопированного каталога `handsOn5`, что он содержит несколько файлов с фильмами. Действие `loadMovieNum()` задает загрузку внешнего Flash-фильма в главный фильм. Для этого необходимо указать местонахождение внешнего фильма. В первом аргументе действия `loadMovieNum()` нужно указать URL, то есть путь к внешнему фильму. Путь может быть либо полный, либо относительный. В нашем случае указан относительный URL, так как все фильмы находятся в одном каталоге `handsOn5`.

8. Сохраните фильм, если вы еще этого не сделали. Вернитесь к главной временной шкале фильма и протестируйте его, выбрав команду Test Movie меню Control. Перетащите бегунок к первой точке. После того как он зафиксирован, загрузится фильм `20.swf`.
9. Вернемся еще раз к главной шкале фильма и добавим действия к другим экземплярам клипа. С помощью добавляемых действий каждая точка временной шкалы станет точкой привязки. При помещении на нее бегунка новый фильм будет загружаться на уровень 1. Снова переключитесь в режим редактирования символа slider knob, выберите его и откройте панель Actions. Добавьте следующие действия таким образом, чтобы полный код обработчика событий `on(release, releaseOutside)` соответствовал приведенному ниже фрагменту сценария.

## ПРИМЕЧАНИЕ

Два фильма не могут занимать одновременно один и тот же уровень. Так как в данном случае каждый SWF-файл загружается на уровень 1, предыдущий фильм будет заменяться последующим.

```
on(release, releaseOutside){
 stopDrag();
 if(eval(_root.slider.dropTarget)==_root.drop1){
 _root.slider_x=100;
```

```

 lock.start();
 loadMovieNum("20.swf",1);
 }elseif (eval(_root.slider._dropTarget)==_root.drop2){
 _root.slider._x=200;
 lock.start();
 loadMovieNum("40.swf",1);
 }else if (eval(_root.slider._dropTarget)==_root.drop3){
 root.slider._x=300;
 lock.start();
 loadMovieNum("60.swf",1);
 }elseif (eval(_root.slider._dropTarget)==_root.drop4){
 _root.slider._x=400;
 lock.start();
 loadMovieNum("80.swf",1);
 }elseif (eval(_root.slider._dropTarget)==_root.drop5){
 _root.slider._x=500;
 lock.start();
 loadMovieNum("00.swf",1);
 }else{
 unloadMovie(1);
 }
}

```

Приведенный сценарий может показаться вам сложным, но на самом деле это довольно простой пример. Восприятие данного кода улучшится, если рассматривать каждое положение бегунка в виде четырех выражений:

условие для проверки положения  
 позиция ползунка (эффект привязки)  
 проигрывание звука  
 загрузка внешнего фильма

### Использование конструкции **switch...case**

Flash MX позволяет также переписать сценарий, применив конструкцию `switch... case` вместо конструкции `if...else`. При использовании `switch...case` выполняются выражения, соответствующие условиям которых возвращают `true`. Синтаксис вышеприведенного сценария можно сделать более удобным для чтения:

```

on(release,releaseOutside){
 stopDrag();
 dragging=false;
 switch (eval(_root.slider._dropTarget)){
 case eval(_root.drop1):
 _root.slider._x=100;
 lock.start();
 loadMovieNum("20.swf",1);
 break;
 case eval(_root.drop2):
 _root.slider._x=200;
 lock.start();
 loadMovieNum("40.swf",1);
 break;
 case eval(_root.drop3):
 _root.slider._x=300;
 lock.start();
 loadMovieNum("60.swf",1);
 }
}

```

```

 break;
 case eval(_root.drop4):
 _root.slider._x=400;
 _lock.start();
 _loadMovieNum("80.swf", 1);
 break;
 case eval(_root.drop5):
 _root.slider._x=500;
 _lock.start();
 _loadMovieNum("00.swf", 1);
 break;
 default:
 _unloadMovie(1);
}
}

```

В каждой ветви case сценария содержатся условие и выражения, управляющие соответственно каждым экземпляром от `drop1` до `drop5`. Если бегунок остановлен над одним из этих экземпляров, то выполняются выражения соответствующего условия. Обратите внимание, что каждый набор выражений уникален. С их помощью бегунок привязывается к различным точкам, и загружается внешний SWF-файл.

Действие `unloadMovie()` выполняется по умолчанию. Если бегунок не остановился ни над одним из экземпляров клипа, то выгружается любой фильм, который воспроизводится на уровне 1.

---

#### ПРИМЕЧАНИЕ

Чтобы выгрузить фильм, не требуется указывать URL, для этого необходимо указать только уровень, который вы хотите удалить.

---

10. Вернитесь к главной временной шкале фильма и сохраните ваш фильм. Затем выберите команду **Control ▶ Test Movie** и проверьте новые возможности фильма. Каждый раз, когда бегунок переместится к следующей точке, загружается новый фильм. И наоборот, когда бегунок покидает точку, фильм выгружается.

## Добавление кнопок управления воспроизведением

11. На следующем шаге необходимо добавить некоторые кнопки, которые позволят управлять загруженным фильмом из главного фильма. Перетащите символ кнопки `navButton` из панели `Library` на область действия фильма главной временной шкалы. Поместите его в правый нижний угол области действия, ниже отметки 150 пикселей вертикальной линейки.

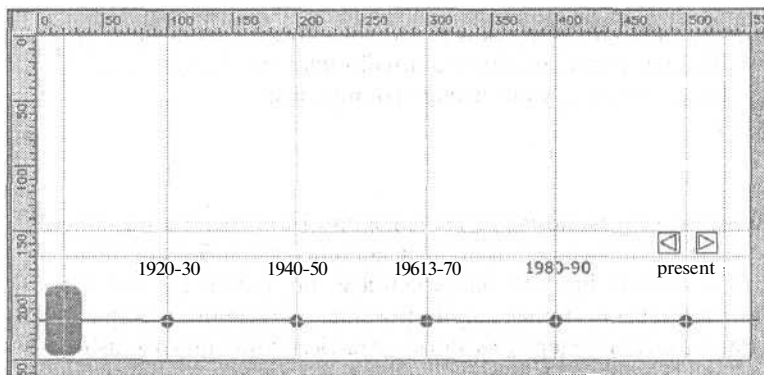
---

#### СОВЕТ

Каждый загружаемый фильм имеет высоту области отображения кадров 150 пикселей. Если вы разместили кнопки над отметкой 150 пикселей, то во время загрузки внешних фильмов в основном они будут закрыты. Для того чтобы на экране отображались линейки, выберите команду **View ▶ Rulers**.

---

Откройте панель Transform (Window ▶ Transform). В поле Rotate, введите значение 180, чтобы зеркально отразить кнопку по горизонтали. Одна кнопка готова. Для создания второй кнопки перетащите еще раз символ `navButton` на область действия из окна библиотеки и поместите его правее уже готовой кнопки. На данном этапе ваш экран должен соответствовать приведенному ниже рисунку.



12. Далее мы создадим две функции, которые будут определять реакцию фильма на нажатие пользователем соответствующей кнопки. Щелкните на первом кадре слоя `targets` на главной временной шкале фильма. Откройте панель Actions (Window ▶ Actions) и введите следующие строки сценария:

```
function rewind() {
 _level1.gotoAndPlay(1);
}
function skip() {
 _level1.gotoAndStop("end");
}
```

В данном коде создаются две функции. При вызове функции `rewind()` головка воспроизведения переходит к первому кадру фильма и с этого места начинается воспроизведение фильма, находящегося на уровне 1 (`_level1`). Функция `skip()` работает подобным образом: когда она вызывается, то головка воспроизведения переходит к кадру с меткой `end`, и воспроизведение фильма, находящегося на уровне 1 (`_level1`), прекращается. Применение функций имеет следующее преимущество: вместо того чтобы писать каждый раз сценарий для реакции фильма на одинаковые действия пользователя, вы просто вызываете функцию. Это экономит время, особенно если функция будет использоваться в фильме многократно. (Подробная информация о функциях содержится в главе 15 или приложении Б.)

Также очень важно правильно связать сценарий с кадрами фильма. В нашем фильме он связан с первым кадром сцены `Scene 1` главной временной шкалы. Это означает, что при воспроизведении фильма данный сценарий обрабатывается в первую очередь и все используемые функции станут доступными.

13. Создание функций — только первая часть процесса. Для использования в фильме их нужно вызвать действиями **ActionScript**. Выберите кнопку с изображением треугольника, указывающего влево. Откройте панель **Actions (Window ▶ Actions)** и введите следующий код:

```
on(release) {
 rewind();
}
```

Это действие вызывает функцию `rewind()` после щелчка мыши. Теперь аналогичным образом свяжите приведенный ниже код с другой кнопкой (с изображением треугольника, указывающего вправо):

```
on(release) {
 skip();
}
```

В этом сценарии после нажатия связанной с ним кнопки вызывается функция `skip()`. Поскольку обе функции используются без аргументов, они будут работать в любом фильме, загруженном на уровне 1 (`_level1`). Однако есть одно но: чтобы выполнялась функция `skip()`, последний кадр каждого внешнего фильма должен содержать метку `end`. Без этой метки Flash не будет знать, как интерпретировать функцию, потому что не сможет найти метку, на которую есть ссылка в теле функции `skip()`.

#### ПРИМЕЧАНИЕ

В этом примере нет необходимости указывать путь к объекту для функции, поскольку она была определена и вызывается из фильма главной временной шкалы (`_root`). Если функции вызываются из фильма уровня `_level1` или из клипа, нужно указывать путь, например `_root.rewind()`. Функции, подобно переменным, имеют область действия только в пределах фильма, где они были созданы.

14. Закройте панель **Actions** и вернитесь к главной временной шкале фильма. Сохраните фильм и протестируйте его, выбрав команду **Control ▶ Test Movie**. Когда вы щелкните мышью на кнопках, которые были добавлены последними, загружаемый внешний фильм должен соответственно или начать воспроизводиться, или остановиться на последнем кадре.

## Резюме

В данном практикуме представлен процесс создания интерактивного фильма, в который загружаются внешние SWF-файлы. Вы научились программировать функции, позволяющие управлять воспроизведением внешнего фильма из главного фильма. И это только начало. Знания, полученные вами, позволяют поднять данный проект на новый уровень. Теперь к нему можно добавить:

- ▶ большое количество внешних фильмов;
- ▶ дополнительные навигационные кнопки, которые позволят зрителю переходить к различным частям (а не только в начало или конец) внешних фильмов;
- ▶ динамически изменяемый цвет фона (если ползунок находится не над точкой временной шкалы).





# ЧАСТЬ 6

---

## Работа со звуком

*Часть 6 нашей книги посвящена аспектам работы со звуком в программе Macromedia Flash MX. В фильме, созданном при помощи Flash, под звуком подразумевают различные звуковые эффекты, музыку, диалоги и т. д. Звук наполняет жизнью фильмы и анимацию. Наделив персонажей голосами, вы усилите впечатления зрителей от просмотра комических или драматических сцен фильма, чего невозможно добиться при помощи субтитров или иного текстового сопровождения.*

*Звук позволяет установить более тесный контакт со зрителем. Например, посредством звукового сигнала, сопровождающего нажатие клавиши или щелчок мышью, можно отреагировать на какое-либо действие пользователя. А как много возможностей таит в себе музыка! Сопровождая различные сцены фильма, она придает им тот или иной эмоциональный окрас. Создать интерактивную аудиосреду, в которой звуки выполняли бы те или иные функции, можно при помощи макроязыка ActionScript. В настоящей главе рассмотрено многообразие средств, предоставляемых приложением Flash для работы со звуком.*

|                                                                    |     |
|--------------------------------------------------------------------|-----|
| <b>Глава 19.</b> Принципы цифровой записи звука в среде Flash..... | 386 |
| <b>Глава 20.</b> Основы работы с аудиофайлами в среде Flash.....   | 400 |
| <b>Глава 21.</b> Синхронизация звука и анимацией.....              | 415 |
| <b>Глава 22.</b> Программное управление звуком.....                | 427 |

## Глава 19

# Принципы цифровой записи звука в среде Flash

Звук является одним из наиболее важных компонентов мультимедиа. Под словом *звук* подразумеваются различные звуковые эффекты, музыка, диалоги и т. п. Хотя все эти элементы могут значительно улучшить мультимедийные проекты, разработчики не всегда используют звук при создании Flash-фильмов. Тем не менее звук является прекрасным дополнением при разработке приложений, позволяющим удержать внимание зрителя возле вашего Flash-продукта.

Существует много причин того, что звук играет не самую важную роль в мультимедийных приложениях, особенно в тех, которые используются в Интернете. Однако большинство этих причин не касается непосредственно Flash, поэтому здесь мы не будем их обсуждать. Главное то, что в среде Flash можно без проблем работать со звуком. Описываемая среда предоставляет различные варианты использования звука и музыки в создаваемых продуктах. Каждый разработчик, имеющий в распоряжении копию приложения Flash, может выбрать для себя наиболее подходящий из них.

В этой главе мы обсудим возможности работы со звуком в среде Flash. Но сначала познакомим тех, кто является новичком в области звукозаписи, с некоторыми техническими терминами и основными принципами цифровой записи звука. Здесь будут рассмотрены следующие темы:

- ▶ частота и глубина выборки при цифровой записи звука;
- ▶ стерео- и монозвук;
- ▶ типы цифровых аудиофайлов, поддерживаемые средой Flash;
- ▶ где и каким образом подобрать звуковое оформление для вашего проекта;
- ▶ подготовка аудиофайлов для работы в среде Flash;
- ▶ планирование использования звуков;
- ▶ работа с циклическими звуками.

## Основные принципы цифровой записи звука

Техническая часть, касающаяся цифровой записи звука, может вам показаться довольно скучной и сложной. Однако эта тема является очень важной, особенно если ваш проект находится на завершающей стадии разработки. Знание особенностей цифровой записи звука позволяет разработчику с самого начала работы над проектом не только выбрать наиболее подходящее звуковое оформление для своего фильма, но и успешно решать проблемы, возникающие в процессе его создания. Кроме того, хорошее знание работы звуковых файлов будет полезным при их использовании в различных частях проекта.

## Частота выборки

Вероятно, большинство читателей уже имеет некоторый опыт записи звука на магнитную ленту. Вам наверняка приходилось использовать магнитофоны, диктофоны или другие устройства записи звука. При записи на магнитную ленту звуковые волны сохраняются на ней, будучи преобразованными в магнитные импульсы. При воспроизведении происходит обратное преобразование магнитных импульсов в звуки.

Работая с подобными устройствами записи звука, вы приобрели навыки *аналоговой* звукозаписи. Аналоговой ее называют потому, что записанная магнитная информация по возможности наиболее приближена (является *аналогом*) к оригиналу. Положительные и отрицательные полупериоды звуковой волны имитируются магнитными доменами, ориентированными своими положительными либо отрицательными концами к поверхности носителя (в данном случае — магнитной ленты), на котором записан звук (рис. 19.1).

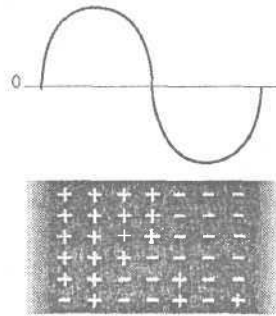


Рис. 19.1. Звуковая волна и имитирующие ее домены на магнитной ленте

Кроме аналоговой существует и *цифровая* запись звука, в которой вместо аналоговых сигналов используются цифры 1 и 0. В процессе цифровой записи звука записывающее устройство анализирует звуковую волну и записывает величину ее амплитуды в отдельные моменты времени. Такая запись называется *выборкой*, а частота, с которой производятся подобные записи, — *частотой выборки*. Сказанное иллюстрирует рис. 19.2.

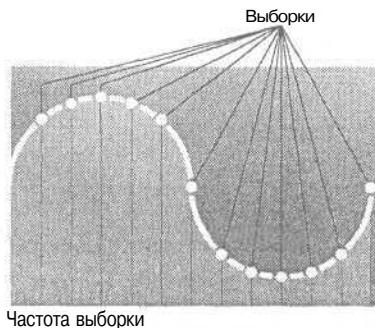


Рис. 19.2. Схематичное представление процесса оцифровки звуковой волны

Как правило, при цифровой звукозаписи используют частоту выборки, равную 44,1 кГц, или 44100 копий в секунду. Это означает, что за каждую секунду звукозаписи в цифровом виде записывается более 44000 единиц информации, последовательность которых моделирует звук длительностью в одну секунду. Если понизить частоту выборки в два раза (22,05 кГц), возможность представления звуковой волны останется, однако качество звука при этом будет хуже, поскольку объем информации уменьшится и останется меньше «мгновенных снимков» звуковой волны для ее воспроизведения.

Значение 44,1 кГц для частоты выборки не является случайным. Оно было определено при исследовании передачи сигналов данных.

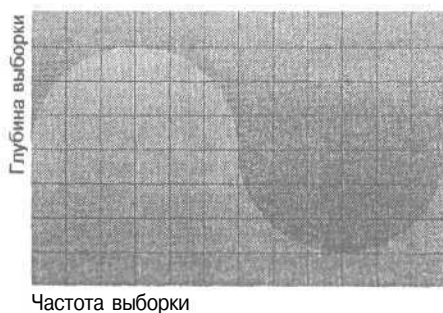
В 1920 году Найквист разработал множество теорий, касающихся звукозаписи. Одна из них оказала значительное влияние на технологию цифровой звукозаписи. В теореме Найквиста говорится, что для качественного представления звука частота выборки должна как минимум в два раза превышать наибольшую частоту спектра записываемого звука. Объяснение этому довольно простое. Звуковая волна состоит из двух полупериодов: положительного и отрицательного. Поэтому для ее имитации необходимо иметь хотя бы по одной выборке на каждом из полупериодов.

Человек способен воспринимать звуки частотой до 20 кГц. Это означает, что по теореме Найквиста частота выборки при записи звуковых колебаний должна равняться как минимум 40 кГц. Тогда сохраненные выборки позволят воспроизводить звуковую волну внутри диапазона, воспринимаемого человеческим ухом.

Частота выборки для стандартного звукового компакт-диска составляет 44,1 кГц, что на 4,1 кГц больше минимального значения. В этот интервал попадают все необходимые значения частот, а кроме того, остается некоторый частотный резерв.

## Глубина выборки

Еще одним важным параметром является *глубина выборки* — число возможных значений, которыми можно описать амплитуду волны для данной выборки при цифровой звукозаписи. Частота выборки характеризует точность описания звука по времени, а глубина выборки — точность описания звука по амплитуде (рис. 19.3).



**Рис. 19.3.** Глубина выборки характеризует амплитуду волны (ось Y) для каждой выборки (ось X) в звуковом файле

Глубина выборки, как и частота выборки, имеет определенные ограничения. Большинство устройств могут записывать звук лишь при частоте выборки 44,1 кГц или 48 кГц и глубине выборки, равной 8 или 16 бит. Для представления глубины выборки используются двоичные числа. При 8-битном представлении это соответствует 256 значениям ( $2^8=256$ ), в диапазоне от 00000000 до 11111111. При 16 битах количество возможных значений увеличивается до 65536 ( $2^{16}=65536$ ), в диапазоне от 0000000000000000 до 1111111111111111. Видно, что спектр значений в 16-битном представлении гораздо богаче. При таком разрешении копия волны больше похожа на оригинал, чем в случае 8-битного представления звука. Обратите внимание на рис. 19.4. При 8-битном представлении звука многие выборки округлены до ближайшего значения и форма волны менее похожа на оригинал. Во втором же случае (то есть при 16-битном представлении) округленных значений меньше, а форма волны больше соответствует оригиналу (хотя все равно, далеко не идеальна).

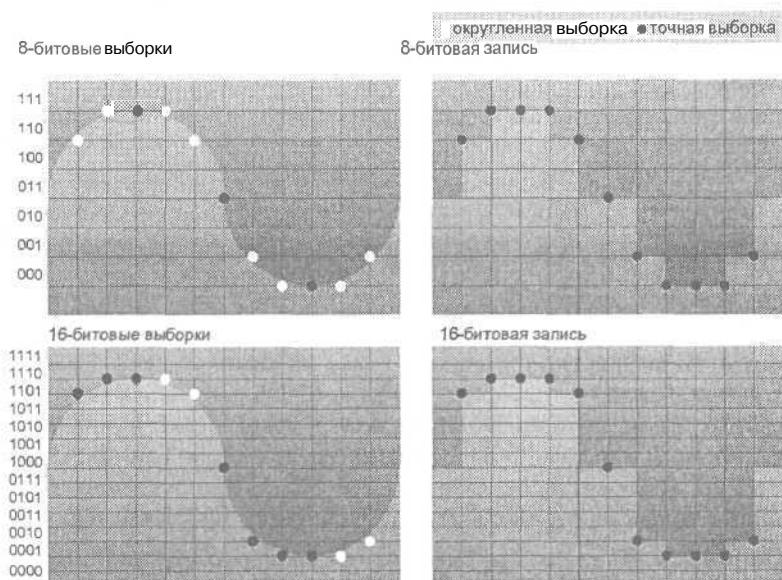


Рис. 19.4. Выборки при 8- и 16-битном представлении звука

## Сtereo- и монозвук

Слово *стерео* вряд ли является незнакомым для вас. Но даже если вы сталкиваетесь с ним впервые, это не означает, что вам никогда не приходилось слышать стереозвук. Что же такое стереозвук? Большинство людей скажут: «Этот звук лучше!» Конечно, в некотором роде это правильный ответ, но *полным* его не назовешь. Под словом *стерео* подразумевается процесс распространения звука по двум *синхронным* каналам — левому и правому, что соответствует их физическому расположению. При этом для воспроизведения звука можно задействовать как оба канала *одновременно*, так и каждый из них по отдельности. Использование двух каналов *одновременно* позволяет добиться эффекта присутствия.

Процесс привязки звука к левому и правому каналам называют *панорамированием*. Звук от каждого отдельного источника в стереозвучании обладает *уникальной* панорамированной информацией. Например, при записи музыки в стиле рок-н-ролл барабаны и бас вы можете слышать в обоих каналах, в то время как звучание соло-гитары будет доноситься из левого канала, а ритм-гитары — из правого. Не существует жестких правил, определяющих распределение звучания инструментов по каналам. Далее в этой главе будут приведены некоторые соображения по использованию стереозвучания в звуковом оформлении.

Альтернативой стереозвучанию является *монозвук*. Монозвук передается только по одному каналу, и его можно воспроизвести через обычный громкоговоритель. При *двухканальной* установке монозвук будет поровну разделен между левым и правым каналами. Монозвук не содержит никакой информации о панорамировании, поэтому он больше подходит для воспроизведения диалога, звуковых эффектов и других звуков, когда расположение источника звука не играет роли или звуковая позиция может быть задана позже, например посредством команд макроязыка *ActionScript*.

#### ПРИМЕЧАНИЕ —

В среде Flash существует ряд возможностей управления параметрами стерео- и монозвучания в фильме. Основные параметры описаны в главе 20. В главе 22 будет рассмотрено использование макроязыка *ActionScript* для управления звуком.

## Сохранение звука: типы цифровых аудиофайлов

Чтобы использовать созданный звуковой файл, его нужно прежде всего сохранить. Для сохранения цифрового аудиофайла требуется сравнительно большой объем памяти. В частности, при записи звука на компакт-диск создается 44100 копий в секунду, каждая из которых включает 65356 разрядов для описания значения выборки. Если же производится запись стереозвучания и информация направляется по двум каналам, объем данных удваивается. Так, десятисекундная стереозапись на компакт-диске занимает 1,7 Мб. Соответственно, для трехминутной записи понадобится около 30 Мб свободного пространства.

Существует множество способов уменьшения размеров звукового файла. Далее мы ознакомим вас с ними. А сейчас рассмотрим форматы звуковых файлов, которые поддерживаются в среде Flash. Вы уже знаете, что цифровая аудиоинформация имеет три составляющие: частоту и глубину выборки, а также *стереоинформацию*. Все эти данные организованы и упакованы в цифровой аудиофайл, который может быть прочитан соответствующими программами, включая Flash.

В среде Flash возможно использование звуковых файлов различных форматов. Каждый формат имеет свои особенности. Ниже приводится список различных типов звуковых файлов, которые поддерживаются средой Flash. Включение этих файлов в Flash-фильмы будет рассмотрено в главе 20.

- ▶ **AIFF: Audio Interchange File Format (.aif).** Данный формат предназначен для применения на компьютерах типа Macintosh; с ним также можно работать на компьютерах, где установлена операционная система Windows, используя программу QuickTime версии 4 или выше.

- ▶ **WAV: Windows Audio File Format (.wav).** Этот формат главным образом предназначен для использования в ОС Windows, хотя на компьютерах типа Macintosh его также можно применять при работе с QuickTime версии 4 или выше.
- ▶ **MP3: MPEG Layer 3 (.mp3).** Формат с внутренним сжатием данных, который используется как в ОС Macintosh, так и в ОС Windows. Это единственный аудиоформат со сжатием данных, который поддерживается в среде Flash. О MP3-сжатии речь пойдет в главе 20.
- ▶ **Sound Designer II.** Несмотря на то что фирма Digidesign прекратила выпуск программного обеспечения, в некоторых других программах редактирования звука формат Sound Designer II все еще применяется.
- ▶ **QuickTime movies (только звук).** Фильмы QuickTime не обязательно содержат видеoinформацию. Фильмы, не содержащие видеоданных, называются *звуковыми*. Они могут использоваться в среде Flash как на компьютерах типа Macintosh, так и на компьютерах с ОС Windows.
- ▶ **AU: Sun AU (.au).** Этот формат применяют в основном на компьютерах Sun и Unix, но он также может поддерживаться средой Flash на компьютерах типа Macintosh и компьютерах с ОС Windows.
- ▶ **System 7 Sound.** Используется на компьютерах типа Macintosh и является основным форматом этой операционной системы.

## Подготовка аудиофайлов для работы в среде Flash

Добавление соответствующего звукового сопровождения является завершающим этапом при разработке фильма в среде Flash.

Однако вы ошибаетесь, если думаете, что включение звука во Flash-фильм — это такое же простое дело, как его прослушивание. Прежде всего следует убедиться, что звуковое сопровождение соответствует вашему замыслу. После того, как вы выбрали подходящий звуковой файл или создали его самостоятельно, предстоит проделать множество операций по его редактированию. Выполнив их, вы можете быть уверены, что звук готов к воспроизведению.

## Звуковые ресурсы

Существует много источников, где можно найти звуковое сопровождение для фильма. Однако лучше всего проявить творческие способности. Если вы поете, играете на каких-либо музыкальных инструментах или музицируете, стуча по кастрюлям и сковородкам, то вполне можете сами создать звуковое сопровождение для своего фильма. Фантазируйте! Ведь, работая над фильмом, вы представляете, какие именно звуки вам нужны. Что же мешает самостоятельно их создать? Помните, что во многих случаях свежесть идеи и оригинальность оформления делают незаметным техническое несовершенство. Так что, если вы обладаете музыкальными способностями, дерзайте!



Если вы создаете фильм об окружающем мире, возьмите портативное записывающее устройство и в поисках подходящих звуков выйдите на улицу. Вам пригодятся любые звуки: пение птиц, шелест листвы и т. д. Профессиональные звукооператоры не всегда используют «натуральные» звуки, некоторые звуковые эффекты они получают сами. Нужно лишь немного пофантазировать. Например, звук падения какого-либо предмета на землю можно симитировать, ударив чем-нибудь по обычной картонной коробке. А звукового эффекта потрескивания дров в костре можно добиться, комкая в руках обычный полиэтилен. Возможности для фантазии здесь безграничны.

Подобрав подходящие звуковые эффекты, перенесите их на компьютер для дальнейшего использования в среде Flash. Для этого понадобится записывающее устройство, кабель для присоединения данного устройства к компьютеру, звуковая плата и соответствующие программы для работы со звуковыми файлами. Если у вас имеется портативный компьютер, звуки на него можно записывать непосредственно.

Если же самостоятельные попытки записать звуковые эффекты оказались неудачными или вы ограничены во времени, воспользуйтесь приведенными ниже вариантами.

- ▶ **Звуковые эффекты на компакт-дисках.** Для использования в мультимедийных проектах можно приобрести библиотеки звуковых эффектов, доступные либо через соответствующие фирмы, либо в Интернете. Но будьте осторожны и всегда внимательно читайте лицензионное соглашение для любой библиотеки, какой бы она ни была.
- ▶ **Свободно распространяемая музыка и компакт-диски с циклическими звуками.** Это также достаточно хороший ресурс для подбора звуков к вашим фильмам. Но опять же, если внимательно поискать в Интернете, вам представится большой выбор подобных ресурсов. И вновь напоминаем вам о лицензионных соглашениях.
- ▶ **Среда Flash и звуковые ресурсы всемирной сети Интернет.** Различные звуковые ресурсы, включая аудиоэффекты, музыку и циклические звуки, можно найти по таким адресам, как [www.flashplanet.com](http://www.flashplanet.com), [www.sonify.org](http://www.sonify.org) и [www.flash-kit.com](http://www.flash-kit.com). Не забывайте внимательно читать лицензионные соглашения перед использованием ресурсов, полученных из Интернета.
- ▶ **Семейство звуков DoReMedia.** Компания DoReMedia ([www.doremmedia.com](http://www.doremmedia.com)) разработала музыкальные «семейства», специально предназначенные для создания многоканальных записей в среде Flash. Их можно получить прямо на Web-узле DoRoMedia. Более подробно про эти технологии рассказывается в главе 21.

## Редактирование звуковых файлов

Независимо от того, где и каким образом вы приобрели звуковые файлы для своего проекта, их нужно отредактировать и подготовить для использования в среде Flash. Под редактированием подразумевается следующее. Необходимо добиться того, чтобы звук был чистым, то есть исключить нежелательные звуковые эффекты. Поскольку аудиофайлы занимают много места и существенно

увеличивают объем конечного продукта, следует также убедиться в отсутствии щелчков, треска и прочих шумов, производимых аппаратурой в процессе звукозаписи.

В следующих двух разделах речь пойдет о наиболее важных операциях редактирования, служащих для подготовки звуковых файлов к использованию в Flash-фильмах. Хотя операций редактирования аудиофайлов довольно много, описанные ниже являются необходимыми и должны производиться над каждым обрабатываемым звуковым файлом.

## Удаление ненужных участков звуковых файлов

Название данного раздела говорит само за себя, но почему так важно удалять ненужные участки в звуковых файлах? Потому что лишняя секунда монозвука может стоить вам 80 или 90 килобайт. В частности, трехсекундным файл с записью барабанной дроби будет иметь такой же размер, как и трехсекундный файл, где не записано ничего, кроме тишины. Поэтому за размером аудиофайлов необходимо тщательно следить.

Открыв сохраненный звуковой файл, вы увидите в начале и в конце записи пустые участки. Удаление пустого пространства позволит не только уменьшить объем файла, но и точнее синхронизировать его воспроизведение с анимацией. Если в начале аудиофайла записана пауза длительностью в ... секунды, звук будет стартовать позднее, чем нужно, и вовсе не потому, что Flash работает медленно (рис. 19.5). В «очищенных» звуковых файлах исключается наличие подобных пауз.

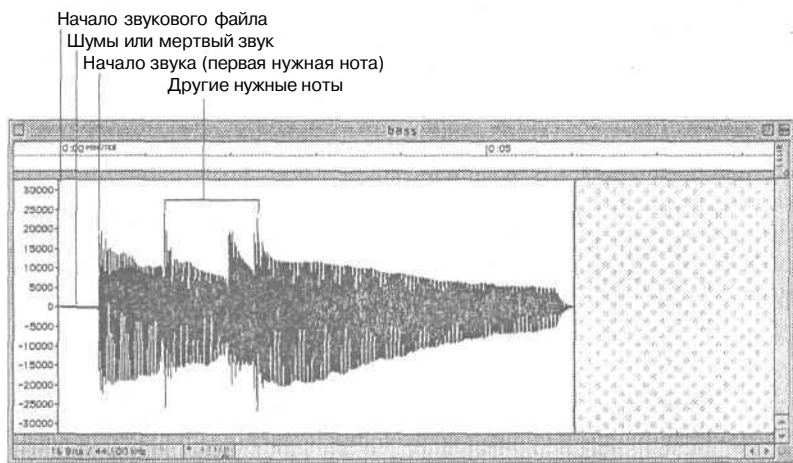
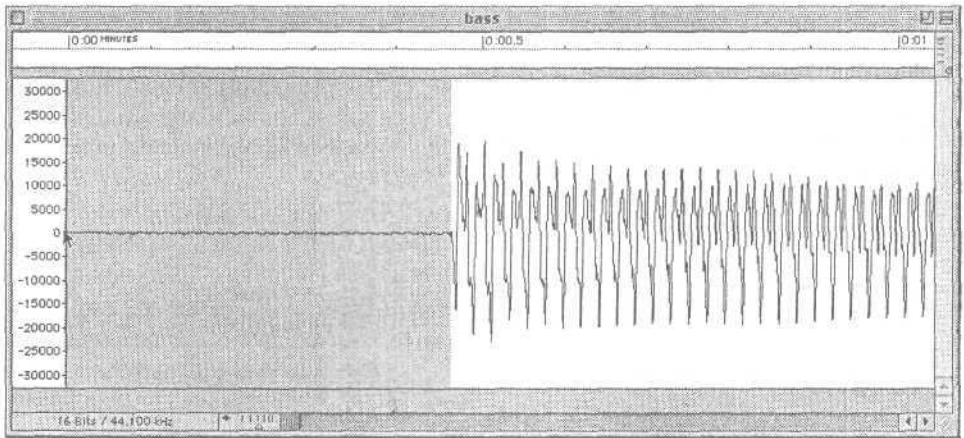


Рис. 19.5. Данный звук содержит подлежащий удалению участок

Чтобы «очистить» звуковой файл от ненужных участков, выполните следующие действия.

1. Откройте файл с помощью соответствующего приложения. В ОС Windows с этой целью можно использовать программу Sound Forge. Если же вы работаете на компьютерах типа Macintosh, выберите одну из двух программ — SoundEdit 16 или Peak.

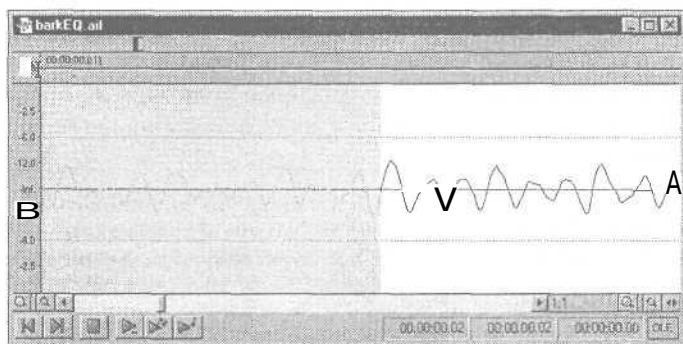
2. Перейдите к началу файла. В зависимости от инструмента или объекта, издающего звук, волны будут выглядеть по-разному. Воспроизведите звук и следите за воспроизводящей головкой при прослушивании. Наблюдая за ее перемещением, вы сможете найти точку, в которой начинается основная часть записи.
3. Когда вы нашли настоящее начало звука, щелкните левой кнопкой мыши в этой точке и перетащите курсор в начало файла, тем самым выделив участок, подлежащий удалению. Это будет выглядеть примерно так, как на рис. 19.6.



**Рис. 19.6.** Чтобы очистить звуковой файл, прежде всего необходимо выделить ненужную часть

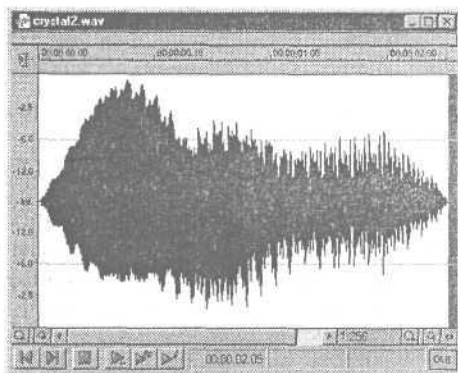
4. Прослушайте еще раз выделенный участок. Содержит ли эта запись какие-либо важные звуки? Вспомните, что вы удаляете *ненужную* информацию. При воспроизведении выделенной части вы будете слышать только ее, иными словами, часть, которую нужно удалить. Если ваш редактор поддерживает функцию воспроизведения невыделенной части, вы сможете прослушать очищенный звуковой файл.
5. Убедившись, что выбор участка корректен, удалите ненужную часть нажатием клавиши Delete или путем использования команды Delete (Удалить) из меню Edit (Правка) в звуковом редакторе. Если же вы сомневаетесь в правильности выбора, выделите фрагмент заново. В случае необходимости посредством команды Edit ▶ Undo (Правка ▶ Отменить) можно отменить операцию удаления, после чего произвести ее более тщательно.
6. Теперь очистите конец файла, выполнив аналогичные операции.

Существуют некоторые особенности, которые следует учитывать при очистке звуковых файлов. Переход от удаляемого участка к остальной части звукового файла надлежит по возможности производить как можно ближе к точке пересечения волны с осью X. Таким образом при воспроизведении звукового файла удастся исключить появление резких перепадов громкости звука (рис. 19.7).

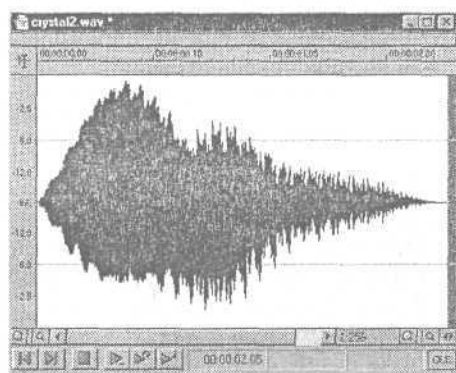


**Рис. 19.7.** После удаления выбранной части волна начинается с нулевого значения амплитуды

Альтернативным вариантом является применение эффекта затухания на коротком интервале времени (длительностью в несколько миллисекунд, что соответствует нескольким выборкам) после удаления ненужного участка. Если вы отметили отрезок в начале или конце волны и задали эффект затухания, звуковой редактор сделает так, что громкость выбранного участка будет постепенно уменьшаться (рис. 19.8). Эффект затухания позволяет снизить продолжительность какого-либо звука, например звука, возникающего при ударе в гонг.



Перед применением затухания



После применения затухания

**Рис. 19.8.** Вид звуковой волны до и после применения эффекта затухания

## Нормализация параметров звуковых файлов

Одним из наиболее важных параметров при работе со звуком является уровень громкости. Требуемая громкость устанавливается при отключении всех лишних звуков. Создавая звуковое оформление для своего фильма, не забудьте убедиться, что все звуки слышны хорошо.

При настройке звуковых файлов достаточно сложно предугадать, какова будет громкость звука в той или иной ситуации. Однако в среде Flash имеется одна очень удобная особенность — управлять уровнем громкости звука можно программно. В связи с этим необходимо, чтобы при воспроизведении звуков их

уровни громкости изменялись пропорционально. Это достигается путем *нормализации* звука. В процессе нормализации для звукового файла задается определенный диапазон уровней громкости звука. При этом звук не ограничивается и не искажается.

#### ПРИМЕЧАНИЕ

Ограничение и/или искажение происходит тогда, когда громкость звука выходит за рамки возможностей воспроизводящего устройства. В некоторых аналоговых приложениях искажения могут приводить к интересным эффектам, искажение же в цифровом аудио крайне нежелательно. Исключением является случай намеренного искажения звука для получения какого-либо необычного эффекта.

Нормализация звука производится следующим образом.

1. Откройте звуковой файл, который нужно нормализовать, и примените команду Edit ▶ Select All (Правка ▶ Выделить все).
2. Выберите команду Normalize (Нормализовать). В зависимости от используемого редактора команда нормализации может быть активизирована различными способами. Обычно она находится в меню Effects (Эффекты) или Process (Обработка).
3. В появившемся диалоговом окне укажите уровень нормализации. Обычно выбирается значение между 96 % и 98 %. Это означает, что максимальный уровень громкости нормализованного звукового файла составит от 96 % до 98 % громкости исходного файла.
4. Сохраните звуковой файл.

Различия между нормализованными и ненормализованными звуковыми файлами продемонстрированы на рис. 19.9.

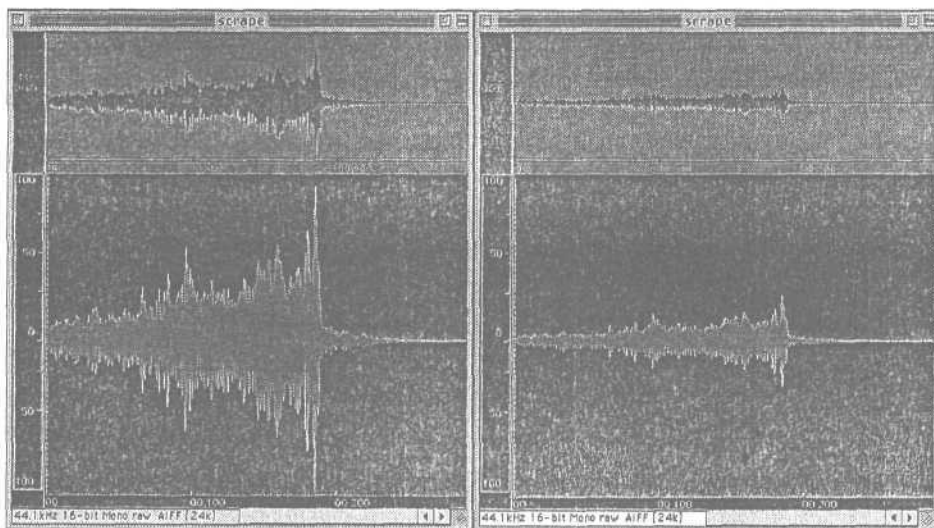


Рис. 19.9. Слева представлена запись нормализованного звука, справа — ненормализованного

## Планирование использования звуков

При создании звуковых эффектов вы должны четко представлять, какую роль они будут играть в вашем фильме. Уделяя много времени повышению визуальной интерактивности своих Flash-продуктов, разработчики нередко забывают, что и при работе со звуком необходим творческий подход. Если вы хотите создать привлекательный Web-узел или интерактивную анимацию, наряду с графическими возможностями Flash следует обратить внимание на некоторые невидимые аспекты.

Звук является одним из ключевых элементов создания *иммерсии*, то есть ощущения «погружения» в какую-либо среду. Этот эффект позволит удержать посетителя на вашем Web-узле и вызвать еще больший интерес к вашему фильму. Звук в значительной степени способствует появлению подобного ощущения, привлекая внимание зрителя к изображению и снабжая его дополнительной информацией об увиденном на экране. С помощью звука можно усилить впечатления зрителя от просмотра различных сцен, а также определенным образом повлиять на его настроение.

Чтобы в полной мере задействовать возможности звукового сопровождения, следует предварительно спланировать его использование в фильме. Определитесь, где будут звучать диалоги, а где — музыка, и какие звуковые эффекты наиболее точно смогут отразить характер тех или иных сцен.

Допустим, вы хотите создать фильм в виде детской книги, содержащей собрание разных коротких рассказов и стихов. Какие звуки и в каких фрагментах фильма вы будете использовать? Вариантов может быть множество. Главное, чтобы звуковое сопровождение способствовало наибольшему «погружению» в события, о которых повествует фильм, превращая его просмотр в увлекательное путешествие. Можно, например, использовать звуки колокольчиков, бубенчиков, эффекты жужжания и т. д., связав их с различными кнопками или навигационными элементами. Или же сопроводить повествование классической музыкой, а также задействовать по мере необходимости такие звуковые эффекты, как стук в дверь или звук шагов. Кроме того, можно связать звуки с элементами управления для подтверждения того, что был произведен щелчок мышью или перемещение курсора в области элемента управления, подобно тому как это осуществляется графическими средствами, скажем посредством всплывающих подсказок.

Возможности применения подобных эффектов неограниченны. Следует лишь помнить о планировании. Так же, как при выборе цветовых палитр, шрифтов и других графических параметров, перед тем как добавлять звук, необходимо тщательно обдумать, где и в каких ситуациях его использование будет наиболее уместным и выигрышным. Работу со звуком лучше не откладывать на потом, а выполнять в процессе создания фильма.


## Работа с циклическими звуками

Циклические звуки являются одним из наиболее полезных инструментов, предоставляемых средой Flash. Под циклическим звуком подразумевается музыкальная фраза или иной звуковой фрагмент, который постоянно повторяется.

Существует мнение, что использование циклических звуков является плохим тоном. Это объясняется не всегда удачным их применением. Однако грамотное использование различных приемов работы с такими звуками позволяет внести в фильм разнообразие.

Прежде всего необходимо отметить, что в среде Flash работать с циклическими звуками очень легко. Другое преимущество использования звуков подобного типа состоит в том, что они являются компактными. Цикл может занимать всего несколько килобайт, но если циклический звук преобразовать в обычный, его размер достигает нескольких сотен мегабайт, что может отразиться на окончательном размере вашего фильма. Однако короткие циклы, то есть такие, которые содержат всего несколько звуков, быстро надоедают. Поэтому по возможности старайтесь создавать более содержательные звуковые циклы.

Рассмотрим звуковые циклы с точки зрения музыки. Музыкальная фраза обычно состоит из нескольких нот или тактов. Неважно, сколько их, однако если это число *четное*, то циклическое повторение такой фразы более заметно. Напротив, музыкальная фраза, состоящая из *нечетного* числа нот или тактов, воспринимается гораздо лучше. Иначе говоря, наиболее эффективным средством замаскировать повторение звукового фрагмента является использование нечетного числа тактов.

 Откройте файл `loops.swf`, который находится на прилагаемом к этой книге компакт-диске в каталоге Chapter 19. Вы увидите три зеленые кнопки, при нажатии на которые будут запущены соответствующие звуковые файлы: `oneBar`, `fourBar` и `fiveBar`. Внимательно прослушайте эти записи и обратите внимание на различия в их звучании. Заметьте, как при проигрывании музыкальной фразы, состоящей из одного такта, сразу же проявляется циклическость звука. Фраза, содержащая четыре такта, является более благозвучной. И, наконец, прослушайтесь к звучанию музыкальной фразы из пяти тактов: циклическость такого звукового фрагмента наименее заметна.

Создавать звуковые циклы довольно легко, но это требует некоторой подготовки. Вам понадобится звуковой редактор, который поддерживает функцию воспроизведения циклических звуков. С этой целью можно воспользоваться такими редакторами ОС Windows, как Sound Forge и Cool Edit Pro, или редакторами SoundEdit 16 и Peak на компьютерах типа Macintosh. Указанные приложения включают соответствующие инструменты, которые позволяют создать качественный звук без шелчков, хлопков и других дефектов редактирования. Кроме того, для создания циклических звуков и основанной на них музыки предусмотрена специальная программа Acid, разработанная компанией Sonic Foundry. Более подробная информация о работе этих программ содержится в главе 26.

## Разработки

Звук играет значительную роль при создании приложений в среде Flash. Удачным примером этому служит Web-узел группы художников и музыкантов Nineteen Point Five Collective ([www.npfc.org](http://www.npfc.org)), которые специализируются на создании различных мультимедийных произведений (музыки, видео, трехмерных изображений и т. п.), а также на разработке интерактивных Web-страниц.

Web-узел создай в среде Flash и является превосходным справочником, знакомящим посетителей с различными музыкальными группами (рис. 19.10). Обратите внимание на удачное сочетание дизайна с музыкальным сопровождением, позволяющее наиболее выгодно представить ту или иную музыкальную группу, отразив ее индивидуальность.

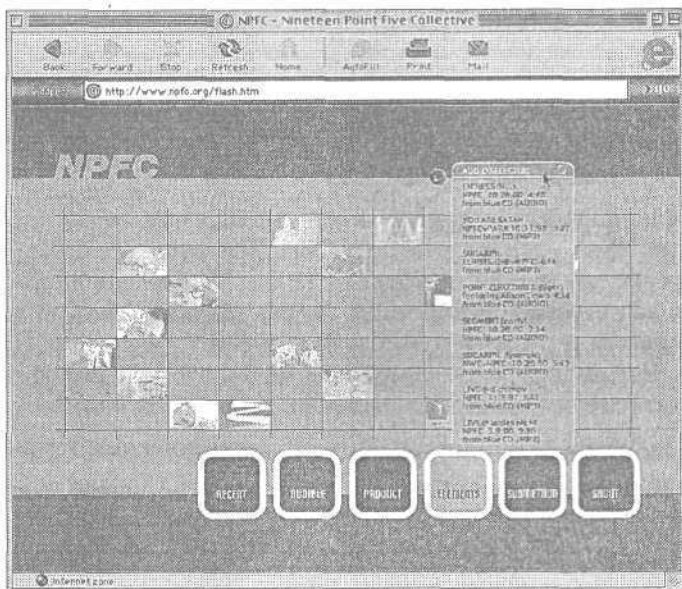


Рис. 19.10. Web-узел компании Nineteen Point Five Collective (www.npfc.org)

## Резюме

В этой главе вы изучили, как с помощью звукового оформления усовершенствовать свои Flash-фильмы, ознакомились с основными техническими терминами, такими как частота и глубина выборки, стерео- и монозвук. Перечисленные параметры, используемые при цифровой записи звука, служат для представления звуковой волны и определяют качество звука. Наряду с этим вы получили информацию о различных форматах звуковых файлов, поддерживаемых в среде Flash.

Flash предоставляет достаточно ресурсов для создания качественного звука. Выбрав необходимые звуковые файлы, их следует подготовить к использованию в видеопродукции. Такая подготовка осуществляется как на техническом, так и на концептуальном уровнях. В настоящей главе было показано, как музыка и звуковые эффекты способны преобразить ваш проект.



## Глава 20

# Основы работы с аудиофайлами в среде Flash

Использование различных звуковых эффектов, музыки, диалогов и т. д. позволяет значительно усовершенствовать продукты, разработанные в среде Flash. Если вы уже ознакомились с предыдущей главой, то можете приступить к работе с цифровыми аудиофайлами в Flash.

Как и в случае использования других интерактивных приложений, при работе со звуком в среде Flash следует руководствоваться определенными правилами, которые будут рассмотрены в настоящей главе. Однако эти правила не стоит воспринимать как строгие предписания. Лучше рассматривать их как обычные технические приемы, которыми нужно овладеть для успешного решения поставленных задач. Наряду с этим не следует забывать, что при работе со звуком не менее важной составляющей успеха является творческий подход.

В настоящей главе предлагаем ознакомиться со следующими темами:

- ▶ внедрение звуков в среду Flash и добавление звукового сопровождения к фильмам;
- ▶ привязка звуков к кнопкам;
- ▶ воспроизведение звука;
- ▶ использование звуковых эффектов;
- ▶ экспортирование и сжатие звуковых файлов.

## Импортирование звуковых файлов

Чтобы использовать звуковые файлы в среде Flash, прежде всего их необходимо добавить в библиотеку. Процесс помещения аудиофайлов в библиотеку называется импортированием. Типы файлов, которые могут быть импортированы, определяются операционной системой, установленной на компьютере. В ОС Windows чаще всего используются файлы типа WAV, а в ОС Macintosh — файлы типа AIFF. На обеих платформах поддерживаются файлы типа MP3. Более детальную информацию о типах файлов, используемых в среде Flash, можно найти в главе 19.

Импортирование звуковых файлов можно осуществить следующим образом:

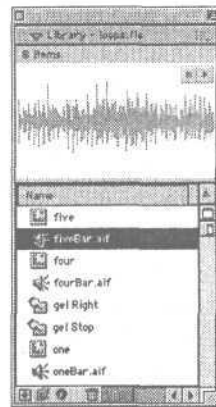
1. Активизируйте команду File > Import (Файл ▶ Импортировать) или нажмите комбинацию клавиш `Cmd/Ctrl+R`. Появится диалоговое окно Import (Импорт).
2. В раскрывающемся списке выберите тип All Sound Formats (Все звуковые форматы), тем самым отобразив в окне импортирования только звуковые файлы. Найдите каталог, где находятся файлы, которые вы собираетесь импортировать.

Чтобы выбрать несколько файлов, используйте клавишу Shift (на компьютерах Macintosh) или Ctrl (при работе с ОС Windows).

### ПРИМЕЧАНИЕ

Звуковые файлы можно импортировать как группами, так и по отдельности.

3. Завершив выбор файлов, нажмите кнопку Open (Открыть). Все отмеченные файлы будут импортированы в библиотеку. Вы можете убедиться в этом, просмотрев ее содержимое. Чтобы отобразить панель библиотеки, активизируйте команду Window ▸ Library (Окно ▸ Библиотека). Все звуковые файлы, которые были в нее импортированы, помечены значком с изображением динамика. При выборе одного из файлов в верхней части окна библиотеки отобразится форма соответствующей звуковой волны. Щелкнув на кнопке Play (Воспроизвести) в правом верхнем углу, вы сможете прослушать данный файл.



Звуковые файлы, импортированные в библиотеку, допускаются использовать в фильме многократно. Подобные файлы также могут являться частью библиотек, общих для нескольких фильмов. Совместно используемые библиотеки подробно рассматриваются в главе 7.

## Включение звука в фильм

Импортирование аудиофайлов — это только начало. Далее звуковые файлы следует поместить из библиотеки во временную шкалу фильма, с тем чтобы сделать их его частью. Для вставки звуков во временную шкалу используйте панель Properties (Свойства), представленную на рис. 20.1. Посредством этой панели можно задать различные параметры звука для фильма.

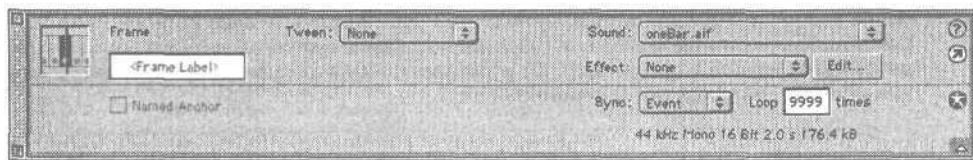


Рис. 20.1. Панель Properties предназначена для управления параметрами звука

Чтобы добавить звук в фильм, проделайте следующие операции:

1. Создайте новый слой и назовите его, скажем, sound. Это необязательная операция, однако она позволит сохранить звуковые файлы на отдельных слоях. Слоям желательно присваивать имена, которые характеризовали бы содержащиеся на них файлы (например, introRiff). Кроме того, если звуки отделены от других элементов фильма, ими гораздо легче управлять.
2. Выделите кадр, с которого должно начаться воспроизведение звука, после чего активизируйте команду Insert ▸ Keyframe (Вставить ▸ Ключевой кадр) для задания нового ключевого кадра.

**ПРИМЕЧАНИЕ**

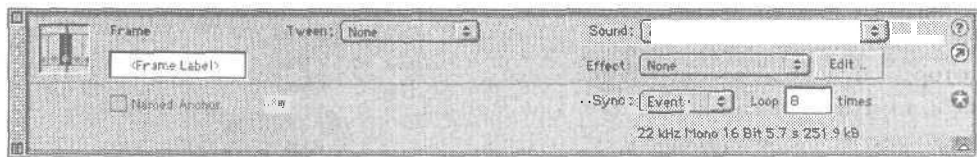
Несмотря на то что Flash может одновременно воспроизводить только восемь звуков, число звуковых слоев фильма неограниченно. Эта особенность делает Flash многотрековым аудиоприложением. Чтобы больше узнать о возможностях Flash и многотрековом аудио, рассмотрите практическое задание 6.

3. Отобразите панель Properties, воспользовавшись командой Window ▸ Properties (Окно ▸ Свойства). Другой способ вызвать эту панель — нажать либо клавишу **Cmd** (при работе на компьютерах Macintosh), либо выбрать команду Properties из контекстного меню, щелкнув правой кнопкой мыши (при работе с ОС Windows) на ключевом кадре. К этому моменту вы уже должны импортировать файлы в свой фильм. Если это не сделано, вернитесь к предыдущему разделу главы.
4. Из списка Sound (Звук) выберите нужный звуковой файл. В окне свойств отобразятся его атрибуты: частота и глубина выборки, продолжительность звука в секундах, а также размер файла. Указанный файл будет активизирован при воспроизведении выбранного кадра.
5. В списке Effect (Эффект) вы можете выбрать какой-либо звуковой эффект. Более подробная информация о звуковых эффектах содержится в разделе «Создание аудиоэффектов» этой главы.
6. В списке Sync (Синхронизация) выберите необходимый тип синхронизации. В частности, элементом Event (Событие) данного списка обычно задается синхронизация коротких звуков с изображением, например ударов, щелчков и т. д. Такие звуки воспроизводятся до тех пор, пока не будут выключены или не прекратятся сами. При помощи элемента Stream (Поток) этого списка устанавливается покадровая синхронизация изображения со звуком.

**ПРИМЕЧАНИЕ**

В отличие от звуков типа Event, звуки типа Stream воспроизводятся вместе с кадрами, которые они захватывают во временной шкале. Чтобы осуществить подгонку длины потокового звука, иногда требуется добавить дополнительные кадры или удалить лишние. Более подробно эти и другие элементы списка Sync описаны в главе 21. Если вы планируете использовать MP3-файлы в качестве потоковых звуков, то сначала их придется декомпрессировать в среде Flash. Об этом рассказывается в разделе «Экспортирование и сжатие звуковых файлов» данной главы.

7. В поле Loop (Цикл) укажите, сколько раз требуется воспроизвести заданный звуковой файл. По завершении ввода данных на экране появится окно, подобное представленному на рис. 20.2.



**Рис. 20.2.** Для звука, записанного в файле drumGroove.aif, выбран тип синхронизации Event и задано число повторений 8

## Остановка воспроизведения звука

Обычно воспроизведение звука прекращается с окончанием просмотра кадров (если это звук типа Stream) или при завершении выполнения цикла. Это зависит от продолжительности звукового фрагмента, а также от параметров, выбранных при первом воспроизведении звука в фильме. Однако в некоторых случаях возникает необходимость принудительной остановки воспроизведения звука или серии звуков.

Чтобы прекратить воспроизведение определенного звука, сделайте следующее:

1. Щелкните на кадре в звуковом слое, где вы хотите выключить звук.
2. Активизируйте команду **Insert ▶ Keyframe** для задания нового ключевого кадра.
3. Воспользовавшись командой **Window ▶ Properties**, отобразите панель свойств.
4. В списке **Sound** выберите имя звукового файла, воспроизведение которого следует прекратить, а в списке **Sync** — элемент **Stop (Стоп)**. Воспроизведение звука будет прервано на заданном ключевом кадре.

Выключить все звуки можно таким образом:

1. Выберите в вашем фильме кадр, где вы желаете остановить воспроизведение всех звуков. Для выключения звука в фильме обычно создают специальную кнопку.
2. Откройте панель **Actions (Действия)** для любого кадра или кнопки, которую вы хотите использовать. В режиме **Normal (Обычный режим)** перетащите команду **stopAllSounds 0** из категории **Actions ▶ Movie Control (Действия ▶ Элементы управления фильмом)** в область сценария. В режиме **Expert (Режим эксперта)** эту команду можно не только перетащить, но и ввести в данную область вручную. (Режимы **Normal** и **Expert** выбираются в меню кнопки со всплывающей подсказкой **View Options** в области сценария, расположенной в правой части панели **Actions**.) При воспроизведении указанного кадра или нажатии кнопки все звуки будут выключены.

## Привязка звука к кнопкам

Привязка звуков к различным кнопкам позволяет повысить интерактивность создаваемого продукта. Подобно визуальным индикаторам, которые изменяются при переходе кнопки из одного состояния в другое, звуковые эффекты помогают установить обратную связь с пользователем. Например, вы можете создать кнопку с изображением двери и связать с ней звук, имитирующий открытие двери, который будет раздаваться при переходе кнопки в нажатое состояние, а также звук, имитирующий щелчок замка, появляющийся при следующем нажатии на кнопку. Это сделает диалог интереснее и приведет зрителей в состояние напряженного ожидания.

Привязка звука к кнопке — очень простой и эффективный способ привлечь аудиторию к вашей разработке. Производится это следующим образом:

1. Импортируйте в фильм звук для вашей кнопки. О том, как это сделать, мы уже рассказывали (см. раздел «Импортирование звуковых файлов» этой главы).

### СОВЕТ

Большой выбор звуков для кнопок предоставляет встроенная звуковая библиотека, которую можно активизировать командой Window ▶ Common Libraries ▶ Sounds (Окно ▶ Общие библиотеки ▶ Звуки). Выберите подходящий звук и перетащите его в панель библиотеки вашего фильма.

2. Выделите кнопку, с которой вы хотите связать звук, и в меню Edit (Правка) выберите элемент Edit Symbols (Правка символов). Flash перейдет в режим редактирования символов для указанной кнопки. Появится временная шкала кнопки со следующими возможными состояниями ключевого кадра (то есть событиями кнопки): Up, Over, Down и Hit. Для более детального ознакомления с событиями кнопки обратитесь к главе 13.
3. Добавьте новый слой во временную шкалу кнопки и назовите его sounds.
4. Выберите событие Over кнопки, после чего воспользуйтесь командой Insert ▶ Keyframe для задания нового ключевого кадра.
5. Отобразите на экране панель Properties, активизировав команду Window ▶ Properties. Так же, как вы добавляли звук в главную временную шкалу, добавьте его во временную шкалу кнопки (рис. 20.3). С этой целью в списке Sound укажите необходимый звуковой файл, в списке Effect выберите желаемый эффект, а в списке Sync — элемент Start (Старт). Более подробно эта команда рассматривается в главе 21.
6. В меню Edit выберите команду Edit Document (Правка документа) для возврата к главной временной шкале. Опробовать кнопку со звуком можно в режиме редактирования фильма, активизировав в меню Control (Управление) команду Enable Simple Buttons (Разрешить проверку кнопок). Если же вы хотите проверить результаты работы во время просмотра фильма, воспользуйтесь командой Control ▶ Test Movie (Управление ▶ Проверить фильм). Наведите указатель мыши на кнопку, чтобы получить ее звуковую реакцию на это событие.

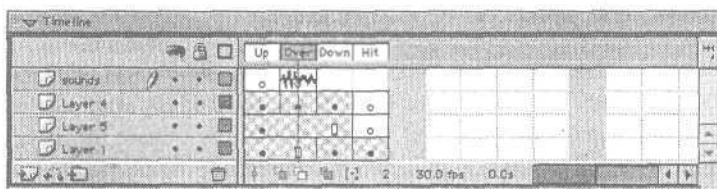


Рис. 20.3. Временная шкала кнопки со звуком, заданным для события Over

Аналогичным образом звук можно связать и с остальными событиями кнопки, например с событием Down. Следуя описанным выше инструкциям, создайте новый ключевой кадр для события Down и свяжите с ним нужный звук. Поскольку два кадра представляют разные события, они будут действовать раздельно —

один при наведении указателя мыши на кнопку (событие Over), а другой — при нажатии на нее (событие Down).

---

### СОВЕТ

При работе с кнопками очень важно очистить звуковые файлы так, чтобы они не имели так называемой «мертвой» зоны в начале файла. Если этого не сделать, при нажатии на кнопку звук будет воспроизводиться с задержкой.

---

## Создание аудиоэффектов

Одним из наиболее интересных моментов при работе со звуком является использование аудиоэффектов. Аудиоэффектами можно назвать изменение уровня громкости звука, а также позиции панорамирования звука. Подобные изменения вносят разнообразие в звучание аудиофайлов. Вы можете использовать эффекты для получения нужного вам звука, например имитировать звук от близко или далеко расположенного источника либо создать «движущийся» звук. Путем создания эффекта присутствия аудиоэффекты делают звук трехмерным.

Используя звуковой редактор, можно добавить аудиоэффекты в звуковой файл до его импортирования в среду Flash. Однако создание аудиоэффектов непосредственно в среде Flash более предпочтительно и имеет ряд преимуществ. В частности, добавление звуковых эффектов после помещения аудиофайла в фильм избавляет вас от импортирования многочисленных копий данного файла с разными эффектами. Кроме того, создание стереоэффектов в среде Flash позволяет уменьшить размер аудиофайла с панорамированным звуком. Как было сказано в предыдущей главе, стереозвуки (то есть воспроизводимые по двум каналам), требуют вдвое больше места в библиотеке, чем монозвуки. Используя звуковые эффекты Flash, можно применить панорамирование к монозвуку и получить те же результаты, но при меньшем размере звукового файла. Неплохо, правда?

---

### ПРИМЕЧАНИЕ

Панорамирование имеет наибольший эффект, когда применяется для монозвука. Например, чтобы создать продолжительный звук, начинающийся в левом канале и заканчивающийся в правом, вы можете использовать монозвук, а Flash позаботится о его панорамировании. Альтернативой является создание панорамированного звука в звуковом редакторе. В этом случае звук будет таким же, однако размер файла окажется вдвое большим, поскольку файл будет включать информацию как для левого, так и для правого каналов.

---

Чтобы применить звуковые эффекты, проделайте следующие операции на панели Timeline.

1. Выделите ключевой кадр, с которого начинается воспроизведение звука, и выберите команду Window ► Properties, чтобы отобразить на экране панель свойств, если она не открыта.

2. В поле Sound панели свойств появится имя звукового файла, в поле Sync — тип его синхронизации, а в поле Loop — количество повторений. Здесь также находится поле со списком Effect, который содержит несколько элементов:
  - ▷ Left/Right Channel (Левый/Правый канал) — воспроизведение звука либо в левом, либо в правом каналах;
  - ▷ Fade Left to Right (Плавный переход слева направо) и Fade Right to Left (Плавный переход справа налево) — панорамирование звука из одного канала в другой во время воспроизведения;
  - ▷ Fade In (Плавное возрастание) — постепенное возрастание уровня громкости звука от нулевого значения до максимального;
  - ▷ Fade Out (Плавное затухание) — постепенное снижение уровня громкости звука от максимального значения до нуля;
  - ▷ Custom (Пользовательский) — вызов диалогового окна Edit Envelope (Правка огибающей), используемого для создания собственных аудиоэффектов (об этом речь пойдет далее).

При использовании перечисленных эффектов для циклических звуков необходимо учитывать некоторые особенности. К примеру, эффект Fade Left to Right работает только при первом прохождении цикла, после чего звук воспроизводится только на правом канале. Поработав над этим, вы можете создать собственный эффект для циклических звуков. Более подробно данная тема обсуждается в следующем разделе.

3. Чтобы оценить созданный эффект в контексте фильма, выберите команду Control ▶ Test Movie.

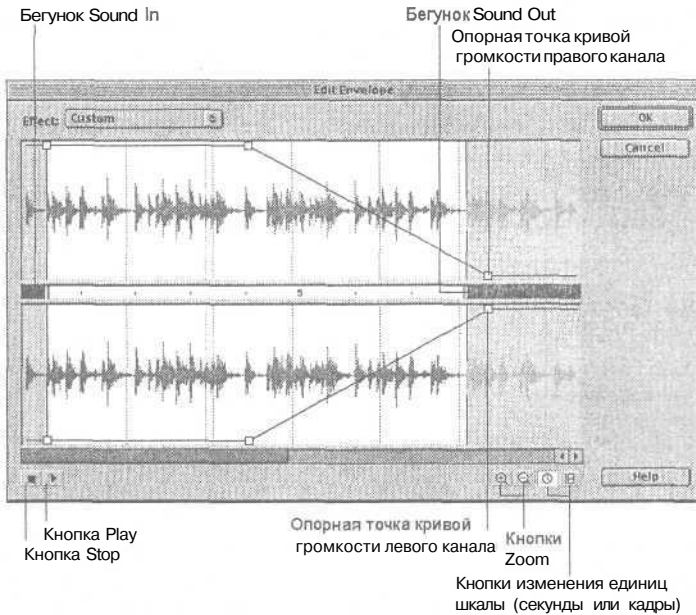
Если ваш фильм представляет собой линейную анимацию (то есть анимацию, определяемую лишь расположением ключевых кадров во временной шкале) и не требует дополнительных команд ActionScript, аудиоэффекты можно прослушать в режиме Movie Editing (Правка фильма). С этой целью следует выбрать команду Control ▶ Play (или нажать клавишу Return/Enter).

## Создание эффектов с помощью окна Edit Envelope

Возможно, вам потребуются более сложные аудиоэффекты, чем те, которые предлагает список Effects на панели свойств. Создать собственные звуковые эффекты в Flash можно путем редактирования громкости звука во времени и использования различных комбинаций изменения громкости для левого и правого каналов.

Чтобы создать собственный звуковой эффект, выполните следующие действия:

1. Выделите в слое звука ключевой кадр, с которого вы хотите начать редактирование. Если на экране нет панели свойств, активизируйте команду Window ▶ Properties для ее отображения.
2. В панели свойств из списка Effect выберите элемент Custom или нажмите кнопку Edit. На экране появится диалоговое окно Edit Envelope (рис. 20.4).



**Рис. 20.4.** Диалоговое окно Edit Envelope

#### ПРИМЕЧАНИЕ

Даже при работе с монозвукоем вы имеете возможность создавать стереозвукоев, распределив при редактировании звук по двум каналам.

3. Для регулировки громкости звука во времени следует нажать на метку кривой уровня громкости каждого из каналов и перетащить ее в нужное место. Например, чтобы добиться эффекта перехода звука из одного канала в другой и обратно, щелкните на кривой громкости для левого канала в конце звуковой волны. Точка, в которой произведен щелчок, станет опорной. Таким же способом создайте опорную точку посередине линии, после чего перетяните ее вниз. Сделайте то же самое для правого канала, но в зеркальном отображении (рис. 20.5). Требуемый эффект готов.
4. После того как вы внесете изменения, нажмите кнопку ОК для возврата к главной временной шкале.

Помимо этого, в диалоговом окне Edit Envelope имеются другие элементы.

- ▶ Кнопки Play и Stop служат для прослушивания эффекта.
- ▶ Кнопка Zoom (Масштаб) предназначена для более детального рассмотрения звуковой волны. Это может пригодиться для создания коротких динамичных звуковых эффектов.
- ▶ Элементы Fade In и Fade Out списка Effect используются для очистки начала или конца файла. Эти команды могут быть полезны для создания звуков, которые будут привязаны к кнопкам.



- ▶ Если вы применяете эффект к циклическому звуку, все повторения цикла, кроме первого, будут выделены серым цветом. Для создания эффектов на интервале, включающем все повторения, следует отредактировать график огибающей.

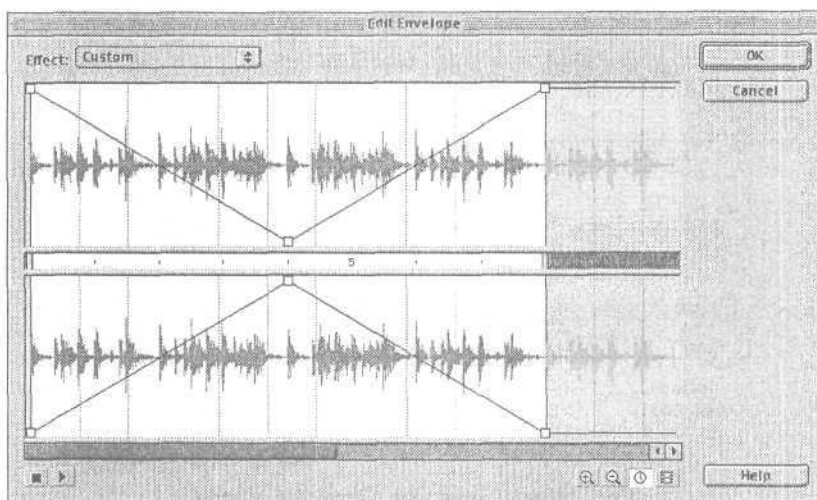


Рис. 20.5. Такая конфигурация обеспечивает эффект панорамирования

## Экспортирование и сжатие звуковых файлов

На заключительном этапе создания фильма следует выполнить его *публикацию*, то есть такую обработку, после которой фильм будет готов для просмотра вне среды Flash. В процессе публикации все звуковое оформление фильма экспортируется и помещается в выходной файл. Как правило, при экспортировании звукового файла происходит изменение его формата на такой, который поддерживается вне среды Flash. Поскольку звуковые файлы оказывают значительное влияние на размер окончательного продукта, процесс экспортирования *обычно* включает операцию их *сжатия*.

Сжатие производится с целью уменьшения размеров аудиофайла. Однако эта операция может неблагоприятно повлиять на звуковой файл. Поскольку размер файла и качество звука — параметры взаимозависимые, сжатие файла может отразиться на качестве звука.

Хотя при сжатии звуковой файл становится более компактным, некоторые особенности звука утрачиваются. В отдельных случаях это может испортить такие характеристики, над которыми вы долго работали. Однако не огорчайтесь раньше времени. Flash предоставляет различные варианты сжатия и методы аппроксимации, использование которых позволяет найти наиболее оптимальное соотношение между размером файла и его качеством.

## Задание индивидуальных параметров экспортирования

Как правило, лучше использовать индивидуальные настройки для каждого звукового файла. Это следует делать по двум причинам. Во-первых, звуки бывают разные. Некоторые звуки являются приемлемыми даже при низком уровне качества или значительном сжатии. Во-вторых, для коротких или не очень важных звуков можно задать максимальное сжатие с минимальным качеством.

Чтобы задать параметры экспортирования для звуковых файлов, проделайте следующее:

1. Отобразите панель библиотеки, активизировав элемент Library в меню Window.
2. В библиотеке щелкните правой клавишей мыши на имени звукового файла, после чего выберите команду Properties из появившегося контекстного меню на панели библиотеки. Альтернативой является выполнение двойного щелчка мышью на значке динамика. В результате появится окно Sound Properties (Свойства звука).

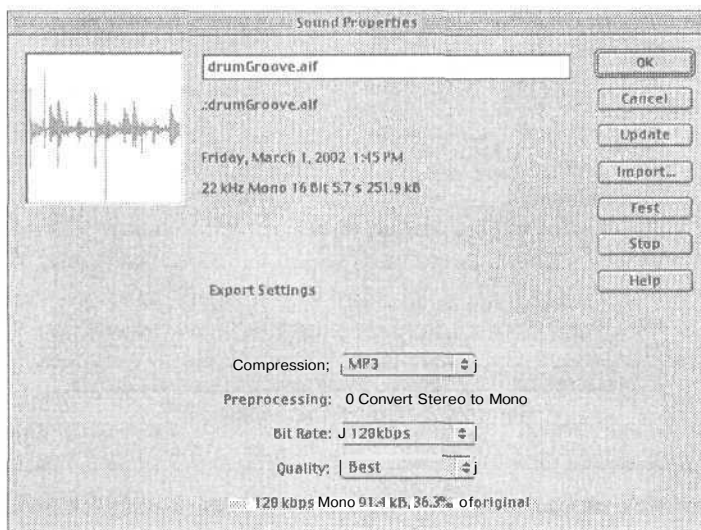


Рис. 20.6. Окно Sound Properties

3. Если звуковой файл был изменен вне среды Flash перед его импортированием, нажмите кнопку Update (Обновить). Это позволит получить более новую версию файла.
4. В списке Compression (Сжатие) выберите способ сжатия и задайте необходимые вам параметры сжатия. О типах сжатия и их параметрах вы узнаете из следующего раздела.
5. Нажав кнопку Test (Проверка), прослушайте звук и, если нужно, измените параметры сжатия. По завершении работы щелкните на кнопке OK для возврата в ваш фильм. В итоге каждый указанный вами звуковой файл будет экспортирован с заданными параметрами.

## Выбор параметров сжатия

В среде Flash существует несколько типов сжатия. При указании одного из типов сжатия внизу диалогового окна Sound Properties появляются его параметры (частота, глубина выборки и т. д.). Если эти установки изменить, информация о размерах сжатого файла будет обновлена.

### Сжатие по умолчанию

Чтобы задать сжатие по умолчанию, установите в поле Compression значение Default (По умолчанию), тем самым выбрав параметры сжатия, заданные в диалоговом окне Publish Settings (Параметры публикации). Для проверки параметров публикации вашего фильма воспользуйтесь командой File ▶ Publish Settings. В открывшемся окне Publish Settings (рис. 20.7) вы можете задать параметры сжатия для звуков типа Stream и Event.

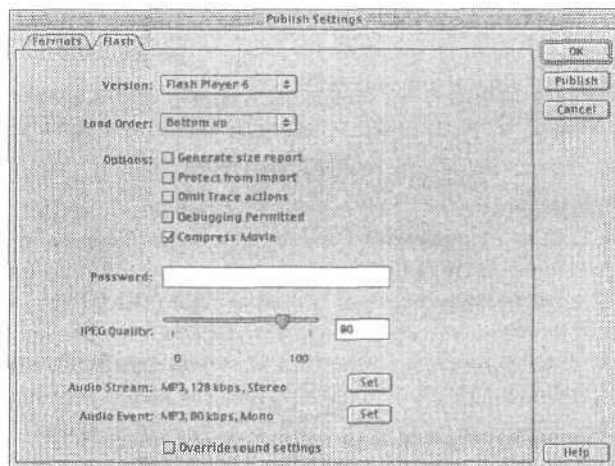


Рис. 20.7. Окно Publish Settings

### ПРИМЕЧАНИЕ

Об установках, касающихся экспортирования, вы узнаете в следующем разделе.

Если на вкладке Flash установлен флажок Override Sound Settings (Заменять параметры звука), Flash будет экспортировать все звуковые файлы со значениями параметров, определенными в окне Publish Settings, невзирая на то, какие значения параметров заданы для этих файлов в библиотеке.

### Сжатие ADPCM

Сжатие ADPCM (Adaptive Differential Pulse-Code Modulation) является наиболее подходящим типом сжатия для коротких звуков, которые могут хорошо восприниматься даже при низком качестве воспроизведения.

Чтобы задать ADPCM-сжатие, выполните следующие действия.

1. Из списка Compression выберите тип сжатия ADPCM.

2. Если вы хотите преобразовать стереозвук в монозвук, установите флажок **Convert Stereo to Mono** (Преобразовать стерео в моно).
3. В списке **Sample Rate** задайте частоту выборки. Предлагаемые для выбора значения находятся в диапазоне 5–44 кГц. Как уже говорилось в главе 19, понижение частоты выборки приводит к ухудшению качества звука.
4. В списке **ADPCM Bits** выберите количество битов от 2 до 5. Чем больше это значение, тем выше качество.

## MP3-сжатие

При записи файлов на компакт-диски широко используется MP3-сжатие, при котором звуковые файлы представляются в компактном цифровом формате. Этот тип сжатия является наиболее подходящим для музыки, диалогов, продолжительных аудиоэффектов и различных **Stream-синхронизированных** звуков в вашем фильме. MP3-сжатие сочетает в себе хорошее качество и сравнительно небольшой размер звукового файла. Данный тип сжатия появился только в версии **Flash 4**, поэтому проверьте на вашем компьютере наличие программы **Flash Player**, которая поддерживает формат MP3.

Для использования MP3-сжатия выполните следующие действия.

1. Из списка **Compression** выберите тип сжатия MP3.
2. В списке **Bit Rate** укажите глубину выборки. Предлагаемые для выбора значения находятся в диапазоне 8–160 кбит/с. Чем больше значение глубины выборки, тем выше качество звука. Однако следует помнить, что, улучшая качество, вы увеличиваете размер звукового файла. Например, установка значения 112 кбит/с будет стоить вам лишнего места на компакт-диске, но зато вы получите звук отличного качества. Установка же значения 56 кбит/с позволит добиться компромисса между размером файла и качеством звука.
3. Выбрав глубину выборки 20 и выше, вы сможете преобразовать стереозвук в монозвук, установив флажок **Convert Stereo to Mono**.
4. Из списка **Quality** (Качество) выберите параметр, определяющий качество звука:
  - ▷ **Fast** — низшее качество с максимальным сжатием. Эту команду нежелательно применять для основных звуков вашего фильма.
  - ▷ **Medium** — среднее качество со средней степенью сжатия.
  - ▷ **Best** — наилучший вариант для музыки и основных звуков вашего фильма. В результате выбора данного параметра сжатие будет меньше, чем в случае установки параметра **Medium**, но при этом обеспечивается наилучшее качество MP3-файлов (которое, разумеется, ограничено заданным значением глубины выборки).

## Грубое сжатие

Выбор данного типа не приводит к сжатию файла. Он применяется для преобразования стереозвука в монозвук и/или уменьшения частоты выборки.

Чтобы задать такой способ сжатия, выполните следующие действия.

1. В списке **Compression** укажите значение **Raw**.

2. Если вы хотите преобразовать стереозвук в монозвук, установите флажок Convert Stereo to Mono.
3. Из списка Sample Rate выберите частоту выборки. Предлагаемые значения находятся в диапазоне 5–44 кГц. Понижение частоты выборки, как известно, ухудшает качество звука.

### Сжатие речи

**MX** В приложение Flash MX был добавлен тип сжатия, специально предназначенный для применения при записи речи. Использование такого типа сжатия для записи повествования или диалога позволяет получить файлы достаточно хорошего качества и небольшого размера.

Задать указанный тип сжатия можно, выполнив следующие действия.

1. Отметьте элемент Speech (Речь) в списке Compression.
2. В списке Sample Rate укажите частоту выборки. Предлагаемые для выбора значения находятся в диапазоне 5–44 кГц. Понижение частоты выборки ухудшает качество звука. Не забывайте тестировать звуки при изменении параметров сжатия.

### ПРИМЕЧАНИЕ

Стереозвук при сжатии речи не используется. Все звуки автоматически преобразуются в моно.

### Параметры сжатия для фильмов в формате QuickTime

Flash позволяет публиковать фильмы в формате QuickTime, что предпочтительнее формата Shockwave. Формат QuickTime дает возможность устанавливать дополнительные параметры для улучшения качества звука. Следует обратить внимание, что методы сжатия, описанные здесь, подходят *только* для QuickTime-фильмов.

Работая с фильмами, представленными в формате QuickTime, можно более гибко управлять звуком. Здесь нет ограничений на количество аудиотреков, поэтому вы можете использовать столько звукозаписей, сколько *потребуется*. Кроме того, в файле QuickTime все звуки будут собраны в один аудиотрек (или в два, если используется стереозвук).

Установка сжатия для QuickTime осуществляется следующим образом.

1. Выберите команду File ► Publish Settings. На вкладке Formats (Форматы) установите флажок QuickTime. При этом в окне появится дополнительная вкладка QuickTime. Перейдите на нее.
2. Для потокового звука установите флажок Use QuickTime Compression (Использовать сжатие QuickTime). В результате станет доступна кнопка Settings (Параметры).
3. Нажмите кнопку Settings, тем самым открыв окно Sound Settings (Параметры звука). В данном случае появится окно для задания параметров потоковых звуков.

В этом окне произведите необходимые **установки**. Предлагаемые для выбора параметры подробно **описаны** в табл. 20.1. Вы можете поэкспериментировать с ними, перед тем как сделать окончательный выбор.

**Таблица 20.1.** Параметры сжатия звука для фильмов в формате QuickTime

| <b>Метод сжатия</b>                            | <b>Описание</b>                                                                                                                                                                                    |
|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24-bit Integer, 32-bit Integer                 | Увеличивает глубину выборки (размер выборки) до значений 24 и 32                                                                                                                                   |
| 32-bit Floating Point<br>64-bit Floating Point | Увеличивает глубину выборки (размер выборки) до значений 32 и 64. Большинство компьютеров не могут воспроизвести звук с такой глубиной выборки, поэтому преобразуют его обратно в 8- или 16-битный |
| Alaw 2:1                                       | Маленький коэффициент сжатия; не рекомендуется                                                                                                                                                     |
| IMA 4:1                                        | Неплохой коэффициент сжатия; глубина выборки должна быть равна 16; хорош для компакт-дисков                                                                                                        |
| MACE3:1, MACE6:1                               | Устаревший метод                                                                                                                                                                                   |
| Q Design Music 2                               | Наилучший коэффициент сжатия; хорош для потоковых звуков                                                                                                                                           |
| Qualcomm PureVoice                             | Наилучший коэффициент сжатия; хорош для записи диалогов и повествований                                                                                                                            |
| ULaw 2:1                                       | Устаревший метод                                                                                                                                                                                   |

#### **СОВЕТ**

Свежую информацию о QuickTime вы можете найти на странице [www.apple.com/quicktime/products/](http://www.apple.com/quicktime/products/).

- После того как вы установили параметры сжатия, нажмите кнопку ОК в диалоговом окне Sound Settings. Затем откройте вкладку Flash. Выполните щелчок на кнопке Set (Установка), **расположенной** справа от надписи Audio Stream (Потоковый звук). Вновь появится окно Sound Settings для задания параметров потоковых звуков. Из списка Compression выберите значение Disable (Отключить), что позволит предотвратить экспортирование дополнительных экземпляров потоковых звуков. Выполните щелчок на кнопке ОК.

Проделайте те же операции с кнопкой Set, **расположенной** справа от надписи Audio Event. Это позволит отключить сжатие звуков типа Event в вашем фильме. По завершении нажмите кнопку ОК в диалоговом окне Publish Settings.

#### **СОВЕТ**

При экспортировании Flash-фильма в формате QuickTime необходимо отключить сжатие звуков, для которых выбран тип синхронизации Stream или Event, иначе в выходном файле QuickTime они будут дублироваться.

После задания этих параметров все параметры сжатия QuickTime будут применены для **потоковых** звуков любого **публикуемого** фильма формата QuickTime. Звуки типа Event публиковаться с установленными параметрами QuickTime не будут. Если вы работаете со звуками типа Event, перед публикацией вашего фильма воспользуйтесь панелью свойств и измените тип их синхронизации на Stream.

## Разработки

Звуковые и анимационные возможности делают Flash идеальной средой для создания видеоклипов. Одним из наиболее удачных примеров этого служит созданный Джоем Спарксом музыкальный Web-сериал Radiskull & Devil Doll (<http://joesparks.com>), потешные персонажи которого поют песни и рассказывают о своих фантастических приключениях (рис. 20.8). В сети Интернет вы можете найти множество других примеров использования среды Flash для создания подобных фильмов.

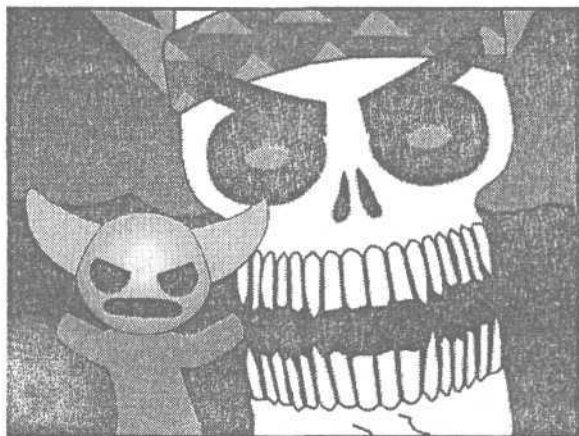


Рис. 20.8. Герои музыкального сериала Radiskull & Devil Doll

## Резюме

В данной главе освещались технические аспекты, касающиеся воспроизведения звуков в среде Flash. Здесь рассказывалось о применении звуковых эффектов, сжатии аудиофайлов, о том, как вставлять звуки в фильм и т. д.

Вы освоили технику воспроизведения и выключения звуков, а также принципы использования звуковых эффектов. Кроме того, узнали, как осуществлять сжатие звуковых файлов в требуемом формате. Полученные знания понадобятся вам при дальнейшей работе в среде Flash.

## Глава 21

# Синхронизация звука и анимации

Ярким примером анимации являются мультфильмы компании Warner Brothers. При их просмотре слышны разнообразные звуки: музыка, диалоги персонажей или их возгласы, стук, взрывы и т. д. Они оживляют персонажи мультфильмов, придают им реалистичность.

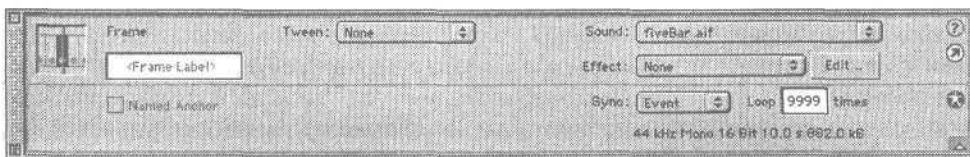
Из предыдущих глав вы знаете, что Flash можно использовать не только в качестве анимационной программы, но и для создания интерактивных системных меню, игр и Web-узлов. Является ли все это анимацией? В принципе нет, однако все объекты, находящиеся в среде Flash, можно считать анимационными. Временная шкала Flash применима к каждому объекту, включая кнопки, клипы, сцены и даже графику, поскольку любой создаваемый нами объект *сразу же* попадает во временную шкалу. Вне зависимости от того, какое приложение вы разрабатываете во Flash, вы должны исходить из этого постулата.

Как и мультфильмы, анимация Flash значительно выигрывает, если она дополнительно оформлена звуковыми эффектами. Для того чтобы в своих проектах вы могли использовать звук, необходимо научиться его синхронизировать. С этой целью в данной главе представлены следующие темы:

- ▶ типы синхронизации в среде Flash: Event, Stream, Start и Stop;
- ▶ синхронизация речи с анимацией;
- ▶ привязка звуков к событиям кнопок;
- ▶ синхронизация звука с клипами.

## Типы синхронизации в среде Flash

При привязке звука к кадру на временной шкале необходимо установить тип его синхронизации с анимацией на панели свойств (рис. 21.1). Тип синхронизации — Event, Stream, Start или Stop — определяет связь между звуком (музыкой, звуковыми эффектами, диалогами и т. д.) и анимированными компонентами проекта.



**Рис. 21.1.** Панель свойств содержит список Sync, в котором устанавливается тип синхронизации

Для синхронизации аудио- и видеоэффектов можно связывать воспроизведение звуков с частотой кадров фильма или начинать и прерывать воспроизведение



звуков независимо от анимации. (Для прерывания воспроизведения звука в нужное время существует тип синхронизации Stop.) Знание этих приемов очень важно для получения максимального эффекта при использовании звуковых файлов в проекте.

## Синхронизация типа Event

Синхронизация типа Event очень часто используется в среде Flash. Ее имя говорит само за себя — она связывает звуки и события. Как только Flash отображает ключевой кадр, который содержит Event-синхронизированный звук, этот звук полностью воспроизводится независимо от временной шкалы. Даже после последнего кадра звук с синхронизацией типа Event будет воспроизводиться до достижения конца файла. Поэтому, если вы хотите использовать короткий звук в качестве сигнала, удобно выбрать для него тип синхронизации Event.

Звуки типа Event очень легко применять в фильме, потому что все что вам нужно сделать — запустить воспроизведение звука с определенного момента. Так как эти звуки являются независимыми от анимации, то нет необходимости беспокоиться о количестве кадров, нужных для воспроизведения звука. Звуки с синхронизацией типа Event применяют в следующих случаях.

**Кнопки.** Звуки обычно привязывают к событиям кнопок Over и Down. Когда на кнопку наводится указатель мыши, воспроизводится один звук, а когда на кнопке щелкают мышью — другой. Поскольку изменения состояния кнопок (Over и Down) происходят быстро, очень важно использовать тип синхронизации, который обеспечивает соответствующую реакцию на эти изменения. Если несколько раз щелкнуть на кнопке мышью или многократно провести над ней указателем, звук, для которого выбран тип синхронизации Event, будет воспроизводиться каждый раз, когда выполняются эти действия. В результате звуки будут смешиваться. Такая особенность синхронизации типа Event позволяет создавать интересные музыкальные эффекты. (Более подробная информация об этом содержится в разделе «Звуковые эффекты типа Event и музыка».) Если же появление таких эффектов нежелательно, следует использовать тип синхронизации Start, который рассмотрен в конце этого раздела.

**Звуковые эффекты.** Если в проекте предусмотрены короткие звуковые эффекты, следует использовать только тип синхронизации Event. Это позволит полностью синхронизировать звуки с анимационными событиями, например связать характерный звук с баскетбольным мячом, проходящим через кольцо. Создайте в одном из звуковых слоев ключевой кадр и добавьте в него звук с синхронизацией Event. Позиции этого ключевого кадра и соответствующего кадра анимации должны совпадать. При просмотре фильма вы увидите, как мяч проходит через кольцо, и услышите характерный звук. Данный прием рассматривается более подробно в разделе «Звуковые эффекты типа Event и музыка».

**Внезапные звуки.** В фильмах внезапные звуки сопровождают драматические события. Например, при появлении вампира у кровати спящей женщины раздается дрожащий звук скрипки, от которого волосы на голове становятся дыбом. Такой прием использовался многие годы при создании драматических сцен в кино и

мультипликации. При необходимости вы тоже можете применить опцию синхронизации Event для усиления драматического эффекта в фильме.

## Синхронизация типа Stream

Синхронизация типа Stream отличается от синхронизации всех остальных типов, доступных в среде Flash. Она связывает скорость воспроизведения анимации с темпом звука, при этом в случае необходимости кадры анимации могут пропускаться или добавляться. Эта функция превращает Flash в видеоредактор, который позволяет синхронизировать видео- и аудиоинформацию. В отличие от звуков типа Event, Stream-звуки воспроизводятся только тогда, когда хватает кадров для обеспечения их продолжительности. Иногда для полного воспроизведения звука следует добавить кадры.

На рис. 21.2 у верхнего звука (Layer 2) не имеется достаточно кадров для полного воспроизведения, поэтому звуковая волна обрывается на последнем кадре. У нижнего звука (Layer 1) достаточно кадров (на рисунке видно, что звуковая волна заканчивается горизонтальной линией), и поэтому он воспроизводится полностью.

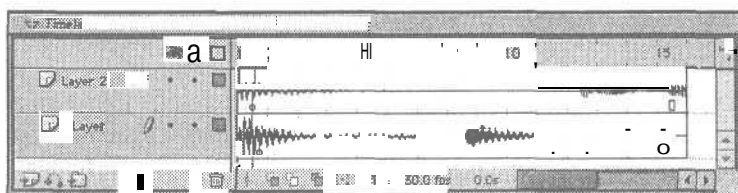


Рис. 21.2. Пример, иллюстрирующий синхронизацию типа Stream

Учитывая эту особенность, мы рекомендуем при использовании данной опции по возможности избегать циклического воспроизведения звука. Иначе может получиться, что Flash добавит ненужную информацию в проект и увеличит размер SWF-файла. Рассмотрим случаи, когда следует применять тип синхронизации Stream.

**Музыкальное сопровождение.** Если в проекте предусматривается частое использование музыки, лучше всего применять к звуку Stream-синхронизацию. Такой звук будет задавать анимации необходимый темп, обеспечивая синхронизацию изображения с музыкой.

**Синхронизация речи.** Для полной синхронизации разговора или слов песни с анимацией необходимо использовать Stream-синхронизацию. Тогда Flash будет поддерживать темп движения губ вашего персонажа в соответствии с темпом диалога или музыки.

**Многотрековый и многослойный звук.** Каждый из типов синхронизации позволяет воспроизводить до восьми разных звуков одновременно. Если поместить каждый звук в отдельный слой, можно создать в фильме многотрековый звук. Использование опции Stream позволит сохранить взаимную синхронизацию треков (см. практикум 6).

## Синхронизация типа Start

Этот тип синхронизации обладает определенным сходством с синхронизацией типа Event. Как только воспроизводится кадр, куда помещен звук с типом синхронизации Start, начинается воспроизведение звука, которое продолжается до полного завершения звука. Единственное отличие состоит в том, что если звук уже воспроизводится, другого экземпляра этого звука слышно не будет. Вы можете использовать Start-синхронизацию в качестве фильтра, предотвращающего дублирование звука.

Например, для озвучивания событий кнопок лучше применять Start-синхронизацию. Так, если вы привяжете звук к событию кнопки Over и используете Event-синхронизацию, звук будет слышен при каждом наведении курсора на кнопку. При частом повторении такого действия воспроизводится большое количество звуков, которые станут накладываться друг на друга. Чтобы предотвратить это, в данном случае вместо Event-синхронизации следует применить Start-синхронизацию.



Примеры использования Start- и Event-синхронизации содержатся в файле `buttonSync.swf`. Данный файл расположен в каталоге Chapter 21 на компакт-диске нашей книги. Откройте этот файл и несколько раз быстро переместите указатель над кнопкой Event. Вы услышите, что звук повторяется при каждом перемещении указателя мыши над кнопкой. Теперь точно также несколько раз проведите курсором над кнопкой Start. В этом случае звук будет воспроизводиться только один раз.

## Синхронизация типа Stop

Последний тип синхронизации — Stop, как вы уже, наверное, догадались, нужен для остановки звука. Такая синхронизация используется с целью прерывания звуков, для которых был задан тип синхронизации Event или Start. Чтобы задать остановку звука, выполните следующие действия.

1. Вставьте новый ключевой кадр, на котором вы хотите прервать звук. Ключевой кадр при этом должен соответствовать определенному событию анимации.
2. Активизируйте команду Window ▸ Properties для отображения панели свойств. В списке Sound (Звук) выберите имя звука, который нужно остановить, а в списке Sync — элемент Stop. Когда головка воспроизведения будет проходить этот ключевой кадр, звук, выбранный вами, прервется.

Этим методом тип синхронизации Stop можно применять к звукам с синхронизацией типа Event и Start. Остановку воспроизведения звуков с синхронизацией типа Stream следует задавать по-другому. Необходимо согласовать последний кадр звука с анимационным событием. Остановка звука с синхронизацией типа Stream осуществляется таким образом.

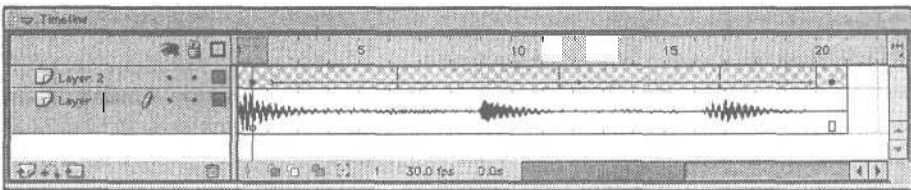
1. Определите анимационное событие, которое будет синхронизировано с последним кадром звука. Номер кадра, где происходит это событие, следует запомнить.

2. Подсчитайте количество лишних кадров, занимаемых звуком. Для этого из номера последнего кадра потокового звука вычтите номер кадра анимационного события, на котором нужно остановить звук. Например, если звук воспроизводится до кадра 46, а вы хотите, чтобы он останавливался на кадре 20, значит, на временной шкале 26 лишних кадров звука, которые необходимо удалить.
3. Выделите первый удаляемый кадр. Затем, последовательно для каждого удаляемого кадра нажмите комбинацию клавиш **Shift+F5**. В приведенном выше примере, вам придется нажимать клавиши 26 раз. Изображение звука с синхронизацией типа **Stream**, который остановится на 20 кадре, показано на рис. 21.3.

#### ПРИМЕЧАНИЕ

Можно удалять кадры другим способом — выбором диапазона удаляемых кадров на временной шкале. Для этого выделите первый кадр требуемого диапазона и задайте длину диапазона, выбрав последний удаляемый кадр и удерживая нажатой клавишу **Shift**. Нажмите **Shift+F5**, и кадры выбранного диапазона будут удалены из временной шкалы.

4. После удаления ненужных кадров активизируйте команду **Control ▸ Test Movie** для проверки синхронизации. Чтобы сделать ее более точной, можно добавлять и удалять кадры в звуковом слое.



**Рис. 21.3.** Звук с синхронизацией типа **Stream** начинается в кадре 1 и заканчивается в кадре 20

## Приемы синхронизации

Теперь, когда вы получили начальные сведения о приемах синхронизации анимации и звука, давайте углубимся в эту тему и посмотрим, как данные приемы применяются в различных ситуациях. Конечно, представленные нами примеры не являются готовыми рецептами на все случаи жизни, но наверняка смогут стать отправной точкой для ваших разработок в среде **Flash**.

### Создание анимации с синхронизацией речи

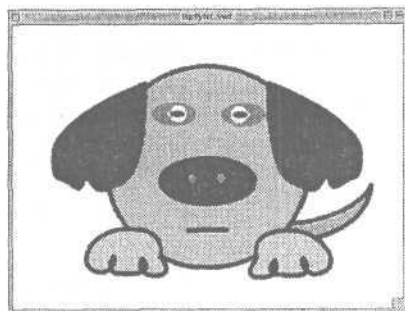
Существует множество способов создания анимации с синхронизацией речи в среде **Flash**. В зависимости от цели, которую вы преследуете, можно синхронизировать звук и изображение с максимальным приближением к реальности либо, напротив, стилизовать воспроизведение звука. Удачная синхронизация речи может «оживить» героев вашего фильма.

Если нужно согласовать анимацию с аудиотреком, используйте тип синхронизации *Stream*. Другой способ синхронизации звука и графики — подгонка изображения к аудиотреку. Передвигая воспроизводящую головку по временной шкале с помощью мыши, можно определить, в какой части фильма не выполнена синхронизация.

Для создания анимации с синхронизацией речи нужны нарисованные персонажи и записанный диалог, импортированный в фильм. Особенность такой анимации состоит в том, что изображение рта персонажа должно находиться на отдельном слое. Естественно, при синхронизации речи используются несколько изображений рта, которые представляют разные позиции губ при разговоре персонажа.



В примере, который мы сейчас рассмотрим, все графические элементы и звук созданы. На прилагаемом компакт-диске откройте файл *lipSync.fla*, находящийся в каталоге *Chapter 21*, и сохраните его на жестком диске. Результат, который должен у нас получиться, можно увидеть, если открыть файл *lipSync.swf*.



Чтобы синхронизировать анимацию с речью, выполните следующие действия.

1. Откройте файл *lipSync.fla* и создайте во временной шкале два новых слоя с именами *dialog* и *mouth*.
2. Активизируйте команду *Insert* ▶ *Keyframe* и вставьте во второй кадр слоя *dialog* новый ключевой кадр. Вызовите команду *Properties* в меню *Window*, а затем в списке *Sound* выберите имя звукового файла *woofwoofbark.aif*. В списке *Sync* установите тип синхронизации *Stream*. В данном случае целесообразно использовать именно такую синхронизацию, поскольку она подчиняет анимацию звуку.
3. Теперь можно создать анимацию для фразы «ваф, ваф, барк». Обратите внимание, что график звука в одних местах имеет форму волны, а в других — прямой линии (рис. 21.4). Наличию звука соответствуют волны на графике. Если с помощью мыши перетащить воспроизводящую головку вдоль временной шкалы, вы услышите медленное воспроизведение звука. Это может дать представление о «словах», звучащих в каждом кадре.

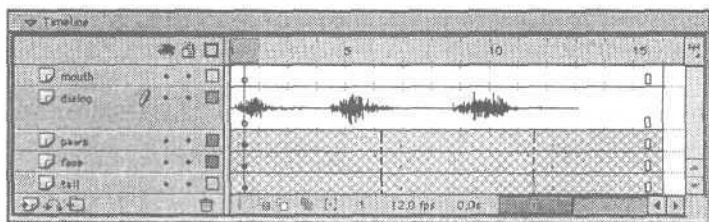


Рис. 21.4. Звук есть там, где график имеет форму волны

4. Выберите кадр 1 в слое `mouth` и вставьте новый ключевой кадр. Перетащите графический символ `closed` из библиотеки в то место области действия, где должно быть изображение рта. Это определит закрытую позицию рта.
5. Создайте ключевой кадр в кадре 2 временной шкалы слоя `mouth`. Перетащите графический символ `W` в область действия и поместите его там, где должно быть изображение рта (приблизительные координаты —  $X = 275$ ,  $Y = 300$ ). Для выравнивания изображений можете использовать панели `Info` или `Align`. Графический символ `W` задаст позицию рта при произнесении звука «в». Таким образом мы начали создать анимацию, отвечающую первому слову «ваф».
6. Создайте еще один ключевой кадр в кадре 3 временной шкалы слоя `mouth` и перетащите графический символ `F` в область действия. Добавьте еще один ключевой кадр и перетащите в область действия графический символ `closed`. На этом этапе создание анимации для первого слова «ваф» завершено.
7. Продолжайте добавлять новые ключевые кадры и перетаскивать изображения рта в область действия, пока не закончите создавать анимацию для всей фразы. Для проверки можете обратиться к табл. 21.1, в которой описано размещение символов в кадрах.
8. Активизируйте команду `Control ▶ Test Movie`, чтобы просмотреть и прослушать результаты своей работы. Созданный вами диалог является очень коротким, поэтому, прослушивая его, трудно в полной мере оценить качество синхронизации. Однако вы можете выбрать опцию `Stream` для длинных диалогов и убедиться, что анимация синхронизирована с речью.

**Таблица 21.1.** Кадры и графические символы для фильма `lipSync.fla`

| Кадр | Графический символ (позиция рта) |
|------|----------------------------------|
| 1    | <code>Closed</code>              |
| 2    | <code>W</code>                   |
| 3    | <code>F</code>                   |
| 4    | <code>Closed</code>              |
| 5    | <code>W</code>                   |
| 6    | <code>F</code>                   |
| 7, 8 | <code>Closed</code>              |
| 9    | <code>B</code>                   |
| 10   | <code>A</code>                   |
| 11   | <code>⌘</code>                   |
| 12   | <code>Closed</code>              |

Как видно из приведенного примера, синхронизация речи предоставляет достаточно большие возможности. Для того чтобы применять их более профессионально, следует придерживаться дополнительных правил.

- ▶ Пользуйтесь зеркалом, чтобы по аналогии с положением своих губ при произношении нужных слов создать такую же позицию губ в анимации.
- ▶ Рассматривайте всю фразу целиком, так как движения губ для отдельных букв и для тех же букв, но объединенных в слова, не одинаковы.

- ▶ Используйте действие `gotoAndPlay()`, для того чтобы временно изолировать и циклически повторять части анимации, относящиеся к речи.
- ▶ Правильное изображение формы рта играет существенную роль в анимации речи. Поэтому рекомендуем обратиться к соответствующей литературе об искусстве анимации.

## Звуковые эффекты типа Event и музыка

Звуки типа Event можно использовать в проектах для создания разнообразных эффектов. Но при этом не следует забывать, что синхронизация таких звуков осуществляется только в момент их запуска. Поэтому в одних случаях не рекомендуется применять этот тип синхронизации, а в других, напротив, его применение весьма эффективно.

Звук типа Event всегда связывается с соответствующим событием в проекте. Поэтому данный тип синхронизации и назвали Event (событие). Событие служит сигналом для начала воспроизведения звука. Затем звук воспроизводится независимо от временной шкалы до тех пор, пока не закончится сам или пока его не **остановят**. Например, если при падении какого-либо предмета на пол раздается звон, он может долго звучать после этого независимо от дальнейшего развития событий фильма.

Так как звук типа Event после запуска воспроизводится до тех пор, пока не будет достигнут конец звукового файла, вы можете создать в фильме большое количество интересных многоуровневых эффектов, используя наложение нескольких звуков. Например, если ваш герой роняет на пол несколько тарелок, будет слышен не один звук разбитой посуды, а сочетание различных звуков в зависимости от количества упавших тарелок. Основываясь на анимации сцены, можно связать каждый звуковой эффект с соответствующим событием, которым является падение определенной тарелки. Для этого надо создать отдельный слой для каждого звука, а затем вставить новый ключевой кадр там, где происходит озвучиваемое событие, и добавить в этот кадр звук с типом синхронизации Event. Во время проигрывания серии анимационных событий каждый звук начинает воспроизводиться в нужное время. Поскольку каждый звук воспроизводится отдельно, он будет звучать до тех пор, пока не закончится, создавая прекрасный многоуровневый эффект. Преимущество этого приема очевидно. Он достаточно прост, но в большинстве случаев позволяет получить более качественный результат, чем при создании сложных звуков «вручную» с отладкой каждого элемента.

При работе со звуками типа Event следует придерживаться определенных правил:


- ▶ Эти звуки слышны при их первой вставке во временную шкалу. Для предотвращения задержек удостоверьтесь в том, что звуковой файл очищен от «мертвого пространства» и тишины в начале файла. Очистка звуковых файлов рассмотрена в главе 19.
- ▶ Если вы создаете слой с несколькими громкими звуками, следует установить зависимость их уровня громкости от времени так, чтобы звуки не искажались при взаимном наложении. Работа с уровнем громкости описана в главе 20.

- ▶ Flash может воспроизводить не более восьми звуков одновременно. Посмотрите на временную шкалу и, если вы видите участок, в котором активны восемь слоев одновременно, отрегулируйте шкалу для предотвращения слишком большого количества совпадений.

Данный тип синхронизации хорош также и для музыки. *Поэкспериментируйте*, чтобы иметь представление о том, какие музыкальные эффекты можно создать с его помощью.

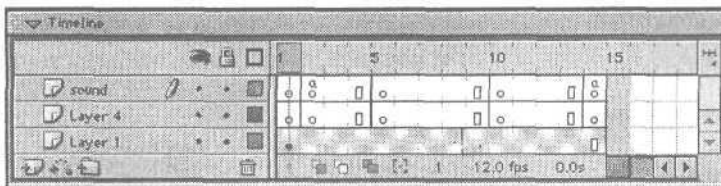
## Синхронизация звука с клипами

Клип — это один из наиболее важных элементов фильма в среде Flash. Анимация в клипе может осуществляться независимо от главной временной шкалы, поэтому он является хорошим средством для синхронизации звуков с изображением в фильме. Если создать клипы, которые содержат звуковые файлы, привязанные к различным кадрам, то воспроизведением этих файлов можно управлять с помощью кадров клипа. Наряду с главной временной шкалой это дает вам дополнительную возможность управления звуком.

 В рассматриваемом ниже примере показано, как использовать клипы для управления воспроизведением звука. Внимательно просмотрите файлы `audioMC fla` и `audioMC.swf`, которые находятся в каталоге Chapter 21 на прилагаемом компакт-диске. Использование клипов для синхронизации звуков выполняется следующим образом. Сначала создается клип, а затем в его кадры добавляется звук и организуется управление проигрыванием кадров клипа.

Для создания клипа, в котором синхронизируется звук, выполните следующие действия:

1. Если вы еще не импортировали звук в фильм, сделайте это. Затем выберите команду `Insert` ▶ `New Symbol` и в открывшемся окне в поле `Name` введите имя клипа `audio`, а в группе `Behavior` установите переключатель `Movie Clip`.
2. Щелкните на кадре 2 временной шкалы клипа `audio` и вставьте новый ключевой кадр. Затем дважды нажмите клавишу `F5` для вставки двух дополнительных кадров.
3. Создайте новый ключевой кадр в позиции 5 временной шкалы и четыре новых кадра после него. Прделайте тоже самое в позиции 10 (последний добавленный кадр будет четырнадцатым). И, наконец, добавьте последний ключевой кадр в позицию 14. Созданная структура показана на рис. 21.5.



**Рис. 21.5.** Структура кадров для звукового клипа имеет ключевые кадры во 2, 5, 10 и 14 кадрах



4. Теперь, когда основная структура клипа создана, можно приступить к заполнению его компонентов. Привяжите следующие команды к ключевому кадру 2:

```
stop();
stopAllSounds();
```

Эта последовательность команд задает остановку воспроизведения клипа и, соответственно, воспроизведение звуков в кадре 2.

5. Привяжите звук к кадру 5 и назначьте ему тин синхронизации Start. Введите количество повторений и, если нужно, добавьте эффекты. При использовании этой синхронизации звук, который в это время проигрывается, не прервется таким же звуком. Не произойдет и их взаимное наложение.

6. Привяжите к ключевому кадру номер 14 следующую команду:

```
gotoAndPlay(10);
```

Эта строка — наиболее важная, потому что в ней организовано циклическое проигрывание. Воспроизведение звука в клипе начнется в кадре 5 и продлится до кадра 14. Здесь находится команда, фактически задающая цикл. Воспроизводящая головка клипа переместится к кадру 10. Так как в этом кадре нет никаких команд, клип опять будет воспроизводиться до кадра 14, откуда воспроизводящая головка снова переместится в кадр 10 и т. д. Таким образом, звук будет воспроизводиться до тех пор, пока не будет получена другая команда. Так как выбран тип синхронизации Start, это происходит независимо от любой временной шкалы.

Теперь, когда клип создан, можно организовать управление им в фильме. Далее продемонстрированы два приема управления — в клипе и с помощью главной временной шкалы.

Для создания в клипе элементов управления воспроизведением звука выполните следующие действия:

1. Выберите клип audio из меню кнопки Edit Symbols или два раза щелкните на его имени в библиотеке.
2. Вставьте в клип новый слой и перетащите на его область действия кнопку, которая будет служить кнопкой запуска. Свяжите в ней следующие операторы ActionScript:

```
on(release){
 this.gotoAndPlay(5);
}
```

Эти команды обеспечат переход клипа в пятый кадр, чтобы начать воспроизведение звука. Перетяните еще одну кнопку в область действия (она предназначена для остановки звука) и свяжите с ней такие операторы:

```
on(release){
 this.gotoAndPlay(1);
}
```

#### ПРИМЕЧАНИЕ

В данном случае можно обойтись и без префикса this, действия gotoAndPlay(5) и gotoAndPlay(1) будут работать и без него. Этот префикс используют для того, чтобы указать полный путь к текущей временной шкале.

Действие `gotoAndPlay(1)` обеспечит переход клипа в кадр 1. Проигрывание клипа начнется с кадра 1 дойдет до кадра 2 и будет остановлено действиями `stop()` и `stopAllSounds()`. Эти действия остановят воспроизведение всех звуков в фильме.

Давайте разберемся, что же происходит в клипе. Когда он загружается в фильм впервые, то останавливается на кадре 2. При этом не воспроизводится звук (вспомните, что звук начинается в кадре 5). Когда клип получает указание перейти к кадру 5, звук начинается и клип циклически воспроизводит кадры с 10 по 14. Когда вы хотите остановить звук, кнопка остановки обеспечивает переход клипа к кадру 1. Он начинает проигрываться, но по достижении кадра 2 связанные с этим кадром действия `stop()` и `stopAllSounds()` прерывают проигрывание клипа.

- Итак, элементы управления звуком клипа нами созданы. Выйдите из режима редактирования символов и возвратитесь в главную временную шкалу. Перетащите экземпляр клипа `audio` из библиотеки в область действия главной временной шкалы. Активизируйте команду `Control ▶ Test Movie` и поэкспериментируйте с остановкой и воспроизведением звука.

Элементы управления остановкой и запуском воспроизведения звука, размещенные в клипе, — это хорошо, однако неплохо бы научиться управлять клипом из главной временной шкалы. Для этого нужно организовать доступ к клипу так, чтобы вы могли посылать ему команды, управляющие проигрыванием нужных кадров.

- На панели свойств присвойте экземпляру клипа `audio` имя `soundMC`. Затем перетащите две кнопки в область действия главной временной шкалы (если кнопок нет в библиотеке, то создайте их). Одна кнопка будет служить для запуска, а другая — для остановки воспроизведения звука.
- Свяжите с кнопкой запуска воспроизведения звука следующие команды:

```
on(release){
 soundMC.gotoAndPlay(5);
}
```

- С кнопкой остановки воспроизведения звука свяжите такие команды:

```
on(release){
 soundMC.gotoAndPlay(1);
}
```

Префикс `soundMC` в этих командах задает путь к клипу.

- Активизируйте команду `Control ▶ Test Movie`. Теперь вы можете при помощи кнопок включать и выключать звук, как в клипе, так и на главной временной шкале.

Если управление клипом осуществляется из главной временной шкалы, кнопки внутри клипа иметь необязательно. Поэтому вы можете убрать кнопки из клипа и оставить в нем только звук. Для работы клипа не требуется графическая информация, кадры клипа будут содержать только команды, звуки и ключевые кадры.

## Разработки

В Интернете существует много прекрасных фильмов и страниц, на которых использована синхронизация звука средствами Flash. Мы предлагаем вашему вниманию исключительно оригинальный и уникальный проект — Интернет-страницу, находящуюся по адресу [www.eugenemirman.com](http://www.eugenemirman.com) (рис. 21.6). Она создана Юджином Майермэном, а синхронизацию голоса осуществил Скотт Баверс.



Рис. 21.6. Web-узел eugenemirman.com

## Резюме

Различные звуки (музыка, диалог, звуковые эффекты и т. п.) делают фильмы, созданные в среде Flash, более привлекательными. При этом очень важно, чтобы звук и графика являлись взаимно согласованными элементами фильма, для чего и служит синхронизация. Flash позволяет установить синхронизацию изображения с диалогом, музыкой и другим звуковым оформлением. Короткие звуки можно воспроизводить при наступлении заданного события. Вы можете озвучивать события в фильме, подтверждать звуком воздействие пользователя на элементы управления, легко создавая эффекты взаимного наложения звуков. В этой главе рассматривались различные типы синхронизации, поддерживаемые Flash, и с помощью примеров были показаны наилучшие варианты их применения.

Средства синхронизации позволяют манипулировать звуком и анимацией в фильме. В следующей главе рассказывается о языке ActionScript, с помощью которого вы можете полностью управлять звуковым оформлением фильма.

## Глава 22

# Программное управление звуком

Из предыдущих глав вы знаете, что управление звуком в среде Flash осуществляется с помощью привязки звуков к временной шкале фильма. Этот прием достаточно эффективен, так как позволяет синхронизировать музыку, звуковые эффекты и диалоги с анимацией. Единственным его недостатком является ограничение воспроизведения звука кадрами клипа или главной временной шкалой. Поэтому, если временная шкала, содержащая нужный вам звук, не загружена, придется загрузить ее вручную. Но если использовать язык `ActionScript`, поддерживаемый Flash, положение улучшается.

В пятой версии Flash язык `ActionScript` был дополнен новым объектом — `Sound`, который обеспечивает полный контроль над каждым звуком в фильме независимо от временной шкалы. Во Flash MX реализованы новые функции, и теперь вы можете загружать внешние звуки, управлять воспроизведением звуков и отслеживать звуковые события. Объект `Sound` позволяет динамически изменять каждый параметр звука. В этой главе рассмотрены следующие темы:

- ▶ создание объектов `Sound`;
- ▶ воспроизведение и остановка звуков при помощи команд языка;
- ▶ установка и динамическое изменение уровня громкости;
- ▶ установка и динамическое изменение позиции панорамирования;
- ▶ применение стереоэффектов к звукам;
- ▶ использование интерактивных элементов управления для манипулирования звуками в вашем фильме;
- ▶ загрузка внешних звуков формата MP3 в фильм.

## Работа с объектами `Sound`

Объект `Sound` — это один из predeterminedных объектов `ActionScript`. С его помощью вы можете управлять параметрами воспроизведения звуков в фильме. Он позволяет воспроизводить и прерывать звуки, устанавливать уровень громкости, а также изменять громкость в левом и правом каналах. Кроме того, объект `Sound` предоставляет возможность динамически управлять громкостью, панорамированием и эффектами, присущими каждому звуку. Для более эффективного использования этого объекта важно иметь полное представление об его функциональных возможностях и понимать, как он действует в среде `ActionScript`.

## Понятие объекта `Sound`

Как и все объекты `ActionScript`, объект `Sound` содержит информацию об определенных компонентах фильма. Например, объект `MovieClip` хранит данные о свойствах конкретного клипа, а объект `Math` -- об арифметических вычислениях.

В свою очередь, объект `Sound` содержит информацию о звуках фильма и их свойствах. Чтобы управлять звуком с помощью `ActionScript`, нужно создать новый объект `Sound`.

Для обмена информацией между объектом и фильмом в `ActionScript` применяются *методы*. Методы объекта `Sound` позволяют проверять и устанавливать уровень громкости и задавать панорамирование, включать и выключать воспроизведение отдельных звуков. Таким образом, взаимодействие с объектом осуществляется с помощью его методов.

## Использование объекта `Sound`

Чтобы получить возможность управлять каким-либо звуком, вы должны сначала создать новый экземпляр объекта `Sound`. После этого с его помощью можно контролировать:

- ▶ звуки, доступные во `Flash Player` (все файлы `SWF`, загруженные в разных слоях);
- ▶ звуки, имеющиеся в фильме;
- ▶ звуки, определенные в клипе;
- ▶ звуки, которые связаны с этим объектом.

Возможности объекта `Sound` зависят от того, с какими параметрами он создан. Базовый конструктор для создания объекта `Sound` имеет следующий вид:

```
soundObjectName - new Sound("targetInstance");
```

где `soundObjectName` — это имя, которое присваивается объекту, а `targetInstance` — необязательный параметр. Если вы хотите управлять звуком в клипе, необходимо в качестве аргумента указать имя этого клипа.

При создании объекта, который предназначен для управления звуком во всем фильме, аргумент указывать не надо. В этом случае следует использовать такой конструктор:

```
globalMovieSound - new Sound();
```

где `globalMovieSound` — это имя нового объекта, позволяющего управлять звуками во всех временных шкалах.

Для создания объекта, управляющего звуком в клипе или конкретном слое, используйте следующие конструкторы:

```
monkeyClipSound = new Sound("bananas");
loadedSound - new Sound("_level1");
```

В первом случае создается объект с именем `monkeyClipSound`, позволяющий управлять свойствами звуков в клипе `bananas`. Во втором — объект с именем `loadedSound`, предназначенный для управления звуками временной шкалы фильма, которые загружены в слой `_level1`.

Когда объект `Sound` указывает на звук в другой временной шкале, его возможности ограничены, он позволяет управлять уровнем громкости и панорамированием. Воспроизведение и остановка работать не будут, потому что их можно задать только в кадрах временной шкалы, где находится звук. Тем не менее, с помощью объекта `Sound` можно контролировать отдельные звуки, находящиеся в библиотеке.

Для этого необходимо привязать звук непосредственно к объекту. (Более подробная информация содержится в разделе «Привязка звуков с помощью связывающих идентификаторов».)

## Управление воспроизведением звука

Динамическое управление воспроизведением звука осуществляется с помощью методов объекта Sound только после его создания.

Управление звуком с помощью ActionScript имеет некоторые преимущества по сравнению с обычным позиционированием в кадрах. Во-первых, если вы хотите воспроизвести какой-либо звук, нет необходимости использовать клип или временную шкалу для проигрывания соответствующего кадра. Объект Sound позволяет непосредственно обращаться к звуку, и вы можете воспроизвести его без проигрывания кадра, в котором он находится. Стандартные команды ActionScript позволяют останавливать все звуки в фильме. С помощью объекта Sound, кроме этого, можно запустить воспроизведение всех звуков в фильме.

Во-вторых, язык ActionScript обеспечивает гибкость управления звуком. Любой компонент фильма, с которым можно применять ActionScript, способен управлять воспроизведением звуков. Это позволяет создавать сценарии для звукового сопровождения с учетом динамического перемещения графических элементов фильма.

Далее представлены методы объекта Sound, которые применяются для управления воспроизведением звука.

- ▶ `start (secondsOffset, loops)`. Используется для воспроизведения звуков. Метод `start` может принимать два необязательных аргумента. Аргумент `secondsOffset` применяется для воспроизведения звука с указанной позиции звукового файла. Например, для воспроизведения десятисекундного звука с середины следует указать смещение в 5 секунд. Аргумент `loops` задает количество повторений звука. Если этот аргумент не определен, звук воспроизводится один раз. Если же данный аргумент равен 2, звук повторится дважды, если 3 — трижды и т. д.

---

### ПРИМЕЧАНИЕ

Чтобы воспроизвести звук с помощью метода `start()`, следует сначала экспортировать его из библиотеки, используя команду контекстного меню `Linkage`. Более подробно об этом можно узнать в разделе «Привязка звуков с помощью связывающих идентификаторов».

---

- ▶ `stop (soundID)`. Применяется для остановки звуков. При отсутствии аргумента этот метод остановит все звуки, контролируемые объектом. Остановка конкретных звуков, привязанных к объекту Sound, осуществляется с помощью необязательного аргумента `soundID`.

Примеры использования этих методов приводятся далее в разделе «Воспроизведение и остановка звуков».

## Динамическая установка уровня громкости и параметров панорамирования

Вы уже знаете, что манипулировать воспроизведением звуков можно путем размещения их в кадрах главной временной шкалы или клипа. Таким же способом можно устанавливать уровень громкости и панорамирования. Для изменения указанных параметров звука, размещенного в ключевом кадре, применяется окно Sound Envelope (рис. 22.1). Оно обеспечивает наглядность представления выполненных изменений параметров звука. Данные изменения являются постоянными и не подлежат отмене после публикации фильма. Для изменения параметров звука нужно возвратиться к ключевому кадру, содержащему звук, открыть панель Properties и ввести в ней новые значения.

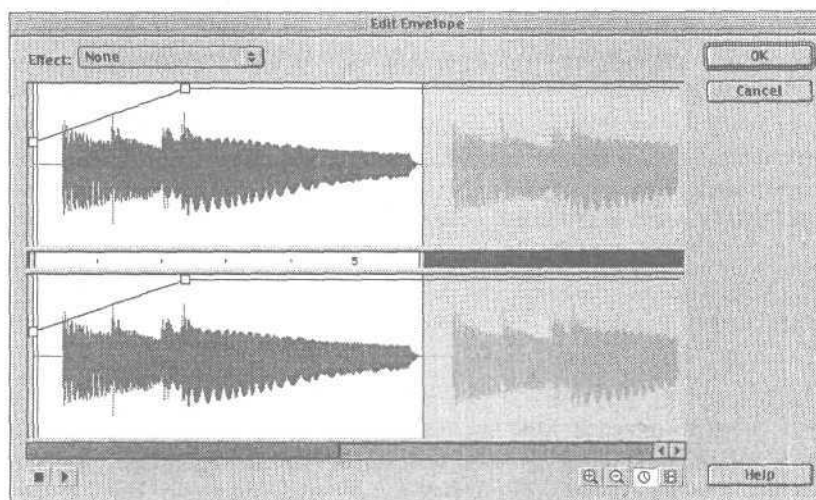


Рис. 22.1. Окно Edit Envelope

В отличие от этого способа, `ActionScript` предоставляет возможность с помощью ряда методов задавать уровень громкости и параметры панорамирования, а также контролировать данные свойства звука в любое время и из любой позиции в фильме. Эти методы представлены ниже.

- ▶ `setVolume(volume)`. Задаёт уровень громкости звука. Аргумент `volume` может принимать значения от 0 (минимальный уровень громкости) до 100 (максимальный уровень), по умолчанию он равен 100.
- ▶ `getVolume()`. Возвращает текущий уровень громкости звука (значение в интервале от 0 до 100).
- ▶ `setPan(pan)`. Устанавливает параметр панорамирования для объекта `Sound`. Аргумент `pan` определяет баланс звука. Если он равен 100, звук воспроизводится полностью в правом канале. Значение -100 указывает, что звук воспроизводится полностью в левом канале. При значении 0 (по умолчанию) звук равномерно распределяется по обоим каналам.

► `getPan()`. Возвращает текущее значение параметра панорамирования. Он представлен значением в диапазоне от -100 (весь звук в левом канале) до 100 (весь звук в правом канале). Значение 0 (по умолчанию) означает, что звук равномерно распределен между обоими каналами.

Примеры использования этих методов приведены в разделе «Задание уровней громкости и эффекта панорамирования» данной главы.

## Создание специальных стереоэффектов

Язык ActionScript предоставляет в ваше распоряжение специальный метод объекта `Sound`, позволяющий создавать в фильме уникальные эффекты со стереозвуками. Речь идет о методе `setTransform()`.

В главе 19 рассказывалось, что стереозвучание создается за счет того, что звуки в левом и правом каналах образуются путем микширования различных компонентов. Например, звучание одних инструментов воспроизводится на левом канале, а других — на правом. При прослушивании такой композиции на одном канале, вы будете слышать только часть инструментов.

С помощью метода `setTransform()` можно управлять звуком, выполнив некоторые дополнительные действия. Нужно создать объект, который будет применять к объекту `Sound` необходимые преобразования. Этот объект (назовем его `transformer`) позволит управлять каждым каналом стереозвука, передавая или принимая значения в процентах, характеризующие распределение звука между каналами. Для каждого канала стереозвука в объекте `transformer` имеются два свойства, которые применяются, чтобы изменять этот звук. Одно из них задает выходные сигналы левого канала, а второе — правого канала (табл. 22.1).

**Таблица 22.1.** Свойства объекта `transformer`, который используется в функции `setTransform()`

| Свойство        | Значение | Описание                                                                        |
|-----------------|----------|---------------------------------------------------------------------------------|
| <code>ll</code> | 0-100    | Уровень звука левого канала, воспроизводимого в левом динамике                  |
| <code>lr</code> | 0-100    | Процентное отношение звука правого канала для воспроизведения в левом динамике  |
| <code>rr</code> | 0-100    | Процентное отношение звука правого канала для воспроизведения в правом динамике |
| <code>rl</code> | 0-100    | Процентное отношение звука левого канала для воспроизведения в правом динамике  |

Непонятно? Рассмотрим два примера использования значений свойств. В первом примере указанные значения задают воспроизведение стереозвука: весь звук левого канала воспроизводится на левом динамике, а весь звук правого канала — на правом динамике:

```
ll = 100
lr = 0
rr = 100
rl = 0
```

Во втором примере иллюстрируется создание реверсивного стереоэффекта: весь звук левого канала воспроизводится на правом динамике, а весь звук правого канала — на левом динамике:

```
ll = 0
lr = 100
```



`rr = 0`  
`rl = 100`

Значит, изменяя соответствующим образом значения этих свойств, можно создавать различные стереоэффекты.

Так как значения данных свойств сохраняются в объекте `transformer`, их можно передать объекту `Sound` посредством метода `setTransform(transformer)`, который применяет свойства, хранящиеся в объекте `transformer`, к звуковому объекту.

Второй метод этого же объекта, `getTransform()`, возвращает значения свойств объекта `Sound`.

Примеры использования метода `setTransform()` и создания объекта типа `transformer` рассматриваются далее в разделе «Программирование специальных эффектов».

## Воспроизведение звуков с помощью объекта `Sound`

Объект `Sound` позволяет достаточно гибко управлять воспроизведением звуков. Использование этого объекта предоставляет вам больше возможностей, чем обычное размещение звука в ключевых кадрах. Теперь вы можете внедрять звуки и управлять их свойствами динамически. В этом разделе рассматриваются использование объекта `Sound` для экспорта и привязки звука из библиотеки и внедрение его в фильм.

### Привязка звуков с помощью связывающих имен

Одна из наиболее полезных функций объекта `Sound` позволяет воспроизводить звуки, не вставленные непосредственно в ключевой кадр временной шкалы. Но перед этим нужно связать звук с объектом, при помощи которого будет осуществляться воспроизведение звука и управление им.

Чтобы связать звук с объектом `Sound`, необходимо сначала экспортировать звук из библиотеки. Для этого произведите следующие действия.

1. Выделите в библиотеке имя экспортируемого звука.
2. В контекстном меню активизируйте команду `Linkage` (Связь). Откроется окно `Linkage Properties` (Параметры связи), показанное на рис. 22.2.
3. Установите флажок `Export for ActionScript` (Экспортировать для `ActionScript`) и введите уникальное имя в поле `Identifier` (Идентификатор). Это имя будет применяться для связывания звука с объектом `Sound`. Например, на рис. 22.2 используется имя `libSound`.
4. Закончив задание параметров, нажмите кнопку `OK`. Теперь вы можете связать звук с объектом `Sound`.

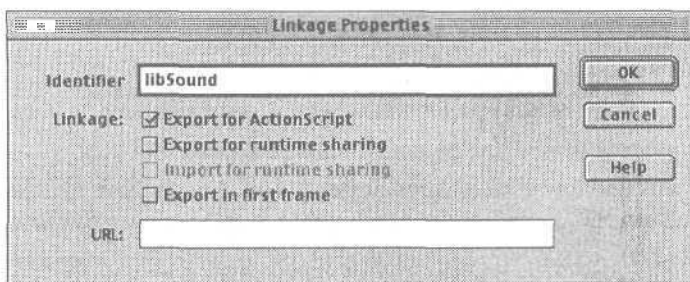


Рис. 22.2. Окно Linkage Properties

**ПРИМЕЧАНИЕ**

Окно Linkage Properties применяется также для включения звуков в общую библиотеку. Общие библиотеки позволяют использовать один и тот же файл в нескольких фильмах, создаваемых в среде Flash. Более детально работа с этими библиотеками рассматривалась в главе 7.

Объект можно одновременно связать только с одним звуком. Привязка нового звука приводит к разрыву предыдущей связи. Метод `attachSound()`, осуществляющий привязку звука к объекту, имеет следующий синтаксис:

```
sndObj.attachSound("idName")
```

Здесь `sndObj` — имя объекта, к которому будет привязан звук. Аргумент `idName` используется для объявления имени идентификатора, связанного с экспортированным звуком.

В следующем примере создается новый объект `Sound` — `movAudio`, с которым связывается звук, экспортированный под именем `libSound`.

```
movAudio = new Sound();
movAudio.attachSound("libSound");
```

После этого методы объекта `movAudio` будут управлять воспроизведением звука `libSound`.

**Воспроизведение и остановка звуков**

Для воспроизведения и остановки звуков используются два наиболее важных метода объекта `Sound`. Перед тем как изменять какие-либо параметры панорамирования или громкости в звуковом сопровождении фильма, следует получить возможность динамически подключать и отключать звуки. Для подключения звука выполните следующие действия.

1. Экпортируйте звук из библиотеки и свяжите его с идентификатором. Порядок действий, необходимых для этого, приведен в предыдущем разделе.
2. После экспортирования звука добавьте к кнопке, кадру или другим элементам фильма следующие команды:

```
mySound = new Sound0;
mySound.attachSound("bkgdLoop");
mySound.start(0, 999);
```

Эти команды выполняют такие действия:

- ▶ Создают новый объект `Sound` с именем `mySound`.
- ▶ Связывают с этим объектом звук, экспортированный как `bkgdLoop`.
- ▶ Начинают воспроизведение звука и повторяют его 999 раз.

Представим, что продолжительность звука равна 4 секундам. Чтобы задать, начиная с середины, 999 повторений звука, используйте следующий синтаксис:

```
mySound.start(2, 999);
```

В данной строке первый аргумент определяет запуск воспроизведения 4-секундного звука с его середины.

Другой вариант установки количества повторений предполагает применение переменной:

```
loop - 4
mySound.start(0, loop);
```

В этом примере переменной `loop` присваивается значение 4, что заставляет объект `mySound` повторить звук четыре раза. При этом вы можете использовать значение переменной `loop` в других частях фильма. Можно создать даже нечто вроде интерактивного контроллера (с применением элементов управления: ползунок, кнопки, комбинации клавиш и т. п.), который позволит изменять количество повторений.

Если звук воспроизводится с помощью объекта `Sound`, можно запрограммировать остановку звука. Рассмотрим следующие команды:

```
addASound - new Sound();
addASound.attachSound("ambient");
addASound.start();
addASound.stop();
```

Этот фрагмент программы выполняет такие действия:

- ▶ создает новый глобальный объект `Sound` с именем `addASound`;
- ▶ связывает звук `ambient` с объектом;
- ▶ воспроизводит звук;
- ▶ останавливает воспроизведение всех звуков в фильме (потому что это глобальный объект). Если объект `addASound` был привязан к определенному слою или клипу, то последняя команда остановит все звуки только в заданном объекте.

Также метод `stopO` может принимать в качестве аргумента идентификатор конкретного звука. Использование данного аргумента позволяет остановить заданный звук, указывая его идентификатор. Рассмотрим следующий пример:

```
addASound = new SoundO();
addASound.attach("ambient");
addASound.startO();
addASound.stop("ambient")
```

Здесь вы видите набор команд, аналогичный приведенному выше. Единственным отличием является аргумент `ambient` метода `stopO`. Хотя в рассматриваемом

случае объект `Sound` является глобальным, метод `stop()` остановит только звук с именем `ambient`. Все остальные звуки в фильме будут продолжать воспроизводиться.

## Программирование уровня громкости и панорамирования

Теперь вы знаете, что объект `Sound` позволяет управлять всеми параметрами звука; а `ActionScript` предоставляет возможность динамического задания уровня громкости и параметров панорамирования. Используя `ActionScript`, вы можете программировать изменения звуков фильма в зависимости от действий пользователя или событий в фильме.

В этом разделе рассматриваются ключевые концепции управления уровнями громкости и параметрами панорамирования с помощью `ActionScript`. Также здесь представлены приемы создания эффекта медленного затухания звука. Фрагменты программ, которые приводятся в данном разделе, позволят вам быстро усвоить излагаемый в книге материал и улучшить звуковое оформление фильма.

### Задание уровней громкости и эффекта панорамирования

Громкость и панорамирование являются наиболее важными и взаимосвязанными свойствами звука. Изменяя их параметры, можно получить интересный эффект при воспроизведении звуков в фильме. Позиционирование звука в стереополе помогает создать аудиосреду, где каждый элемент имеет свое место в общем звуковом оформлении фильма. Использование `ActionScript` позволяет задать это несколькими строками кода.

Изменение уровня громкости и параметров панорамирования осуществляется, когда звук уже воспроизводится. Поэтому задавать изменения можно с помощью методов объекта `Sound`.

Для установки уровня громкости используется метод `setVolume()`. Например, чтобы установить максимальный уровень громкости, нужно ввести строку

```
myObject.setVolume(100);
```

Здесь `myObject` — это имя объекта `Sound`. Данной строке должны предшествовать команды, которые создают объект `Sound` с именем `myObject`, связывают с ним звук из библиотеки и начинают его воспроизводить при помощи метода `start()`. Эти команды не обязательно должны быть расположены в том же окне сценария, что и команда `setVolume()`, но их выполнение является обязательным условием.

С помощью метода `setVolume()` можно управлять громкостью звуков, связанных с кадрами клипа или загруженными SWF-файлами. Рассмотрим следующий пример:

```
loaded - new Sound("_level1");
loaded.setVolume(50);
```

В этом примере создается новый объект `Sound` с именем `loaded` для звуков SWF-файла, воспроизводящихся на первом уровне. Метод `setVolume()` понижает уровень громкости звуков файла до 50 % от его первоначального значения. Использование такого же синтаксиса позволит контролировать уровень громкости звуков в клипе. Для этого вместо имени слоя следует указать полное имя клипа.

Управление панорамированием с помощью ActionScript аналогично управлению уровнем громкости. Единственным отличием является то, что параметр метода `setPan()` может принимать как отрицательные, так и положительные значения. При значении, равном `-100`, звук переходит полностью в левый канал, при `100` — полностью в правый. Рассмотрим следующий пример:

```
bass = new Sound();
bass.attachSound("fenderjazz");
guitar = new Sound();
guitar.attachSound("strat");
bass.start(0, 100);
bass.setPan(-75);
guitar.start(0, 50);
guitar.setPan(75);
```

В этом примере задается воспроизведение двух звуков после создания объектов для каждого из них и привязки звуковых файлов из библиотеки. После каждого запуска метод `setPan()` устанавливает «источник» звука в отдельную позицию: `bass` воспроизводится большей частью в левом канале, а `guitar` — большей частью в правом (*большой частью*, потому что 75 % звука воспроизводится через один канал, а 25 % через другой). Такая конфигурация панорамирования создает эффект стереозвучания, однако при этом звуки слегка смешиваются.

#### СОВЕТ

Точно так же с помощью метода `setPan()` можно управлять звуками в клипе или в загруженных SWF-файлах. Для этого при создании объекта `Sound` введите нужный аргумент.

Для выбора позиции панорамирования не существует никаких «золотых» правил. Чтобы добиться качественного звучания, вы должны иметь на компьютере двухканальную стереосистему. (Если у вас нет колонок, используйте наушники.)

## Программирование элементов управления уровнем громкости

Поскольку вы уже научились управлять уровнем громкости, можете, применив свои знания, создать экранные элементы управления уровнем громкости. В предыдущих главах рассказывалось, как управлять фильмом с помощью графических элементов. Вы использовали ActionScript для изменения визуальных свойств клипов, а теперь появилась возможность изменять свойства воспроизведения звука.

Вам уже известно, что метод `setVolume()` используется для программирования изменений уровня громкости. Этот метод совместно с некоторыми простыми командами в клипе создает гибкий и мощный контроллер уровня громкости.



Откройте файл `volumePanner.swf`, который находится в каталоге Chapter 22 на прилагаемом компакт-диске. В этом примере не показаны подробно все этапы создания графических элементов фильма. Для того чтобы понять, как он создан, вы можете взглянуть на структуру временной шкалы файла `volumePanner fla`.

Создайте набор кнопок управления уровнем громкости с помощью следующих действий.

1. Приготовьте все необходимые компоненты перед началом работы.
  - > Прежде всего, вам нужен объект управления — звук. Импортируйте звук в библиотеку и экспортируйте звуковой символ, используя окно `Linkage Properties`. (Импорт файлов подробно описан в главе 20. Детальная информация о добавлении звуков в фильм содержится в разделе «Привязка звуков с помощью связывающих идентификаторов» данной главы.)
  - > Вам также понадобится кнопка. Одной кнопки будет вполне достаточно, так как ее можно использовать несколько раз в одной и той же временной шкале. Но при желании можете создать и две кнопки. Более подробно об этом рассказывается в главе 13.
2. После того, как вы добавили звук и создали кнопку, активизируйте команду `Insert > New Symbol` (Вставка ▶ Новый символ). В поле `Name` введите имя клипа `volume` и выберите переключатель `Movie Clip`. Нажмите кнопку `OK`, и `Flash` перейдет в режим редактирования клипа.
3. Переименуйте первый слой в `actions` и введите семь новых ключевых кадров в этом слое. Щелкните на первом ключевом кадре и, если на экране нет панели `Actions`, активизируйте команду `Window ▶ Actions`. Введите следующие команды:

```
cue - new Sound();
cue.attachSound("yourLinkedSound");
cue.start(0,100);
```

В этом примере аргумент `yourLinkedSound` является именем звука, экспортированного с помощью окна `Linkage Properties`. При желании можно также изменить имя объекта `cue`, но делать это придется во всей программе.

4. Щелкните на втором ключевом кадре и снова используйте панель `Actions` для ввода таких команд:

```
cue.setVolume(100);
stop();
```

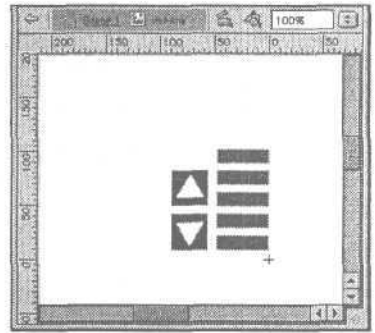
Эти команды определяют исходный уровень громкости и останавливают воспроизведение на кадре номер 2.

5. В остальных пяти ключевых кадрах введите команды, представленные в табл. 22.2. После чего закройте панель `Actions` и вернитесь к временной шкале клипа. Роль этих команд будет описана позже.

**Таблица 22.2.** Команды ключевых кадров для клипа volume

| Ключевой кадр | Команды                       |
|---------------|-------------------------------|
| 3             | stop();<br>cue.setVolume(80); |
| 4             | stop();<br>cue.setVolume(60); |
| 5             | stop();<br>cue.setVolume(40); |
| 6             | stop();<br>cue.setVolume(20); |
| 7             | stop();<br>cue.setVolume(0);  |

6. Выделите кнопку в библиотеке и перетащите ее на область действия клипа volume. Перетащите следующую кнопку на область действия рядом с первой кнопкой. Кнопки, используемые в файле `volumePanner.fla`, идентичны, но одна из них была повернута на 180 градусов, поэтому выглядит по-другому. Разный внешний вид кнопок не влияет на выполнение программы, но такое графическое оформление позволяет легко запомнить, какая из кнопок понижает уровень громкости, а какая повышает.



7. Щелкните на кнопке, которую вы будете использовать для понижения уровня громкости, и откройте панель Actions. Затем введите следующие команды:

```
on(release){
 if(_currentFrame<7){
 nextFrame();
 }
}
```

При щелчке на кнопке данный код проверяет позицию воспроизводящей головки в клипе. Если она установлена над кадром, номер которого меньше, чем 7, код выполняет действие `nextFrame()` — перейти на следующий кадр. В данном случае клип содержит семь кадров. Если воспроизводящая головка установлена над кадром 7, она при щелчке на кнопке не переходит на следующий кадр. Кроме того, поскольку каждый кадр содержит действие `stop()` и команду установки уровня громкости, при переходе на новый кадр останавливается движение клипа по временной шкале и изменяется уровень громкости объекта `Sound`.

8. Щелкните на кнопке, которую вы будете использовать для повышения уровня громкости, и откройте панель Actions. Затем введите следующие команды:


```
on(release){
 if(_currentFrame>2){
 prevFrame();
 }
}
```

Эта кнопка работает, как предыдущая. При каждом щелчке на кнопке, программа проверяет текущий кадр, на котором остановился клип. Если номер кадра больше 2, воспроизводящая головка переходит на предыдущий кадр (налево по временной шкале). При этом метод `setVolume()` каждый раз повышает уровень громкости звука до тех пор, пока воспроизводящая головка не достигнет кадра 2.

9. Закройте панель Actions и возвратитесь в главную временную шкалу. Выделите экземпляр клипа `volume` и перетащите его из библиотеки на область действия. На этом этапе вам не нужно присваивать имя экземпляру клипа. Активируйте команду Control ▸ Test Movie, чтобы запустить созданный контроллер. Если вы щелкнете на кнопке, понижающей уровень громкости, то услышите, как звук затихает; а при щелчке на кнопке, повышающей уровень громкости, — как звук усиливается.

## Программирование эффектов плавного нарастания и затухания звука

Для того чтобы воспроизведение звука начиналось и заканчивалось плавно, часто используются эффекты плавного нарастания (Fade-In) и плавного затухания (Fade-Out) звука. Эти эффекты позволяют управлять уровнем громкости звука аналогично созданному ранее контроллеру громкости. Существует только небольшое отличие: каждый такой эффект действует как переходный процесс; при его активизации медленно повышается или понижается уровень громкости. При этом вы должны организовать структуру клипа так, чтобы уровень громкости звука мог изменяться постепенно (в отличие от его одноразового изменения, которое происходило при нажатии кнопки в предыдущем примере).

 Откройте файлы `fadeInOut.swf` и `fadeInOut.fla`, находящиеся в каталоге Chapter 22 на прилагаемом компакт-диске. Посмотрите, как работают кнопки, с помощью которых задаются эффекты постепенного нарастания и затухания звука.

Для создания элементов управления, плавно включающих и выключающих звук, выполните такие действия.

1. Приготовьте все необходимые для работы компоненты.
  - ▷ В первую очередь импортируйте звук, которым вы будете управлять, в библиотеку. Затем экспортируйте его в фильм с помощью окна Linkage Properties. Импорт звуковых файлов подробно описан в главе 20, а экспортирование звуков в фильм — в разделе «Привязка звуков с помощью связывающих идентификаторов» данной главы.
  - > Кроме этого вам нужна хотя бы одна кнопка, которая будет служить элементом управления уровня громкости.
2. Активируйте команду Insert ▸ New Symbol. В поле Name введите имя клипа `fader` и выберите переключатель Movie Clip. Щелкните на кнопке OK, и Flash перейдет в режим редактирования клипа.
3. Переименуйте первый слой в `actions` и введите в него двенадцать новых ключевых кадров. Щелкните на первом ключевом кадре и, если на экране нет



панели Actions, активизируйте команду Window ▸ Actions. Введите следующие команды:

```
cue - new Sound();
cue.attachSound("yourLinkedSound");
cue.start(0,100);
cue.setVolume(100);
```

В этом примере аргумент `yourLinkedSound` является именем звука, экспортируемого с помощью окна Linkage Properties. Имя объекта `cue` вы также можете изменить, только это придется сделать во всей программе.

- Щелкните на втором ключевом кадре и введите команду `stop` на панели Actions. В остальные 11 ключевых кадров введите команды, приведенные в табл. 22.3.

**Таблица 22.3.** Команды ключевых кадров для клипа `fader`

| Ключевые кадры | Команды                                                         |
|----------------|-----------------------------------------------------------------|
| 3              | <code>cue.setVolume(80);</code>                                 |
| 4              | <code>cue.setVolume(60);</code>                                 |
| 5              | <code>cue.setVolume(40);</code>                                 |
| 6              | <code>cue.setVolume(20);</code>                                 |
| 7              | <code>cue.setVolume(0);</code>                                  |
| 8              | <code>stop()</code>                                             |
| 9              | <code>cue.setVolume(20);</code>                                 |
| 10             | <code>cue.setVolume(40);</code>                                 |
| 11             | <code>cue.setVolume(60);</code>                                 |
| 12             | <code>cue.setVolume(80);</code>                                 |
| 13             | <code>cue.setVolume(100);</code><br><code>gotoAndStop(2)</code> |

Работа над клипом `fader` закончена. Если вы посмотрите файл `fadeInOut.fla` на прилагаемом компакт-диске, то увидите, что в клипе есть несколько графических элементов. Они являются необязательными и не влияют на функциональность клипа. Закройте панель Actions и вернитесь к главной временной шкале.

- Перетащите экземпляр клипа `fader` на область действия и с помощью панели Properties присвойте ему имя `fade`.  
Файл `fadeInOut.swf` имеет графический элемент, представляющий светодиодный индикатор уровня громкости. Вы можете использовать его в своем фильме, но это совсем не обязательно.
- Создайте новый слой в главной временной шкале и присвойте ему имя `buttons`. Перетащите на этот слой две кнопки, которые будут служить в качестве элементов управления для плавного нарастания и затухания звука. При желании можете изменить графическое представление одной из кнопок.

- Щелкните на кнопке, создающей эффект fade-out, и откройте панель Actions. Введите следующие команды:

```
on(release){
 _root.fade.gotoAndPlay(3);
}
```

Данные команды управляют клипом fade. Они задают переход в кадр 3 клипа, поскольку именно с этого кадра начинает постепенно снижаться уровень громкости (см. табл. 22.3). Обратите внимание, что команда stop() содержится только в кадре 8. Начиная с кадра 3, метод setVolume() медленно понижает уровень громкости звука до тех пор, пока он не достигнет в кадре 7 минимального значения. После этого выполняется команда stopO.

- Щелкните на кнопке, создающей эффект fade-in, и откройте панель Actions. Введите следующие команды:

```
on(release){
 _root.fade.gotoAndPlay(9);
}
```

Данные команды переводят воспроизводящую головку в кадр 9 клипа fade. Начиная с этого кадра, уровень громкости плавно повышается и достигает максимального значения в кадре номер 13. После чего воспроизводящая головка переходит в кадр 2 и останавливается до следующего щелчка на кнопке.

- Активизируйте команду Control ▶ Test Movie и протестируйте ваш контроллер эффектов fade-in и fade-out.

Возможно, в результате тестирования, вы обнаружите, что затухание звука происходит слишком медленно или слишком быстро. Исправить это довольно просто.

Если затухание звука происходит слишком быстро, произведите одно из следующих действий.

- ▶ Задайте большую продолжительность анимации. Для этого увеличьте расстояние между ключевыми кадрами, в которых устанавливается уровень громкости.
- ▶ Задайте шаг увеличения уровня громкости, меньший, чем в предыдущем примере.

Для уменьшения времени затухания звука следует выполнить противоположные изменения.

## Создание контроллера панорамирования

Другой очень важный элемент управления звуком — панорамирование. Размещая звуки в стереополе, вы можете создать ощущение трехмерного пространства в фильме. Панорамирование позволяет располагать в пространстве «источники» звуков, создавая звуковые эффекты и осуществляя баланс позиций нескольких звуковых каналов. Объект Sound языка ActionScript можно использовать для создания элементов управления, динамически изменяющих позицию панорамирования заданного звука.



Файлы volumePanner.swf и volumePanner fla, содержащие готовый контроллер панорамирования, находятся в каталоге Chapter 22 на прилагаемом компакт-диске. Кроме кнопок, которые задают уровень громкости, контроллер

имеет регулятор, управляющий эффектом панорамирования. В этом примере представлены приемы создания подобных интерактивных графических элементов. Данная тема более подробно рассмотрена в главе 16.

Для создания эффекта панорамирования выполните следующие действия.

1. Приготовьте все необходимые элементы для работы.
  - > Во-первых, импортируйте звук в библиотеку и экспортируйте его в фильм, используя окно Linkage Properties.
  - > Во-вторых, создайте кнопку, которая будет служить в качестве ручки управления для регулятора.
2. Активизируйте команду Insert ► New Symbol. В поле Name введите имя клипа `slider` и выберите переключатель Movie Clip. Щелкните на кнопке ОК, и Flash перейдет в режим редактирования клипа.
3. Перетащите кнопку на область действия клипа `slider`. Используя панель Align, поместите данную кнопку в нужное место. Щелкните на кнопке и, если панели Actions нет на экране, активизируйте команду Window ► Actions. Введите следующие команды:

```
on(press){
 startDrag("",false,l,t,r,b);
 drag=true;
}
on(release,releaseOutside){
 stopDrag();
 drag=false;
}
```

Эти команды определяют параметры регулятора. (Более подробная информация содержится в главе 16.) Закройте панель Actions и вернитесь в режим редактирования фильма.

4. Создайте новый слой в главной временной шкале и присвойте ему имя `раппег`. Затем перетащите клип `slider` на область действия. Нарисуйте горизонтальную линию длиной в 100 пикселей на слое, расположенном ниже слоя `раппег`. (Для проверки длины вы можете использовать панель Info.) Эта линия является еще одним компонентом ползунка. Расположите ее так, чтобы горизонтальная ось кнопки совпала с линией (рис. 22.3). Значение позиции панорамирования не может быть больше 100. Поэтому, для того чтобы правильно выставить позицию панорамирования, необходимо использовать линию длиной в 100 пикселей.
5. Вы завершили создание графики. Теперь можно приступить непосредственно к программированию эффекта панорамирования.

Активизируйте команду Insert ► New Symbol. В поле Name введите имя клипа, установите переключатель Movie Clip и щелкните на кнопке ОК. Вставьте два ключевых кадра в первом слое. К первому кадру привяжите следующие команды:

```
cue = new Sound();
cue.attachSound("yourLinkedSound");
cue.start(0,100);
```

Здесь, как и в предыдущих примерах, `yourLinkedSound` — это имя звука, экспортируемого из библиотеки.

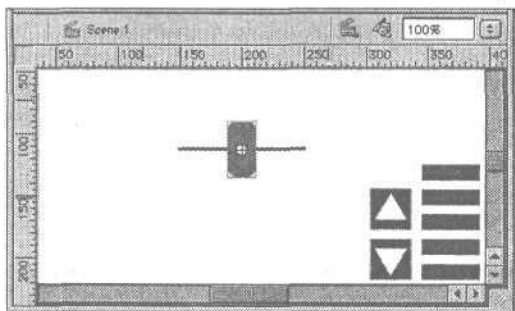


Рис. 22.3. Правильная позиция ручки управления регулятором

### ПРИМЕЧАНИЕ

В файле `volumePanner.fla` объект `Sound` создается в клипе `volume`.

Во втором ключевом кадре введите команду `stop()`. С помощью приведенных выше команд создается объект `Sound`, с которым связывается экспортируемый звук, и начинается воспроизведение звука (все в кадре 1). После выполнения этих команд клип останавливается в кадре 2, но звук продолжает воспроизводиться.

Вернитесь в режим редактирования фильма и перетащите клип на область действия главной временной шкалы. С помощью панели Instance (Экземпляр) присвойте экземпляру клипа имя `audio`.

6. Теперь введем программу для управления позицией панорамирования. Выберите клип `slider` и откройте панель Actions. Введите следующие команды:

```
onClipEvent(load){
 t = _y;
 b = _y;
 l = _x-50;
 r = _x+50;
 mid = _x;
}
onClipEvent(enterFrame){
 if(drag == true){
 _root.audio.cue.setPan((_x-mid)*2);
 }
}
```

Эти команды решают две задачи. При первой загрузке клипа в процедуре `onClipEvent(load)` задаются граничные значения позиции регулятора. Горизонтальные границы определяются путем прибавления и вычитания числа 50 из значения свойства `_x`. Таким образом, регулятор может перемещаться от центра линии к ее границам. Переменная `mid` содержит значение горизонтальной координаты регулятора при первоначальной загрузке клипа.

Второй фрагмент кода выполняется в результате каждого события `enterFrame`. Если пользователь перетаскивает регулятор, активизируется команда, которая управляет эффектом панорамирования. Третья строка в данном фрагменте кода задает позицию панорамирования с помощью метода `setPan`. Результат вычисления выражения  $(\_x\text{-mid})^2$  зависит от текущего значения горизонтальной координаты регулятора. Так как это значение находится между -50 и 50, в результате вычисления выражения всегда будет получено значение от -100 до 100.

7. Создание регулятора закончено. Активизируйте команду `Control ▶ Test Movie` для проверки результатов.

## Программирование специальных эффектов

Метод `setTransform()` применяется для создания разнообразных стереоэффектов в среде Flash. Его можно комбинировать с другими элементами языка ActionScript, как показано в приведенном далее примере.



На прилагаемом компакт-диске в каталоге Chapter 22 расположены три файла: `setTransformer fla`, `setTransformer.swf` и `stereoEffect fla`. Вы можете открыть эти файлы, посмотреть графику и прослушать музыку. Файл `stereoEffect fla` понадобится в дальнейшем, поэтому скопируйте его на жестком диске и приступайте к работе.

Для создания динамически управляемого стереоэффекта выполните следующие действия.

1. Предположим, что вы приготовили все необходимые элементы. В область действия уже помещен клип, который можно перетаскивать мышью, меняя при этом стереокомпозицию звуков. Теперь следует добавить несколько звуков в фильм.

На главной временной шкале создайте новый ключевой кадр в позиции 1 слоя actions. Откройте панель Actions и введите такие команды:

```
1 sfx = new Sound();
2 sfx.attachSound("stereo");
3 sfx.start(0,100);
4 xfadr = new Object();
5 xfadr.l1 - 50;
6 xfadr.l r - 25;
7 xfadr.r r = 50;
8 xfadr.r l - 25;
9 sfx.setTransform(xfadr);
```

Данный код определяет следующие действия. Команды с 1 по 3 создают объект `Sound` с именем `sfx`, связывают с ним звук и начинают его воспроизведение. Звук при этом уже должен быть экспортирован с именем `stereo`.

Командой 4 создается новый объект с именем `xfadr`. Этот объект будет использоваться в качестве объекта `transformer`. В нем будут храниться данные, применяемые к объекту `sfx`. Команды с 5 по 8 присваивают исходные значения свойствам объекта `transformer`. Последняя команда, 9, применяет свойства объекта `transformer` (храняемые в объекте `xfadr`) к объекту `sfx`.

Эти команды создают стереоэффект посредством распределения звука между правым и левым каналами. Когда клип находится в крайней левой позиции, звук слышен только в левом канале. Крайняя правая позиция клипа задает воспроизведение звука только в правом канале. Если клип находится в средней позиции, звук будет равномерно распределен между обоими каналами. При этом 25 % звука левого канала будет воспроизводиться в правом канале и 25 % звука правого канала — в левом канале. При перемещении ползунка в крайние позиции эти значения будут постепенно приближаться к нулю. В табл. 22.4 показаны процентные соотношения уровней громкости каналов в зависимости от позиции клипа.

**Таблица 22.4.** Уровни громкости каналов в зависимости от горизонтальной позиции клипа

| Позиция клипа <i>_x</i> | Уровень <i>ll</i> | Уровень <i>lr</i> | Уровень <i>rr</i> | Уровень <i>rl</i> |
|-------------------------|-------------------|-------------------|-------------------|-------------------|
| 0                       | 100               | 0                 | 0                 | 0                 |
| 100                     | 75                | 12,5              | 25                | 12,5              |
| 200                     | 50                | 25                | 50                | 25                |
| 300                     | 25                | 12,5              | 75                | 12,5              |
| 400                     | 0                 | 0                 | 100               | 0                 |

- Создайте еще один ключевой кадр в кадре номер 2 уровня `actions` и введите для него метод `stop()`. Таким образом, в кадре 1 будет создан стереоэффект, а в кадре 2 фильм остановится, но воспроизведение звука будет продолжаться. Теперь, когда все элементы установлены в исходное состояние, вы можете применить метод `setTransform()` для интерактивного управления стереозвуком.
- Щелкните на красной сфере, которая находится в центре области действия, и отобразите панель `Actions`. Вы увидите, что некоторые команды уже введены. Ознакомьтесь с этими командами, но не изменяйте их.

В окно уже существующего кода введите следующие команды:

```

1 onClipEvent(enterFrame){
2 if(drag == true){
3 _root.xfadr.ll=(400-_x)/4;
4 _root.xfadr.rr=_x/4;
5 if(_x>mid){
6 _root.xfadr.lr=Math.abs(((x-mid)/8)-25);
7 _root.xfadr.rl=Math.abs(((x-mid)/8)-25);
8 }else{
9 _root.xfadr.lr=_x/8;
10 _root.xfadr.rl=_x/8;
11 }
12 }
13 _root.sfx.setTransform(_root.xfadr);
14}

```

Давайте рассмотрим все строки этого кода по порядку и убедимся, что в них нет ничего необычного.

- **Строка 1.** Процедура `onClipEvent(enterFrame)` постоянно отслеживает позицию клипа, в результате чего обновляется информация, передаваемая через объект `transformer` объекту `sfx`.
  - **Строка 2.** Условие в команде `if` выполняется при перемещении клипа по области действия.
  - **Строки 3–4.** Клип можно перемещать в пределах 400 пикселей. Данными командами задаются значения для свойств `l1` и `r1`. С этой целью используются выражения, содержащие текущую позицию. Если клип движется влево, значения `l1` приближаются к 100, а `rr` — к 0, и наоборот.
  - **Строка 5.** Начальное значение горизонтальной координаты клипа составляет 200 пикселей (оно хранится в переменной `mid`). Поэтому команда `if` применяется для определения необходимости использования объекта `Math` при формировании значений `lr` и `rl`. Данный объект применяется в случае перемещения клипа в сторону увеличения горизонтальной координаты. Тогда выполняются команды в строках 6 и 7.
  - **Строки 6-7.** Эти выражения генерируют значения для свойств `lr` и `rl`, если горизонтальная координата клипа больше значения переменной `mid` (200). Возвратитесь к табл. 22.4 и посмотрите, как соотносятся между собой позиция клипа и значения свойств `lr` и `rl`. Для того чтобы этим свойствам невозможно было присвоить отрицательные значения, используется метод `Math.abs()`, который вычисляет абсолютные величины.
  - **Строка 8.** Если условие в команде `if` не выполняется (координата `_x` не больше, чем `mid`), осуществляется переход к строке, следующей за ключевым словом `else`.
  - **Строки 9-10.** Вычисляются значения свойств `lr` и `rl` по формуле `_x/8`.
  - **Строка 13.** Эта команда при помощи метода `setTransform()` передает данные от объекта `xfadg` объекту `sfx` на главной временной шкале.
4. Закройте панель `Actions` и активизируйте команду `Control▶Test Movie`. Переместите сферу из стороны в сторону и послушайте, как изменяется звук. Если клип движется влево, то громкость звука уменьшается в правом канале и увеличивается в левом. Если клип движется вправо, все происходит наоборот.
- Как же все это можно использовать в настоящем фильме? Предположим, что вы создаете игру, в которой герой исследует мир. Рассмотренный прием позволит, увеличивая уровень громкости, создавать эффект приближения героя к источнику звука. Этот прием можно также использовать для динамического изменения звуковых эффектов. Например, вы имитируете характерный звук, издаваемый пропеллером самолета. Если самолет приближается, звук становится громче и наоборот.

## Программирование загрузки внешних звуков


**МХ** Одной из новых возможностей Flash МХ является загрузка внешних звуковых файлов в формате MP3. Теперь вы можете не импортировать звуки в фильм, а вызывать их, когда они потребуются.

Преимущества такого подхода очевидны. Во-первых, в отличие от импортируемых звуков, которые увеличивают размер SWF-файла, внешние звуки не изменяют его. Это позволит ускорить загрузку фильма. Во-вторых, можно создавать общие библиотеки звуков или загружать звуки из сети Интернет. Посетители узла смогут, указав адрес, прослушивать интересующую их музыку.

Возможности Flash можно перечислять очень долго. Их ограничивают только ваши время и фантазия. Ниже представлены дополнительные методы объекта Sound для работы с внешними звуковыми файлами:

- ▶ `Sound.loadSound("url", stream)`. Используется для загрузки внешних файлов в определенный объект Sound. Данный метод имеет два аргумента. Первый аргумент — это строка, в которой указывается URL файла, а второй — переменная типа `boolean`. Если его значение равно `false`, звук должен перед воспроизведением полностью загрузиться. Если же установлено значение `true`, то звук является потоковым и начинает воспроизводиться, не дожидаясь полной загрузки.
- ▶ `Sound.getBytesLoaded()`. Используется для определения количества загруженных байтов внешнего звука.
- ▶ `Sound.getBytesTotal()`. Возвращает общее количество байтов для заданного объекта Sound.

Вы можете применять методы `Sound.getBytesLoaded()` и `Sound.getBytesTotal()` для отображения индикатора процесса загрузки внешних звуковых файлов.

 Теперь, когда вы познакомились с новыми методами, можно приступить к работе. Чтобы упростить вашу задачу, на прилагаемом компакт-диске приготовлен исходный файл. Зайдите в каталог Chapter 22, откройте файл `loadMP3.fla` и скопируйте его на жесткий диск.

Для загрузки внешнего файла выполните следующие действия.

1. Прежде всего, надо иметь MP3-файл, который нужно загрузить. Мы рекомендуем вначале использовать небольшой файл, чтобы при тестировании этой процедуры не тратить много времени на загрузку. Поместите MP3-файл на ваш Web-сервер и запомните его адрес.

### СОВЕТ

Если вы не имеете своего Web-сервера или MP3-файла для загрузки, воспользуйтесь файлом `savvySound.mp3`, который находится по адресу [www.vonflashenstein.com/resources/savvySound.mp3](http://www.vonflashenstein.com/resources/savvySound.mp3).

2. В файле `loadMP3.fla` выберите первый (и единственный) ключевой кадр слоя 1. Активизируйте команду Actions в меню Window или нажмите клавишу F9 для отображения на экране панели Actions.



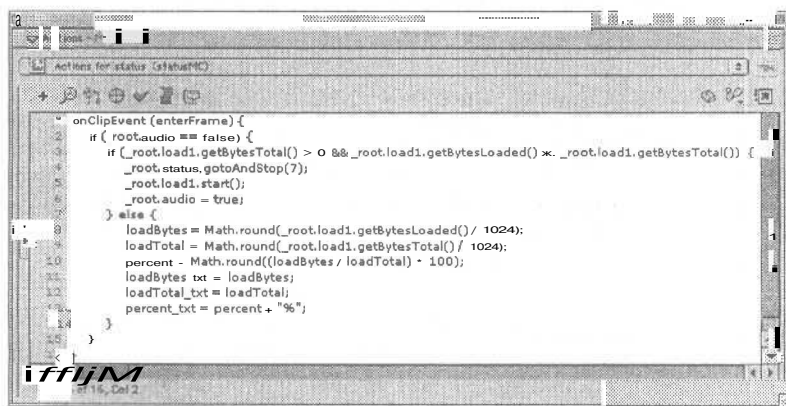


Рис. 22.4. Команды первого ключевого кадра файла loadMP3 fla

- Во второй строке кода, находящегося на этой панели, содержится текст "enterYourURLHere". Замените ее абсолютным адресом URL загружаемого файла. Адрес указывается так: `www.myDomain.com/file.mp3`, где `myDomain` — имя вашего домена и `file.mp3` — имя MP3-файла.

Первая строка кода создает объект `Sound` с именем `load1`. Во второй используется метод `loadSound()` для загрузки указанного внешнего файла. Значение второго аргумента этого метода равно `false`. Это означает, что звук должен полностью загрузиться перед воспроизведением. Третья строка присваивает переменной `audio` значение `false` (оно будет использоваться позже).

- Выберите клип `status` в области действия вашего фильма (рис. 22.5). Откройте панель `Actions`. С клипом `status` связаны команды, контролирующие процесс загрузки звука.



Давайте рассмотрим каждую строку этого кода.

- > **Строка 2.** Переменная `audio` используется в качестве признака завершения загрузки файла. Если ее значение равно `false`, то звук находится в процессе

загрузки. Когда звук загружен, в строке 6 значение переменной изменяется на true.

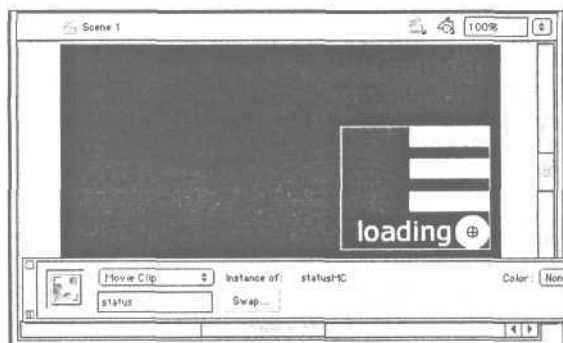


Рис. 22.5. Клип status

- **Строка 3.** Проверяется состояние загрузки звукового файла. Программа проверяет выполнение двух условий: размер звукового файла больше нуля; весь файл загружен. Когда эти условия выполняются, управление передается строкам 4-6.

#### СОВЕТ

Для файлов, которые загружаются слишком медленно, первое условие (значение `Sound.getBytesTotal()` больше нуля) предотвращает преждевременное прекращение загрузки файла. Если загрузка файла не завершена, его размер и количество загруженных байтов равны нулю. В этом случае второе условие выполнится, и, если не будет первого условия, Flash решит, что файл полностью загрузился.

- **Строки 4-6.** После полной загрузки звука эти команды обновляют экземпляр объекта `status`, запускают звук `load1` и присваивают переменной `audio` значение `true` для предотвращения повторных загрузки и воспроизведения звука.
- **Строки 8-13.** Если звук пока не загружен полностью, выполняются команды вычисления значений, применяемых для отображения процесса загрузки. Вот эти значения:

`loadBytes` — количество загруженных байтов.

`loadTotal` — общее количество байтов.

`percent` — процент загрузки файла.

В строках 8 и 9 используется метод `Math.round()` для округления значений счетчика байтов. Значения делятся на 1024 для представления их в килобайтах.

5. Активизируйте команду **Control ▶ Test Movie** для проверки процесса загрузки звука. При этом вы будете видеть на экране индикатор хода загрузки. Когда звук загрузится, в центре вращающегося колеса появится зеленая точка, а рядом с колесом отобразится надпись `playing`.

Если звук не загрузился, проверьте следующее:

- ▶ подключен ли ваш компьютер к сети Интернет,
- ▶ какова скорость соединения,
- ▶ правильно ли указан адрес URL.

В данном разделе рассматривалось использование внешних звуковых файлов. Более подробная информация об этих методах и действиях содержится в приложении Б.

## Разработки

Создать настоящее интерактивное приложение достаточно сложно. Основная проблема заключается в разработке интересного и удобного пользовательского интерфейса. Главная задача — сделать интерактивной звуковую часть приложения. В фильмах, где отсутствует интересная графика, целесообразно создать мощную интерактивную звуковую среду.

Посетите Web-страницу [www.pianographique.com](http://www.pianographique.com) (рис. 22.6), созданную Жаном Люком Ламарком в соавторстве с группой других музыкантов и художников. Она представляет собой сборник интерактивных фильмов с высококачественными музыкой и графикой.

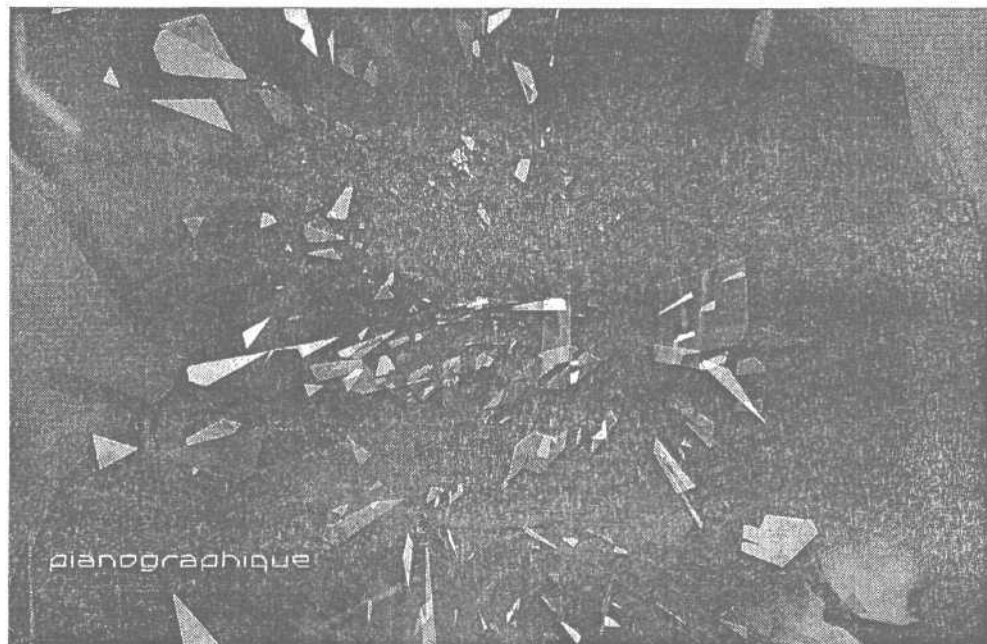


Рис. 22.6. Web-страница Angular entropy

## Резюме

В данной части книги были представлены основы использования цифрового звука. Вы научились внедрять звуки в фильм, импортировать их в библиотеку, синхронизировать звук и анимацию. Рассмотренные в этой части материалы содержат обширные сведения о возможностях, свойствах и роли в управлении звуком объекта Sound, а также о методах ActionScript. Теперь вы знаете, как начинать и заканчивать воспроизведение звуков, устанавливать их уровень громкости и параметры панорамирования. В данной главе рассказывалось, что использование ActionScript позволяет с помощью различных контроллеров организовать управление уровнем громкости, панорамированием, эффектами Fade-In и Fade-Out. В следующей части книги перечисленные сведения понадобятся вам для создания интерактивных фильмов с многоканальным звуком.

## Практикум 6

# Создание звукового сопровождения фильма

Создание звукового сопровождения фильма — задача не из простых. Озвучивая свой проект, вы столкнетесь с множеством проблем. В частности, придется позаботиться об ограничении объемов результирующих файлов, особенно если вы собираетесь публиковать фильм на Web-узле. Цифровые звукозаписи, как правило, занимают довольно большой объем, поэтому звуковое сопровождение целесообразно сохранить в виде отдельного файла и загружать в процессе воспроизведения фильма. Кроме того, много проблем возникает при разработке интерактивного интерфейса фильма.


Измерение, в котором существует звук, — это время. Звуки имеют различную продолжительность звучания, поэтому их довольно сложно комбинировать. Чтобы можно было услышать звуки, воспроизведение которых нельзя заранее запланировать, необходимы интерактивные средства взаимодействия с пользователем. Разработка интерактивного продукта — дело достаточно сложное. Успех проекта в основном зависит от вашего мастерства, творческих способностей, а также от имеющихся в наличии звуковых файлов. Хорошая подборка аудиоматериалов поможет вам создать удачное звуковое оформление фильма.

Компания DoReMedia предоставляет первоклассный звуковой и музыкальный материал, который может быть легко включен во Flash-фильмы. Выполнив приведенное далее практическое задание, вы научитесь использовать различные звуки и музыкальные компоненты для озвучивания фильмов Flash и обеспечения интерактивной навигации.

## Планирование использования звуков в фильме

Поскольку в любом деле анархия редко приводит к хорошим результатам, следует составить хотя бы приблизительный план работы. Постарайтесь представить проект в готовом виде, и вы сможете наметить путь создания интерактивного фильма. В этом случае у вас сразу появится множество более или менее удачных идей. Советуем не отбрасывать даже те идеи, которые на первый взгляд кажутся неподходящими для вашего проекта.

Изучив приведенное далее практическое задание, вы получите представление о технике создания звукового сопровождения фильмов и сможете применить ее для озвучивания Flash-фильмов.

 Прежде всего, откройте файл `whiskey100.swf` из каталога Hands On 6, находящейся на прилагаемом компакт-диске (или файл проекта `whiskey100 fla`). Поэкспериментируйте с элементами интерфейса этого фильма и послушайте музыку, звучащую в различных его частях. Вы услышите, как согласованно работают

## Глава 29

# Публикация Flash-фильма на компакт-диск

Одним из достоинств Flash является его способность комбинировать высококачественный звук и анимацию в одном файле. Данная возможность делает Flash идеальной средой создания фильмов для Web, а также для других приложений, где особенно важна пропускная способность канала. Но при этом не стоит забывать, что Flash-фильмы можно распространять и другими способами, для которых время загрузки и наличие подключаемых модулей не имеют значения.

Одним из наиболее эффективных способов донесения Flash-фильмов до зрительской аудитории является их публикация на компакт-дисках в виде самостоятельных приложений или исполняемых файлов. В таком формате Flash-фильмы могут воспроизводиться либо с помощью приложения Flash Player, либо независимо от других приложений как универсальные исполняемые файлы для платформ Windows и Macintosh. Кроме того, учтите, что на компакт-диск можно поместить огромное количество информации, не будучи ограниченным пропускной способностью канала связи, как при распространении фильмов в Web.

Конечно, ни один из методов публикации не является универсальным, но каждый из них имеет свои преимущества и позволяет расширить возможности Flash. В данной главе будут рассмотрены следующие темы:

- ▶ автономные фильмы для приложения Flash Player;
- ▶ исполняемые файлы Windows projector и Macintosh projector;
- ▶ публикация фильмов для распространения на компакт-дисках;
- ▶ использование действия `fscommand` языка ActionScript;
- ▶ разработка и написание сценариев для элементов интерфейса фильмов, распространяемых на компакт-дисках.

## Автономные Flash-файлы

Хотя Flash в основном применяется для создания содержимого Web-узлов, не следует забывать, что с помощью данного приложения можно также создавать фильмы, формат которых позволяет распространять их на других носителях. Используя Flash, можно создавать автономные фильмы двух типов, предназначенные для воспроизведения на компьютерах без подключения к Интернету. Первый — стандартный файл с расширением SWF. Если на компьютере зрителя установлено приложение Flash Player, фильмы формата SWF можно просматривать без помощи Web-браузера. Другим форматом автономного фильма является исполняемый файл типа Flash projector, представляющий собой приложение, позволяющее воспроизводить Flash-фильм на любом компьютере независимо от установки приложения Flash Player.

## Разработки

Ушли в историю те времена, когда Flash-фильмы можно было встретить только в Интернете и на компакт-дисках. С расширением возможностей приложения расширилась и область его применения.

Английская компания Kerb ([www.kerb.co.uk](http://www.kerb.co.uk)), специализацией которой является разработка продуктов на основе Flash, создала первый анимированный телевизионный сериал исключительно с использованием Flash. Новые серии фильма Hellz Kitchen (рис. 28.9) размещаются на Web-узле компании Bravo Network ([www.bravo.co.uk/toonz/hellz/hellz\\_archive.html](http://www.bravo.co.uk/toonz/hellz/hellz_archive.html)), получившей права показа. На телевизионных каналах сериал демонстрировался в 2001 году.

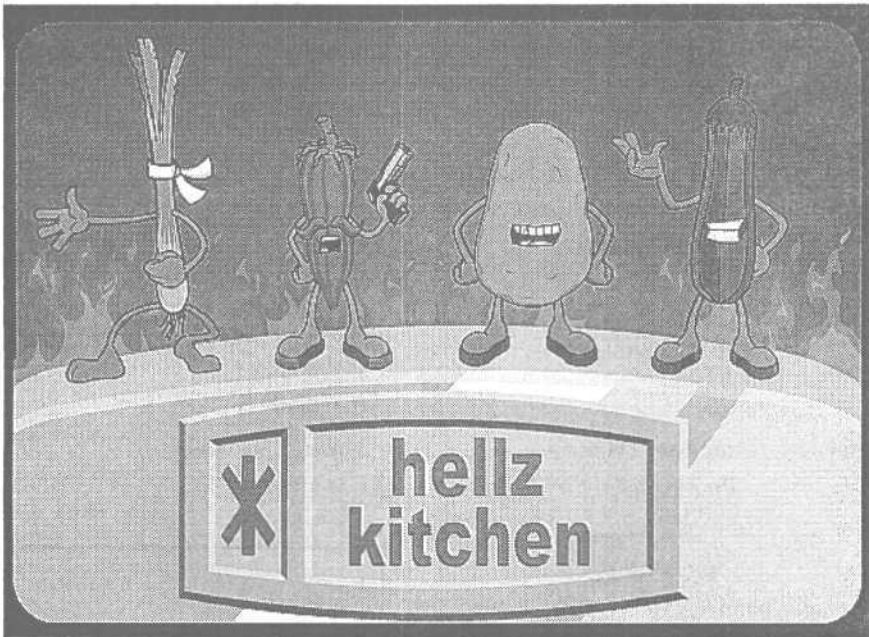


Рис. 28.9. Кадр из анимированного телевизионного сериала об агрессивной компании овощей

## Резюме

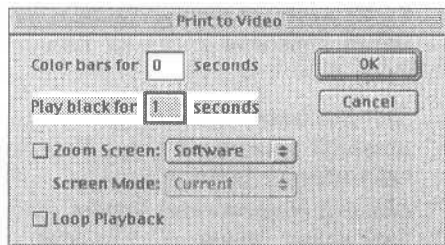
Эта глава посвящена работе Flash с файлами других видеоформатов. В приложении Flash MX реализована поддержка и вложенных, и внешних файлов, поэтому теперь стало осуществимо помещение видеоданных непосредственно во Flash-фильм. Это свойство значительно расширяет возможности Flash и позволяет создавать фильмы на основе как векторной, так и растровой графики. В главе также были рассмотрены вопросы, связанные с экспортированием векторной анимации Flash в растровые цифровые видео, и описан процесс переноса цифровых видеофильмов на видеокассеты.

Используя Adobe Premiere, можно передавать данные с временной шкалы непосредственно на видеокамеру, видеомэгнитофон или на другое цифровое устройство записи видеосигнала. Мы не будем подробно рассказывать о данном процессе, поскольку для различных компьютерных систем, графических адаптеров и устройств записи видеосигнала подходы существенно различаются. Здесь мы остановимся лишь на общих принципах, универсальных для всех типов устройств.

- ▶ При передаче видеосигнала непосредственно на видеомэгнитофон или другое устройство Adobe Premiere использует параметры, установленные в окне Project Settings (Параметры проекта). Убедитесь, что эти параметры соответствуют наилучшему качеству фильма.
- ▶ Для установки устойчивой синхронизации до начала воспроизведения анимации необходимо включить в начало фильма стандартный набор цветных полос или темный экран. При использовании Adobe Premiere для создания проекта, который в дальнейшем будет обрабатываться в телевизионном центре компоновки программ, рекомендуется включать начальный сигнал с изображением цветных полос и звуковой тон продолжительностью не менее 30 секунд.

Для публикации проекта на видеоленте выполните следующие действия.

1. Если предварительный просмотр еще не выполнялся, выберите команду Preview меню Timeline.
2. Подключите устройство записи и убедитесь в наличии видеокассеты.
3. Выберите команду File ▶ Export Timeline ▶ Print to Video. В открывшемся окне Print to Video установите следующие параметры:
  - > в поле Color Bars (Цветные полосы) задайте продолжительность воспроизведения сигнала цветных полос в секундах;
  - > в поле Play Black (Темный экран) установите продолжительность воспроизведения сигнала темного экрана;
  - > установите флажок Zoom Screen (Macintosh) или Full Screen (Windows) для масштабирования видео до размеров экрана. При работе в среде Macintosh данную операцию можно осуществить как аппаратными (с помощью графического адаптера), так и программными средствами;
  - > при необходимости циклического воспроизведения фильма без остановки установите флажок Loop Playback (Циклическое воспроизведение).
4. Включите запись на видеомэгнитофоне и щелкните на кнопке ОК, обо всем остальном позаботится Premiere.
5. Прежде чем остановить запись на видеомэгнитофоне, нажмите клавишу Esc. Premiere остановит передачу сигнала. Перемотайте видеоленту назад и посмотрите на видеокассете Flash-фильм.





Чтобы поместить видеоклип на временную шкалу Premiere, выполните следующие действия.

1. Выберите видеоклип в окне Project. Указатель мыши отобразится в виде изображения руки.
2. С помощью мыши перетащите клип из окна на временную шкалу и поместите его в канал Video 1. При использовании в клипе звука его звуковая дорожка появляется непосредственно под клипом в канале Audio 1. Если звук перенесен из Flash-фильма, его необходимо удалить, поскольку при одновременном переносе аудио- и видеоданных звуковые данные будут повреждены.
3. *Данный шаг необходим, если ваш клип содержит звук.* Щелкните на кнопке Toggle Sync Mode (Переключатель режима синхронизации), при этом будет отключена синхронизация видео- и звуковых компонентов файлов на временной шкале. Выберите звуковую дорожку и нажмите клавишу Delete. Звуковая дорожка будет удалена.
4. Импортируйте звук, который вы ранее создали. При работе на компьютере Macintosh звуковая дорожка создается с помощью приложения QuickTime Pro в формате AIFF. При работе в среде Windows Flash позволяет экспортировать звук в файл формата WAV.
5. С помощью мыши перетащите звуковую дорожку для Flash-фильма из окна Project на временную шкалу Premiere и поместите ее в канал Audio 1. Поскольку из данного канала была извлечена звуковая дорожка исходного фильма, новая звуковая дорожка точно совпадет по размеру с дорожкой в канале Video 1.
6. Для предварительного просмотра видео выберите команду Preview (Предварительный просмотр) меню Timeline (Временная шкала) или нажмите клавишу Return/Enter. Данные на видео- и звуковой дорожках будут воспроизведены в области Program окна Monitor. Данный шаг необходим, потому что без предварительного просмотра приложение Adobe Premiere не сможет воспроизвести фильм.

Все последующие действия зависят от особенностей вашего проекта. Если вы хотите использовать только Flash-фильм, предварительную работу можно считать завершенной и приступить к переносу информации на видеоленту. Если же в проекте Flash-фильм будет комбинироваться с другими видеоматериалами, вам следует продолжить импорт видеофайлов, их размещение на временной шкале и использование окна Source (Источник) для манипулирования компонентами фильма.

## Экспортирование и публикация на видеоленту

Приложение Adobe Premiere позволяет переносить отредактированный фильм на видеоленту. Этот процесс называется *публикацией на видеоленту*. В зависимости от типа оборудования данную операцию можно выполнять различными способами.

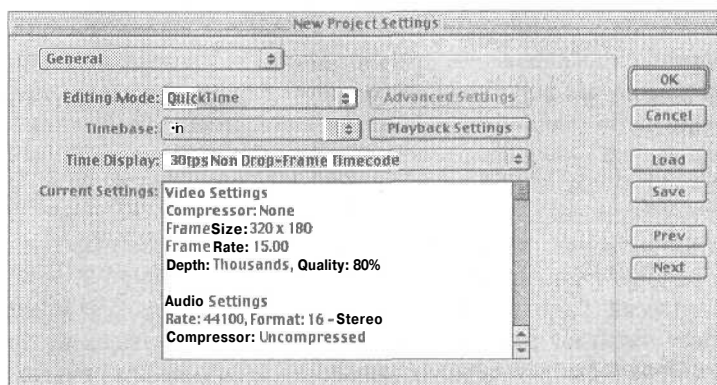


Рис. 28.8. Окно New Project Settings приложения Adobe Premiere

После создания проекта все готово для начала работы над экспортированным из Flash фильмом. Теперь необходимо импортировать этот фильм. В отличие от Flash приложение Adobe Premiere не сохраняет импортированные файлы в библиотеке. При передаче какого-либо компонента фильма в среду Premiere создается ссылка на файл, содержащий данные, а Adobe Premiere обращается к этому файлу при необходимости получить данные. Другими словами, импортируемый файл остается на жестком диске и фактически никогда не является составной частью файла Premiere-проекта. По этой причине нельзя удалять исходные файлы, которые используются в Premiere-проектах.

Чтобы импортировать видеоданные, экспортированные из Flash, выполните следующие действия.

1. Выберите команду меню File.
2. В открывшемся диалоговом окне Import File укажите путь к исходному видеофайлу, который вы собираетесь импортировать. После выбора файла щелкните на кнопке Open. Если вы хотите использовать Flash-фильм с другими видео- и звуковыми клипами, можете одновременно импортировать необходимое количество клипов. Для одновременного выбора нескольких файлов следует при выборе файлов-источников удерживать клавишу Shift (Macintosh) или Ctrl (Windows).
3. В окне Premiere-проекта появится один или несколько значков, указывающих на использование внешних файлов-источников. После импортирования всех необходимых фрагментов можно приступить к построению проекта.

## Компоновка видеопроекта

Вы уже определили задачи своего проекта и импортировали некоторые исходные компоненты будущего фильма. Теперь необходимо соединить отдельные части в единое целое. Среда Adobe Premiere позволяет очень просто выполнить данную операцию: компоновка осуществляется посредством перетаскивания используемых клипов на временную шкалу Premiere. Временная шкала Premiere очень похожа на временную шкалу Flash. При помещении на нее клипа он станет частью видеопроекта и будет включен в конечный фильм.

видеофильмов профессионального уровня на основе использования цифровых видеоматериалов. С помощью приложения Adobe Premiere нельзя переносить видео непосредственно из источника, но можно принимать видеоданные, записанные с помощью цифровой или аналоговой видеокамеры. Данное приложение содержит набор специальных инструментов и средств, позволяющих создавать как короткометражные, так и полнометражные фильмы. Кроме того, приложение Adobe Premiere позволяет сохранять отредактированные фильмы в различных видеоформатах, включая формат для видеокассет, и различные кодеки, применяемые при распространении видеофильмов на компакт-дисках или в сети Интернет. Следующий раздел посвящен основным принципам работы в среде Adobe Premiere. Если вы ранее не работали с этим приложением, рекомендуем начать знакомство с файлом справки формата HTML, доступ к которым можно получить, выбрав команду Help Topics (Темы справки) меню Help.

## Импортирование Flash-компонентов в редактор видео

Исходной точкой является запуск приложения и начало работы с новым проектом. При первом запуске программы появится диалоговое окно Load Project Settings (Загрузка параметров проекта), в котором необходимо указать характеристики фильма. Если ни один из имеющихся в списке типов проектов вам не подходит, щелкните на кнопке Custom. При этом откроется окно New Project Settings (Параметры нового проекта) (рис. 28.8), где можно установить значения параметров специально для вашего проекта.

1. Установите параметры Premiere-проекта, используя список, находящийся в левом верхнем углу диалогового окна New Project Settings:
  - General (Общие). Выберите режим редактирования (QuickTime или Video for Windows) для своей системы. Установите в поле Timebase значение равное или кратное частоте смены кадров (например, для частоты смены кадров 12 fps это может быть значение 24);
  - Video (Видео). Выберите тип сжатия, значения глубины цвета, размера кадров, частоты смены кадров и уровень качества;
  - Audio (Звук). Установите параметры частоты выборки, битовой глубины и степени сжатия для звука вашего проекта;
  - Keyframe and Rendering (Ключевые кадры и визуализация). Используйте данный параметр для управления специфическими настройками визуализации и воспроизведения фильма;
  - Capture (Сбор данных). Поскольку Flash-фильм уже содержит видеоданные, вы не будете использовать эту функцию.
2. После установки всех параметров щелкните на кнопке ОК. При этом будет открыт новый проект.
3. Сохраните файл.

В результате экспортирования вы получите цифровой видеофайл, готовый для работы в среде видеоприложения и распространения на компакт-дисках. Вопросы, связанные с подготовкой файла для переноса на видеоленту будут рассмотрены далее.

## Экспорт звука Flash-фильма в файл формата WAV

Если вы планируете работать с видеофильмом в среде приложения Adobe Premiere, то должны экспортировать видео и звук отдельно. Это позволит сохранить высокое качество звука при переносе звуковых данных Flash-фильма в Adobe Premiere.

Подготовьте звуковые данные Flash-фильма к экспортированию в файл формата WAV в соответствии с инструкцией раздела «Подготовка звуковых компонентов фильма Flash», а затем перейдите непосредственно к процессу экспортирования. Для этого выполните действия, перечисленные ниже.

1. Выберите из меню File команду Export Movie. Откроется одноименное диалоговое окно, где следует указать имя файла и каталог, в котором вы хотите сохранить конечную версию файла. В раскрывающемся списке Save as Type (Сохранить в формате) выберите пункт WAV Audio. Щелкните на кнопке Save.
2. С помощью открывшегося диалогового окна Export Windows WAV (Экспорт в формат Windows WAV) устанавливаются параметры экспортируемого звука:
  - ▷ в списке Sound Format (Формат Звука) задается частота выборки, битовая глубина и режим (стерео/моно) для экспортируемого звукового файла;
  - > при выборе флажка Ignore Event Sounds (Игнорировать звуки событий) из экспортируемого файла исключаются звуки, привязанные к событиям.
3. Выполнив все установки, щелкните на кнопке OK. Flash экспортирует звуковые компоненты в файл формата WAV, который будет сохранен в указанном вами каталоге. Затем при работе в среде Adobe Premiere вы удалите звуковые данные исходного Flash-фильма и импортируете новые из файла WAV.

## Подготовка видео для переноса на видеоленту

Пошаговое выполнение подготовки, переход от одного окна к другому, изменение взаимного расположение кадров и установка различных параметров сжатия видео — задача, конечно, сложная и трудоемкая. Но теперь мы вплотную подошли к завершающему этапу: раздел посвящен описанию того, как используется приложение Adobe Premiere для переноса Flash-фильма формата QuickTime или AVI на видеоленту. Здесь вы только познакомитесь с основными базовыми принципами данного процесса. Мы не сможем рассказать о деталях, поскольку для этого понадобилось бы подробно рассмотреть каждый вариант установки параметров видео и все особенности различного оборудования при работе с Adobe Premiere.

Версия 6.0 приложения Adobe Premiere, самая новая на момент написания данной книги, представляет собой программу для редактирования, сборки и создания

видео и задать его параметры. Чтобы уменьшить размер конечного файла формата AVI рекомендуется использовать сжатие данных.

- Smooth (Сглаживание). При установке флажка Smooth будет использоваться функция сглаживания. Учтите, что использование данной функции может привести к появлению помех на изображении.
- Sound Format (Формат звука). Этот список позволяет выбрать параметры, влияющие на качество звука в экспортируемом фильме. Более подробно о частоте выборки, битовой глубине и других характеристиках звука речь шла в главе 19. При выборе пункта Disable (Блокировать) звуковые компоненты фильма экспортироваться не будут.

#### ПРИМЕЧАНИЕ

Если вы готовите фильм для редактирования с помощью другого приложения, например Adobe Premiere, рекомендуем выбрать пункт Disable в списке Sound Format. Уже говорилось, что звук в файле AVI, экспортированный из Flash-фильма, не будет работать в среде Adobe Premiere, поскольку при переносе звук повреждается. Для сохранения качества звука необходимо экспортировать две версии Flash-фильма: видео в формате AVI без звука и звук в формате WAV. Об экспорте звука Flash-фильма в файл формата WAV мы расскажем в следующем разделе.

3. Для подтверждения всех установок щелкните на кнопке ОК. Flash-фильм будет экспортирован в файл формата AVI.
4. Если вы используете сжатие видео, появится дополнительное диалоговое окно Video Compression, в котором можно выбрать один из доступных кодеков и установить для него некоторые параметры.
  - Compressor (Система сжатия). Используя этот список, можно выбрать кодек сжатия видео, наиболее подходящий для вашего видеоборудования и программного обеспечения. Список рекомендуемых кодеков обычно содержится в документации к видеоборудованию (или программному обеспечению). Если вы собираетесь распространять фильм в качестве отдельного файла формата AVI, советуем использовать кодек Сinераk.
  - Compression Quality (Качество сжатия). С помощью этого регулятора устанавливается степень сжатия видео.
  - Key Frame Every (Интервал вставки ключевого кадра). Установив данный флажок, вы получаете доступ к полю frames, где можно задать значение интервала вставки ключевых кадров. Ключевые кадры помогают системе сжатия работать с видео. Наиболее приемлемым считается интервал, равный частоте смены кадров. Например, при значении частоты смены кадров 30, оптимальной будет вставка через каждые 30 кадров.
  - Data Rate (Скорость передачи данных). При установке флажка Data Rate вы получаете доступ к полю KB/sec, в котором следует указать скорость передачи данных в Кб/с. Скорость передачи данных 150 Кб/с достаточна для нормального воспроизведения видео с компакт-диска.
5. После установки всех параметров сжатия видео щелкните на кнопке ОК.

Interchange File Format -- формат звукового файла на компьютерах Macintosh);

- > в раскрывающемся списке Use выберите пункт, соответствующий требуемой частоте выборки и битовой глубине. Мы рекомендуем использовать максимально возможное качество.

3. Выполнив все установки, щелкните на кнопке Save (Сохранить). QuickTime экспортирует звуковую дорожку в файл формата AIFF, который будет сохранен в указанном вами каталоге.

Теперь у вас есть все необходимые компоненты для создания видео. Позже, работая в среде Adobe Premiere, вы удалите звук исходного Flash-фильма и импортируете новый из файла AIFF.

## Видео для Windows

Если вы работаете с Flash в среде Windows, то будете экспортировать видео в файл формата AVI (Audio Video Interleaved). Как и QuickTime, формат AVI основан на использовании растровой графики. Для экспортирования анимации Flash в файл формата AVI выполните следующие действия.

1. Выберите команду Export Movie меню File. В открывшемся окне Export Movie укажите имя файла и каталог, в котором хотите сохранить файл. Затем выберите пункт Windows AVI в списке Save as Type (Сохранить в формате). Щелкните на кнопке Save.
2. Откроется окно Export Windows AVI (Экспорт в формат Windows AVI), позволяющее установить все параметры для экспортируемого видео.
  - > Dimensions (Размеры). Данные поля предназначены для установки значений Width (Ширина) и Height (Высота) конечного видео. За исключением случаев, когда Flash-фильм содержит растровые изображения, размер фильма может быть изменен без внесения искажений. При установке флажка Maintain Aspect Ratio (Сохранить формат изображения) пропорции фильма, определенные в окне Document Properties (Свойства документа), будут сохранены.
  - > Video Format (Видеоформат). С помощью этого списка можно установить следующие параметры глубины цвета для вашего фильма:
    - 8-bit color (Глубина цвета 8 битов) — низкое качество;
    - 16-bit color (Глубина цвета 16 битов) — соответствует установке High Color на компьютере Windows;
    - 24-bit color (Глубина цвета 24 бита) — соответствует установке True Color на компьютере Windows;
    - 32-bit color (Глубина цвета 32 бита) — то же, что 24 бита плюс дополнительная поддержка данных Alpha-канала для эффектов прозрачности. Данное свойство поддерживается не всеми проигрывателями. Проверьте это в документации приложения для воспроизведения видеофайлов, прежде чем экспортировать фильм в таком формате.
  - > Compress Video (Сжатие видео). Если установлен флажок Compress Video, при щелчке на кнопке ОК текущего окна откроется дополнительное диалоговое окно, с помощью которого можно выбрать один из кодеков сжатия

- Sound Format (Формат звука). Этот список позволяет выбрать параметры, влияющие на качество звука в экспортируемом фильме. Более подробно о частоте выборки, битовой глубине и других характеристиках звука речь шла в главе 19. При выборе пункта Disable (Блокировать) звуковые компоненты фильма экспортироваться не будут.

Если вы готовите фильм к редактированию с помощью другого приложения, лучше всего экспортировать звук с самым высоким качеством. При необходимости в дальнейшем всегда можно изменить параметры звука.

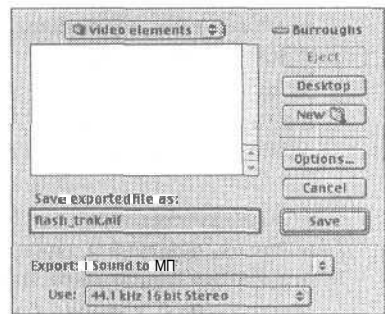
3. После установки всех параметров щелкните на кнопке ОК. Flash начнет процесс экспорта, который иногда занимает несколько минут, особенно если экспортируется анимация большого размера или сложные звуковые компоненты, а также используется сжатие. Конечный файл формата QuickTime будет помещен в указанный вами каталог, и вы увидите характерный значок с буквой Q, а также расширение MOV после имени файла. В зависимости от установленных параметров фильм в таком формате можно распространять на компакт-дисках и в Интернете, а также использовать как основу для других видеопроектов.

## Видео со звуком

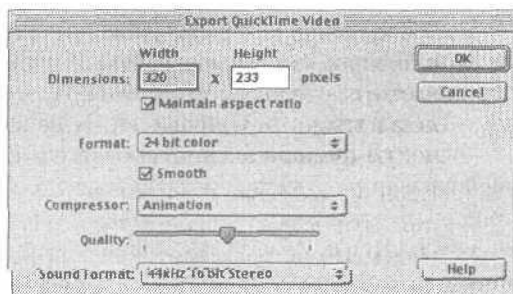
О том, как перенести видеофильм на видеокассету с использованием приложения Adobe Premiere, будет рассказано в разделе «Подготовка видео для переноса на видеоленту». Но если ваш фильм содержит звук, необходимо выполнить дополнительный этап подготовки. Мы уже говорили, что звук, экспортируемый из Flash-фильма в файл QuickTime, не будет работать в среде Adobe Premiere, поскольку при переносе он будет поврежден. Во избежание этого необходимо, прежде всего, используя приложение QuickTime, извлечь звук из видеофайла, а потом перенести его в среду Adobe Premiere отдельно от видео. Затем нужно повторно соединить видео со звуком. Это, конечно, не очень удобный, но пока единственный способ сохранения качества звука.

Чтобы извлечь звуковую дорожку из фильма QuickTime, выполните следующие действия.

1. С помощью приложения QuickTime Pro откройте видеофайл. Если на вашем компьютере отсутствует данное приложение, рекомендуем купить регистрационный ключ, с помощью которого можно превратить стандартную бесплатную версию QuickTime в QuickTime Pro. Подробную информацию об этом можно найти на Web-узле компании Apple [www.apple.com/quicktime/buy](http://www.apple.com/quicktime/buy).
2. Выберите команду Export меню File или воспользуйтесь комбинацией клавиш Cmd+E. В открывшемся диалоговом окне произведите следующие действия:
  - выберите имя файла и каталог для экспортируемой звуковой дорожки;
  - используя раскрывающийся список Export, выберите пункт Sound to AIFF (Экспортировать звук в формат AIFF) (Audio



2. После щелчка на кнопке Save откроется диалоговое окно Export QuickTime Video (Экспортирование в формат QuickTime Video). В данном окне можно установить следующие параметры экспортируемого видео.



- **Dimensions (Размеры).** Данные поля предназначены для установки значений Width (Ширина) и Height (Высота) конечного видео. За исключением случаев, когда Flash-фильм содержит растровые изображения, размер фильма может быть изменен без внесения искажений. При установке флажка Maintain Aspect Ratio (Сохранить формат изображения) пропорции фильма, определенные в диалоговом окне Document Properties (Свойства документа), будут сохранены.
- **Format (Формат).** С помощью этого списка можно установить следующие параметры глубины цвета для вашего фильма:

  - Black and White (Черно-белый) — очень низкое качество;
  - 4-bit color (Глубина цвета 4 бита) — низкое качество;
  - 8-bit color (Глубина цвета 8 битов) — 256 цветов (недостаточное качество);
  - 16-bit color (Глубина цвета 16 битов) — соответствует установке Thousands of Colors на компьютере Macintosh;
  - 24-bit color (Глубина цвета 24 бита) — соответствует установке Millions of Colors на компьютере Macintosh;
  - 32-bit color (Глубина цвета 32 бита) — то же, что 24 бита плюс дополнительная поддержка данных Alpha-канала для эффектов прозрачности.
- **Smooth (Сглаживание).** Установка этого флажка позволяет применить сглаживание при экспортировании фильма QuickTime. Использование функции сглаживания улучшает качество фильма, но могут появиться некоторые искажения. Если искажения значительные, флажок Smooth необходимо снять.
- **Compressor (Система сжатия).** С помощью раскрывающегося списка Compressor можно выбрать один из кодеков сжатия видео, доступных для QuickTime. Для Flash-фильмов, содержащих только векторную графику, наилучший результат можно получить, выбрав в списке пункт Animation. Кодеки Cinepak и Sorenson лучше подходят для фильмов, которые будут распространяться на компакт-дисках. Более подробную информацию о кодеках формата QuickTime можно найти на Web-узле QuickTime [www.apple.com/quicktime/specifications.html](http://www.apple.com/quicktime/specifications.html).
- **Quality (Качество).** С помощью регулятора Quality можно установить степень сжатия видео. При установке в крайнее правое положение сжатие будет минимальным при наилучшем качестве. Если переместить регулятор к левому краю, степень сжатия увеличится за счет потери качества.



Но если вы планируете дальнейшее редактирование видеофайла с помощью других приложений (например, Adobe Premiere), то необходимо выполнить некоторые дополнительные действия для подготовки используемых звуков. Поточковый звук, вложенный в видеофайл Flash, не работает в среде Adobe Premiere, любые звуковые компоненты, включенные во Flash-фильм, будут повреждены после распознавания. Для предотвращения таких повреждений звуковые и видеокомпоненты необходимо переносить в среду Adobe Premiere *раздельно*. При работе на компьютерах Macintosh для извлечения звуковой дорожки из видеофайла можно использовать приложение QuickTime Pro. В среде Windows Flash позволяет экспортировать отдельно и анимацию, и звук. Более подробно об этом будет рассказано в разделах «QuickTime Video» и «Видео для Windows» данной главы.

## Экспорт Flash-фильма в видео

После завершения предварительных настроек и установки необходимых параметров все готово для следующего этапа создания видео: экспорта фильма в видеоформат. При работе на компьютере Macintosh вы можете экспортировать анимацию в файл формата QuickTime, а при работе в среде Windows — в файл формата AVI. В обоих случаях анимация на основе векторной графики будет преобразована в растровый видеофайл. Этим файлом вы сможете манипулировать в среде приложения для правки видео.

### QuickTime Video

При работе на компьютере Macintosh Flash позволяет экспортировать анимацию в файл формата QuickTime. Необходимо знать основные отличия между файлами QuickTime Flash и QuickTime Video формата QuickTime. Одной из новых возможностей версии QuickTime 4 стала поддержка файлов Flash. Теперь фильм Flash может быть экспортирован в файл формата *QuickTime Flash*, позволяющий сохранить все интерактивные свойства фильма и действующий как обычный Flash-файл. Подробно о публикации интерактивных фильмов в формате QuickTime Flash рассказано в главе 27.

Из всех возможных форматов QuickTime Video наименее похож на формат Flash. Данный формат целиком основан на использовании растровой графики и не поддерживает интерактивные свойства, но в то же время он является наиболее простым форматом для преобразования Flash-фильма в видео.

Если вы правильно выполнили все инструкции по предварительной подготовке, то ваш фильм готов для экспортирования. Для экспорта Flash-фильма в файл формата QuickTime Video выполните следующие действия.

1. Выберите команду Export Movie меню File. В открывшемся окне Export Movie выберите каталог, в котором хотите сохранить файл. В поле списка Save as type (Сохранить в формате) укажите формат QuickTime Video и введите имя файла в поле Name. Обратите внимание, что Flash автоматически добавляет к имени расширение MOV, соответствующее формату QuickTime Video.

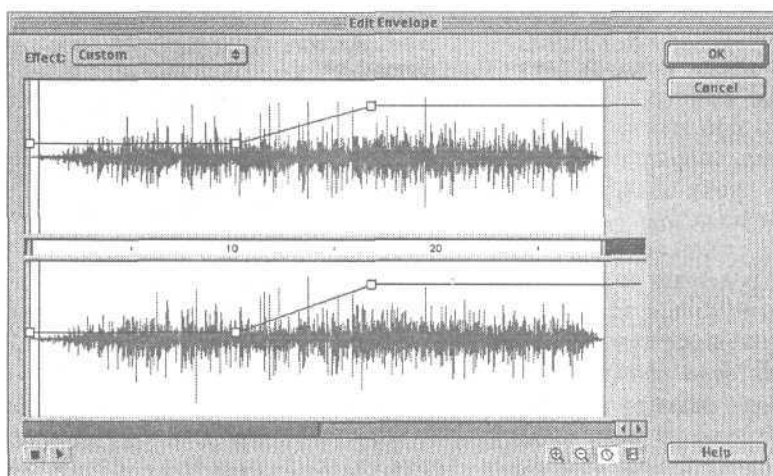


Рис. 28.6. Диалоговое окно Edit Envelope

- **Synchronization (Синхронизация).** Перенос видеоданных в среду Flash и синхронизация видео и звука являются очень важным процессом. Если синхронизация нарушена, звук не соответствует действиям, что создает эффект плохо дублированного кино. При подготовке к экспортированию вне зависимости от параметров синхронизации, установленных в исходном Flash-фильме, для всех звуковых файлов на панели Properties должен быть выбран элемент Stream (Потоковая) списка Sync (Синхронизация). Все звуки в этом случае будут связаны с частотой смены кадров, а это обеспечит точную синхронизацию сигналов. Убедитесь также, что каждому звуку выделено достаточно кадров для полного воспроизведения. Потоковые звуки могут воспроизводиться только на протяжении времени воспроизведения кадров, в которых они содержатся. Более подробную информацию о синхронизации звуков во Flash-фильмах вы найдете в главе 21.

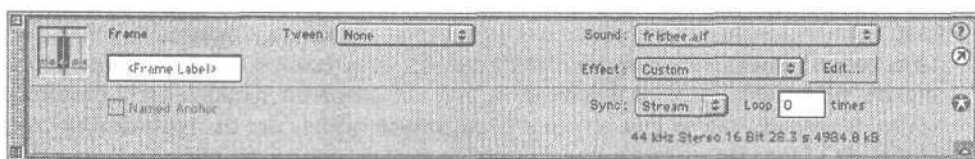


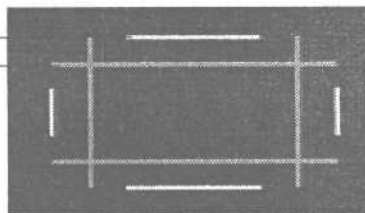
Рис. 28.7. Параметры звука на панели Properties

## Вложенный звук

Экспорт звука из Flash-фильма имеет свои достоинства и недостатки. Одним из преимуществ является сохранение синхронизации, что освобождает от необходимости реконструировать звук в видеофайле. Еще одним плюсом является то, что создается полностью готовый видеофайл, включающий все звуковые и видеокомпоненты.

В дополнение к информации, приведенной выше, следует заметить, что не все мониторы воспроизводят видео одинаково. На некоторых из них информация, находящаяся на краях области действия, может располагаться за рамками

Безопасная зона действий  
Безопасная зона титров



экрана и, следовательно, теряться. Поэтому необходимо убедиться, что ваш фильм не содержит важных деталей на краях области действия. Существуют две «безопасные зоны», зная о которых, вы сможете избежать проблем при использовании нестандартных мониторов: *безопасная зона действий* (action-safe zone) и *безопасная зона титров* (title-safe zone). Безопасная зона действий занимает 90 % общего размера экрана. Для разрешения 640x480 все важные детали изображения должны быть размещены в пределах зоны 576x432. Безопасная зона титров предназначена для воспроизведения текстовой информации и занимает 80 % общего размера экрана, что для разрешения 640x480 соответствует размеру 512x384. Обе эти зоны расположены в центре области действия.

## Подготовка звуковых компонентов фильма Flash

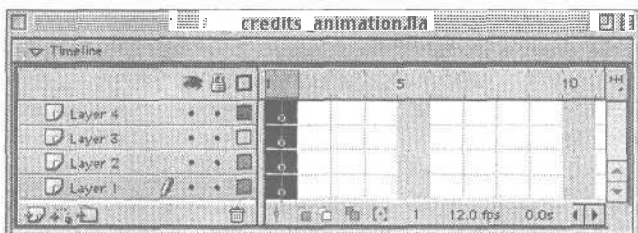
Flash можно использовать как среду создания анимации без звуковых компонентов. Если анимация предназначена для добавления к существующим видеокадрам и последующего совместного с ними редактирования, можно не создавать звуковые эффекты в самом Flash-фильме, а добавлять звук в конечный видеофайл. Но если вы планируете перенести анимацию Flash в видеофайл вместе со звуком, то при подготовке фильма выполните некоторые дополнительные действия.

При переносе Flash-фильма в видеоформат необходимо учитывать следующие особенности использования звука.

- **Mixing (Микширование).** При создании Flash-фильма диалоговое окно Edit Envelope (Редактирование огибающей сигнала) позволяет настраивать относительный уровень каждого звукового сигнала, используемого в фильме (рис. 28.6). Данные настройки очень важны, поскольку позволяют определить точное место каждого звукового сигнала в звуковом спектре. Одни типы звуков, такие как диалоги и комментарии, обязательно должны быть слышны четко, а другие, такие как музыка и звуковые эффекты, настраиваются так, чтобы не служить помехой для других звуковых компонентов.

Ваша задача состоит в настройке огибающей звукового сигнала таким образом, чтобы получить наилучший баланс уровней звуков, используемых в фильме. Прежде чем экспортировать анимацию, проверьте данные настройки и убедитесь, что уровни сигналов установлены правильно, поскольку после экспортирования звук редактировать невозможно. Если вам все же понадобится изменить уровни звуковых сигналов *после экспортирования*, то придется вернуться к исходному Flash-файлу, произвести настройку звуков и повторно экспортировать фильм. Более подробно о настройке звука с помощью диалогового окна Edit Envelope (Редактирование огибающей сигнала) рассказано в главе 20.

4. Вставьте новые слои для размещения кадров клипа. Например, если в клипе использовалось четыре слоя, вставьте четыре новых слоя. Теперь все готово для вставки кадров клипа на главную временную шкалу.
5. Для выбора первых ключевых кадров созданных слоев щелкните на первом ключевом кадре верхнего слоя, затем, нажав и удерживая клавишу Shift, щелкните на первом кадре нижнего слоя.



6. Чтобы вставить кадры клипа на созданные слои с выбранными первыми кадрами, выберите команду Paste Frames меню Edit. Ваша временная шкала должна выглядеть, как показано на рис. 28.5.

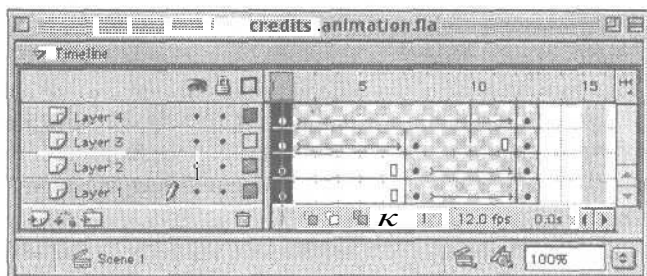


Рис. 28.5. Вставка кадров клипа на главную временную шкалу

Описанный выше процесс несложен, но в зависимости от особенностей использования клипа исходным Flash-фильмом вам могут понадобиться некоторые дополнительные настройки.

## Flash-фильм и параметры мониторов

Еще один нюанс, который необходимо учитывать при подготовке анимации для просмотра на телевизионном мониторе, — *характеристическое отношение экрана*, то есть отношение ширины изображения к его высоте. Характеристическое отношение 4:3 означает, что на каждые четыре единицы по горизонтали приходится три единицы по вертикали. Телевизионному и видеоформату соответствует разрешение 640x480 пикселей. Настроить характеристическое отношение 4:3 можно, изменив значения высоты и ширины в диалоговом окне Document Properties (Свойства документа). Чтобы открыть данное окно, выберите команду Document меню Modify. Если ваш фильм соответствует отношению 4:3, но имеет меньшие размеры области отображения кадров (например, 160x120), то окончательную настройку вы сможете сделать при экспорте видео.

к предыдущему кадру, необходимо удалить данное действие. Если действия являются совместной частью общей анимации, то есть требуется неоднократный повтор некоторых кадров, необходимо задать циклическое копирование кадров и их вставку на временную шкалу. Для этого следует выделить кадры, включенные в цикл, выбрать команду Copy Frames меню Edit, а затем команду Paste Frames в том же меню. Такое повторение кадров позволит восстановить сюжетную структуру фильма. Подробнее о манипулировании кадрами на временной шкале рассказано в главе 9.

Если Flash-фильм содержит несколько сцен, анимация будет экспортирована в том порядке, в котором сцены размещены на панели Scene. Данную панель можно открыть с помощью команды Scene меню Window либо воспользоваться комбинацией клавиш Shift+F2.

Кроме того, некоторые преобразования потребуются, если в вашей анимации будут использоваться клипы. В качестве самостоятельных элементов анимации клипы прекрасно работают во Flash-фильмах и в формате SWF, и в формате QuickTime, но они не поддерживаются в видеоприложениях, где данные представлены последовательно. Поэтому клип, использующийся в фильме, должен быть перемещен на его собственный слой главной временной шкалы. Используя технику копирования и вставки кадров, описанную ранее, можно перенести кадры с временной шкалы клипа на главную временную шкалу.

Для переноса кадров клипа на главную временную шкалу выполните следующие действия.

1. Включите режим редактирования символа, для чего используйте раскрывающееся меню Edit Symbols или выберите символ клипа в библиотеке и выполните на нем двойной щелчок. В этом режиме вы получите доступ к временной шкале клипа.
2. Нажав и удерживая клавишу Shift, выполните щелчок на имени каждого слоя, содержащего кадры, которые составляют анимацию клипа. Запомните количество используемых в клипе слоев.
3. Для копирования кадров в буфер обмена выберите команду Copy Frames меню Edit (рис. 28.4). Возвратитесь в режим редактирования фильма.

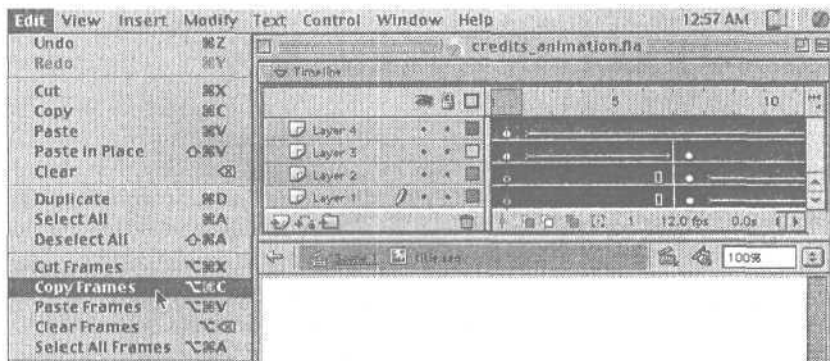


Рис. 28.4. С помощью команды Copy Frames меню Edit можно скопировать кадры клипа в буфер обмена данными

## Публикация Flash-фильмов на видеокассетах

Вы знаете, какой мощью и гибкостью обладают Flash-фильмы при воспроизведении на компьютере, и можете задаться вопросом, зачем переносить их на видеоленту. Ведь на видеокассетах можно просматривать только неинтерактивные фильмы, в которых кадры воспроизводятся последовательно. Но разработчики Flash учли возможность создания фильмов в различных форматах, требующихся потенциальной аудитории. Зрителям же предоставляется возможность самим решать, в каком виде им удобнее просматривать Flash-фильм: на видеоленте или на компьютере.

В определенных ситуациях Flash-фильмы, представленные на видеокассетах, обладают рядом достоинств. Например, на кассету может быть перенесен Flash-фильм, не содержащий интерактивных элементов и служащий только для передачи информации. Кроме того, такой формат больше всего подходит, когда необходимо создать заставку или телевизионный мультфильм. С помощью Flash можно разрабатывать и переносить на видеокассеты фильмы для презентаций выставок и концертов. Возможности данного приложения также могут использоваться при составлении телевизионных программ, например для создания титров. Комбинируя Flash-анимацию и реальные видеокадры можно получить интересные многоуровневые эффекты. Таким образом, благодаря Flash ваши дизайнерские возможности поистине неограниченны.

Далее в этой главе мы рассмотрим вопросы, касающиеся преобразования Flash-фильмов в видеоформат. В зависимости от того, какой формат вы хотите получить, необходима модификация фильма, обеспечивающая получение наилучшего результата. Процесс подготовки включает следующие шаги:

- ▶ подготовка графических объектов, включая клипы;
- ▶ подготовка звука;
- ▶ экспортирование Flash-фильма в формат QuickTime (Macintosh) или AVI (Windows);
- ▶ импорт анимации в приложение для редактирования видео и настройка характеристик в соответствии со специфическими задачами.

Но это только общий план подготовки, специфика каждого этапа будет рассмотрена в следующих разделах.

### Подготовка временной шкалы Flash

Первый этап преобразования Flash-анимации в видео включает подготовку временной шкалы вашего фильма. В данном случае подготовка означает ее упрощение таким образом, чтобы фильм не содержал действий, клипов, навигационных кнопок и других элементов, наличие которых несовместимо с созданием неинтерактивного фильма.

Действия должны быть попросту удалены, поскольку они могут прервать последовательный процесс воспроизведения анимации. Например, если фильм содержит какой-либо цикл `gotoAndPlay()`, который возвращает головку воспроизведения

изменения скорости воспроизведения клипа в ускоренном режиме это число может быть увеличено или уменьшено.

```
trailer.onLoad - function(){
 ffwd - 0;
}
trailer.onEnterFrame - function(){
 if(ffwd==1){
 frame - trailer._currentframe
 trailer.gotoAndPlay(frame + 4):
 }
}
```



Используя действия и методы объекта `MovieClip`, можно создавать самые разнообразные элементы управления воспроизведением клипа. В каталоге Chapter 28 на прилагаемом компакт-диске вы найдете файлы `video2MX.fla` и `video2MX.swf`, содержащие примеры создания элементов управления.

## Управление видеофайлами в фильме Flash

В данном разделе вы познакомитесь с некоторыми техниками, помогающими управлять как вложенным во Flash-фильм, так и внешним видеофайлами. Например, после редактирования исходного вложенного видео с помощью внешних программ его можно не импортировать повторно, а обновить.

Для обновления вложенного видео выполните следующие действия.

1. Откройте окно библиотеки и выберите видеофайл.
2. Щелкните правой кнопкой мыши (Windows) или выполните щелчок при нажатой клавише `Ctrl` (Macintosh) и выберите пункт `Properties` в раскрывшемся контекстном меню.
3. В появившемся диалоговом окне `Embedded Video Properties` (Свойства вложенного видео) щелкните на кнопке `Update` (Обновить).

Кроме того, можно заменить вложенные видеофайлы совершенно другими. Для этого необходимо щелкнуть на кнопке `Import` и в открывшемся диалоговом окне `Open` выбрать новый файл, содержащий видео. Выбранный файл заменит вложенный, хранящийся в библиотеке.

Если в библиотеке хранятся несколько различных видеоклипов, для экземпляра видеоклипа на рабочей области можно заменить ассоциированный с ним родительский видеоклип. Чтобы проделать данную операцию, выполните следующие действия.

1. Выберите на области действия экземпляр клипа, который вы хотите заменить.
2. Откройте панель `Properties`, воспользовавшись комбинацией клавиш `Cmd/Ctrl+F3`.
3. Щелкните на кнопке `Swap` и выберите клип, который будет родительским для экземпляра видеоклипа на области действия.
4. Щелкните на кнопке `OK`. Экземпляр клипа будет обновлен.

При публикации Flash-фильма в формате `SWF` звук любого вложенного видеоклипа будет экспортирован с использованием параметров потокового звука. Параметры потокового звука устанавливаются на вкладке `Flash` диалогового окна `Publish Settings`. (Более подробно о параметрах сжатия звука рассказано в главе 21.)

```

 trailer.gotoAndPlay(frame + 1);
}

```

Объясним функцию данного сценария. Когда кнопка нажата, головка воспроизведения сохраняет свое положение в клипе `trailer`, а воспроизведение начинается со *следующего* кадра. При переходе головки воспроизведения к кадру 1, воспроизведение останавливается. Поскольку другие кадры не содержат действий для остановки клипа, то при перемещении головки к кадру 2 воспроизведение будет проходить без остановки. Кроме того, если воспроизведение клипа остановлено на любом другом кадре кроме первого, его всегда можно продолжить со следующего кадра.

8. Выберите кнопку Stop, откройте панель Actions, выделите действие `stop()` и введите следующие строки-операторы:

```

on(press){
 trailer.gotoAndStop(1);
}

```

Данный код также достаточно прост. При нажатии кнопки головка воспроизведения клипа перемещается на временной шкале к кадру 1. Поскольку первый кадр содержит действие `stop()`, воспроизведение клипа будет остановлено на кадре 1.

9. Чтобы проверить действие кнопок, выберите команду Test Movie меню Control или воспользуйтесь комбинацией клавиш `Cmd+Return` (Macintosh) либо `Ctrl+Enter` (Windows). Обратите внимание, что кнопка, содержащая действие `playO`, инициирует начало воспроизведения клипа, а кнопка с действием `stopO` останавливает его и перемещает головку воспроизведения к кадру 1.

При желании вы можете добавить дополнительные кнопки для управления воспроизведением клипа. Ниже приведены некоторые примеры.

1. Кнопка Pause служит для остановки воспроизведения фильма без изменения положения головки, что позволяет продолжить просмотр клипа с этого же кадра.

```

on(press){
 trailer.stop();
}

```

2. Кнопка Fast Forward позволяет ускорить воспроизведение клипа. В зависимости от состояния кнопки значение переменной `ffwd` будет либо установлено (1), либо сброшено (0).

```

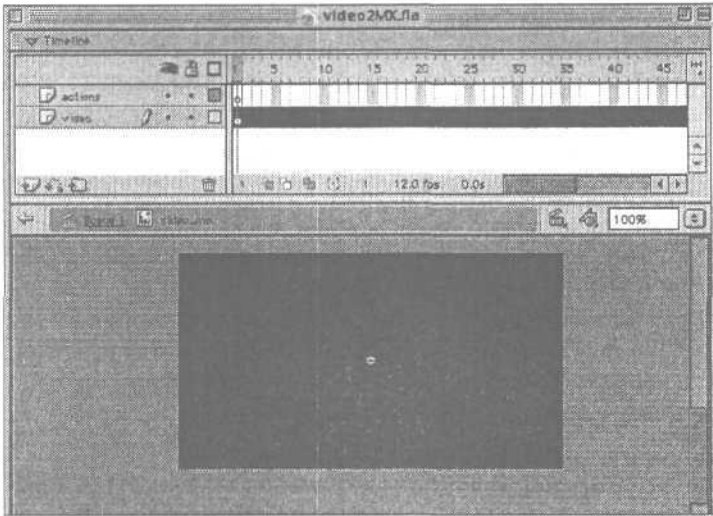
on(press){
 ffwd = 1;
}
on(release){
 ffwd = 0;
}

```

Для того чтобы экземпляр видеоклипа `trailer` реагировал на изменение значения переменной, ему должна быть присвоена функция. Приведенный ниже сценарий нужно добавить к кадру 1 главной временной шкалы. Когда вы установите значение переменной, будет воспроизводиться каждый четвертый кадр. Для



что временная шкала содержит недостаточно кадров для размещения видео, и спросит, нужно ли добавить кадры. Для добавления кадров щелкните на кнопке Yes. После выполнения данных действий первый кадр видео должен появиться на области действия, а видеокadres разместиться на временной шкале (рис. 28.3).



**Рис. 28.3.** Добавленные кадры для размещения видео

4. Чтобы выровнять видео на области действия клипа по горизонтали и вертикали, используйте панель Align, открыть которую можно с помощью комбинации клавиш `Cmd/Ctrl+K`. После завершения выравнивания можно добавить действия ActionScript для управления воспроизведением.
5. Выберите ключевой кадр слоя `actions` и откройте панель Actions. Введите действие `stop()` и закройте панель. Если это действие помещено в первый ключевой кадр временной шкалы, то воспроизведение клипа может начаться только после выполнения команды `Play`.
6. Сохраните файл и перейдите к главной временной шкале. С помощью мыши перетащите символ `video_mc` из окна библиотеки на область действия, создав таким образом экземпляр символа, и с помощью панели Properties присвойте ему любое имя (например, `trailer`). В данном примере мы будем использовать имя `trailer`.
7. Теперь, когда клип имеет уникальное имя, можно добавить кнопки, которые позволят управлять воспроизведением клипа в интерактивном режиме. Переместите две кнопки из окна библиотеки на область действия: одну для команды `Play` (Воспроизведение), вторую для команды `Stop` (Остановка). Выберите кнопку `Play` на области действия, откройте панель Actions, выделите действие `playO` и введите следующие строки-операторы:

```
on(press){
 frame - trailer._currentframe;
```

Flash, и внешнее видео как отдельную дорожку QuickTime. (Более подробно о публикации фильмов Flash в формате QuickTime рассказывается в главе 27.)

Чтобы импортировать видео как связанный видеофайл, выполните следующие действия.

1. Импортируйте видео, как описано в разделе «Импорт видео». В открывшемся диалоговом окне Import Video установите переключатель Import Options в положение Link to external video file и щелкните на кнопке ОК.
2. Если вы выбрали параметр импортирования видео непосредственно на область действия, Flash выдаст предупреждающее сообщение с вопросом, желаете ли вы увеличить количество кадров на главной временной шкале для размещения видеокладов. Выполните одно из действий:
  - > если хотите увеличить количество кадров для размещения видео, щелкните на кнопке Yes;
  - > чтобы не изменять количество кадров, щелкните на кнопке No. При необходимости вы сможете добавить кадры позже.

Импортированный видеофайл появится в окне библиотеки.

## Использование сценариев ActionScript для управления воспроизведением видео

После помещения видео во Flash-фильм у вас появится возможность управлять его воспроизведением (то есть начинать воспроизведение, останавливать, делать паузу, осуществлять перемотку и т. д.). В среде Flash импортированное видео рассматривается как обычный элемент фильма.

Наиболее эффективным способом управления видео является создание видеоклипа (символа типа Movie Clip). Такой клип похож на обычный клип, но на его временной шкале размещаются видеоклады. Кроме того, видеоклип, как и обычным клипом, можно управлять с помощью сценариев ActionScript.



Для того чтобы вы могли создавать видеоклипы, ваша библиотека должна содержать импортированное видео. Если у вас нет собственных видеофайлов, можете использовать те, которые хранятся на прилагаемом компакт-диске в каталоге Chapter 28. Данный каталог содержит видеофайл mxVideo.mov в формате QuickTime и видеофайл mxVideo.flv в формате Macromedia Flash. Видеофайлы могут быть как вложены во Flash-фильм, так и связаны с ним. В данном каталоге также находятся некоторые кнопки управления воспроизведением видео.

Для создания видеоклипа выполните следующие действия.

1. Выберите команду New Symbol меню Insert или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+F8`. Присвойте новому символу имя `video_mc`, установите переключатель типа символа в положение Movie Clip и щелкните на кнопке ОК.
2. Flash переключится в режим редактирования символа. На временной шкале клипа присвойте первому слою имя `video`. Затем вставьте новый слой и назовите его `actions`.
3. Выберите слой `video` и с помощью мыши перетащите символ видео из окна библиотеки на область действия. Flash, вероятно, выдаст предупреждение,

файла уменьшается, а воспроизведение видеок кадров происходит более плавно. Если требуется, чтобы видеок кадры во Flash-фильме были уменьшены в размере, установите соответствующее значение Scale (Масштаб). Это позволит уменьшить размер файла и увеличить скорость воспроизведения видео после его сжатия.

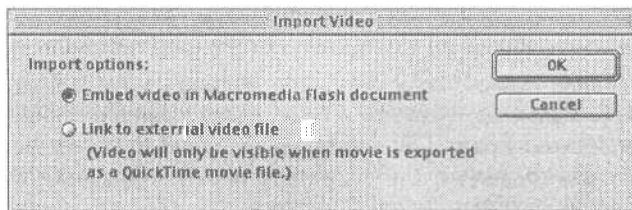
5. При установке флажка Synchronize video to Macromedia Flash document frame rate (Синхронизировать видео с Flash-фильмом) скорость воспроизведения видео будет согласована со скоростью воспроизведения Flash-кадров на временной шкале. Если вы хотите, чтобы воспроизведение видео и Flash-кадров происходило независимо, снимите данный флажок.
6. Установите значение в поле Number of video frames to encode per number of Macromedia Flash frames (Отношение числа кодируемых видеок кадров к числу Flash-кадров). Этим значением определяется отношение числа импортируемых видеок кадров к числу кадров главной временной шкалы Flash. Например, чтобы продолжительность воспроизведения одного видеок кадра была равна продолжительности воспроизведения на временной шкале одного Flash-кадра, установите это отношение равным 1:1. Чтобы продолжительность воспроизведения одного видеок кадра была равна продолжительности воспроизведения четырех Flash-кадров, установите это отношение равным 1:4 и т. д.
7. Чтобы импортировать звук видеофайла, установите флажок Import Audio (Импортировать звук). Если видеофайл не содержит звук либо звук не поддерживается Flash, будет выдано предупреждающее сообщение о невозможности импортирования звука.
8. После установки всех параметров, щелкните на кнопке ОК. Если вы выбрали параметр импортирования видео непосредственно на область действия, Flash выдаст предупреждающее сообщение с вопросом, желаете ли вы увеличить количество кадров на главной временной шкале для размещения видео. Выполните одно из действий:
  - ▷ если вы хотите увеличить количество кадров, щелкните на кнопке Yes;
  - ▷ чтобы не изменять количество кадров, щелкните на кнопке No. При необходимости вы сможете добавить кадры позже.

Импортированный видеофайл появится в окне библиотеки, слева от его имени будет находиться значок с изображением видеокамеры. Теперь вы можете интегрировать этот файл в свой фильм. Подробнее об этом будет рассказано в разделе «Использование сценариев ActionScript для управления воспроизведением видео».

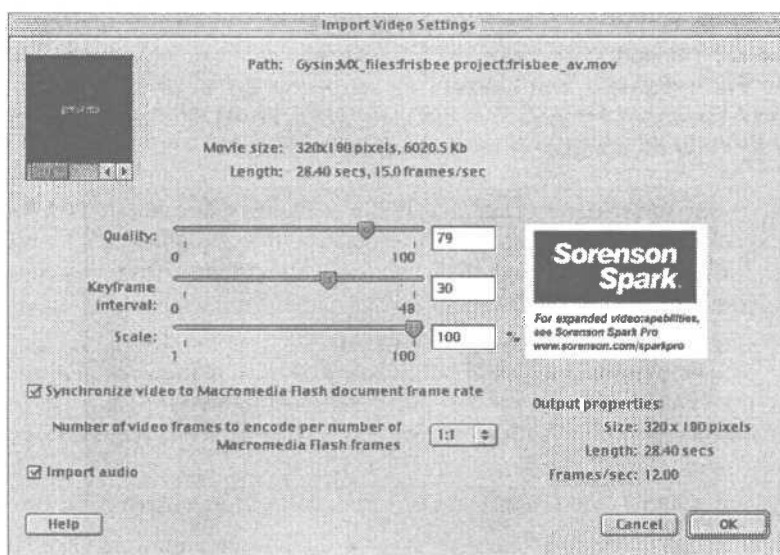
## Использование связанных видеофайлов

При импортировании видео во Flash (это справедливо только для файлов с расширением MOV) вы можете сделать его внешним, или *связанным*. То есть видеоданные не помещаются непосредственно во Flash-фильм как часть фильма, а Flash-фильм содержит ссылку на видеофайл, который хранится во внешнем файле. Flash-фильмы, имеющие ссылки на внешние видеофайлы, не могут публиковаться как SWF-файлы, поскольку этот формат не поддерживает связанное видео. Такие Flash-фильмы должны публиковаться как файлы QuickTime (QuickTime Flash). В результате фильм QuickTime сможет воспроизводить и дорожку

импортирования) в положение Embed video in Macromedia Flash document (Вложить видео в документ Flash) и щелкните на кнопке ОК.



В открывшемся окне Import Video Settings (Параметры импорта видео) вы получите доступ к различным параметрам, влияющим на степень и характер сжатия видео.



**Рис. 28.2.** Окно Import Video Settings позволяет установить параметры сжатия вкладываемого во Flash видео

- Используйте бегунок Quality (Качество) для установки степени сжатия видео. Чем выше данное значение, тем лучше качество видео, но и тем больше размер файла. При установке низкого значения параметра Quality размер конечного файла уменьшится, но за счет потери качества. При установке значения 60 и выше сохраняется хороший уровень качества видео.
- С помощью бегунка Keyframe interval (Интервал между ключевыми кадрами) можно установить частоту вставки ключевых кадров. Обычно необходимость в использовании ключевых кадров невелика, поскольку кодек Spark самостоятельно вставляет их там, где это требуется.
- Для настройки размера (в процентах) импортируемых видеокладов используйте бегунок Scale (Масштаб). При установке меньшего значения, размер

Более подробную информацию о применении кодека Sorenson Spark вы найдете в разделе «About the Sorenson Spark Codec» («О кодеке Spark Codec») файлов справки Flash MX.

Чтобы импортировать видео во Flash, выполните следующие действия.

1. Выберите команду Import меню File либо воспользуйтесь комбинацией клавиш **Cmd/Ctrl+R**. При этом импортируемое видео помещается непосредственно на область действия. Если выбрать команду Import to Library меню File, видеоданные будут размещены в библиотеке текущего фильма, откуда вы сами сможете поместить их на область действия.
2. В раскрывающемся списке Show (Показывать) диалогового окна Import выберите пункт All Video Formats (Все видеоформаты) (рис. 28.1). Данное меню действует как фильтр, позволяя выводить файлы только тех форматов, которые доступны для импорта во Flash. Выберите требуемый файл и щелкните на кнопке Open.
3. Далее в окне Import Video укажите, что следует сделать: поместить видеофайл в файл Flash-фильма или создать на него ссылку и рассматривать данный файл как внешний (только для файлов QuickTime). Установите переключатель в нужное положение и щелкните на кнопке OK.

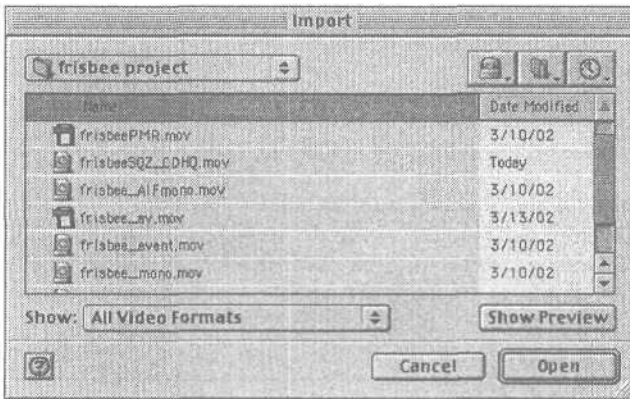


Рис. 28.1. Окно Import с видеофайлами, доступными для импорта во Flash

В следующем разделе будут рассмотрены вопросы работы с видео при выборе различных способов импортирования.

## Использование вложенных видеофайлов

При импортировании видео во Flash видеоданные могут быть вложены во Flash-фильм. При этом видео станет постоянной частью Flash-фильма и будет храниться в библиотеке вместе с другими символами, такими как клипы, кнопки, графические символы и т. д.

Чтобы импортированный видеофайл был вложен во Flash-фильм, выполните следующие действия.

1. Сначала импортируйте видео, как описано в предыдущем разделе. В диалоговом окне Import Video установите переключатель Import Options (Параметры

VBR обычно меньше файлов FLY, поскольку они сжаты «разумно»: видеокадры, содержащие сложные изменения цвета или движения, имеют больший размер, чем простые кадры. Это позволяет применить сжатие на тех участках, где оно наиболее необходимо, и в результате конечный файл заметно уменьшается. Подробнее о приложении Sorenson Squeeze можно узнать, посетив Web-узел компании Sorenson по адресу [www.sorenson.com/sparkpro.html](http://www.sorenson.com/sparkpro.html).

#### ПРИМЕЧАНИЕ -

Приложение Sorenson Squeeze позволяет также сжимать файлы для проигрывателей QuickTime в формате MOV и Flash Player 6 в формате SWF.

## Импорт видеофайлов

Чтобы поместить видеоданные в среду Flash, сначала нужно их импортировать таким же образом, как мы это делали со звуком, растровыми изображениями и другими мультимедийными данными. Flash поддерживает широкий диапазон различных цифровых видеоформатов. Независимо от платформы, Macintosh или Windows, для импортирования большинства типов видеоданных в среду Flash MX необходимо установить приложение QuickTime 4 (или выше). Если вы работаете в среде Windows, то благодаря поддержке приложения DirectX 7 (или выше) можете импортировать видео некоторых дополнительных Windows-форматов.

В табл. 28.1 приведен перечень видеоформатов, импортирование которых поддерживается Flash MX.

**Таблица 28.1.** Форматы видеофайлов, поддерживаемые для импорта в среду Flash MX

| Тип файла                                   | Платформа          |
|---------------------------------------------|--------------------|
| Audio Video Interleaved (.avi)*             | Macintosh, Windows |
| Digital Video (.dv)                         | Macintosh, Windows |
| Moving Picture Experts Group (.mpg, .mpeg)* | Macintosh, Windows |
| QuickTime Movie (.mov)                      | Macintosh, Windows |
| Windows Media File (.wmv, .asf)*            | Windows            |
| Macromedia Flash Video (.flv)**             | Macintosh, Windows |

\* Эти форматы поддерживаются в среде Windows только при установленном приложении DirectX 7 (или выше).

\*\* Файлы Macromedia Flash Video (.flv) могут быть импортированы в среду Flash без помощи приложений QuickTime или DirectX 7, но при этом они должны быть сжаты с помощью кодека Sorenson Squeeze.

Если какой-либо формат не может быть импортирован, Flash выдаст предупреждающее сообщение. Иногда вам потребуется импортировать видео без звука, в этом случае при установке параметров экспорта в окне Import Video Settings (Параметры импорта видео) необходимо снять флажок Import audio (Импортировать звук).

Даже при использовании кодека Sorenson Spark и современных технологий необходимо применять как можно более «дружественное» сжатие видео. Это означает оптимизацию видео- и аудиосодержимого и применение алгоритмов сжатия, соответствующих возможностям оборудования потенциальных зрителей.

## Видео в среде Flash

**MX** Одним из новых свойств Flash MX является поддержка цифрового видео. При работе с более ранними версиями программы разработчику приходилось имитировать видео. Обычно это осуществлялось посредством помещения последовательности статических изображений в клип. С функциональной точки зрения такая техника работала корректно, однако размер конечного файла резко возрастал, поскольку отсутствовала возможность применить для видеоданных сжатие.

Но все эти трудности в прошлом. С точки зрения функциональных возможностей приложения поддержка цифрового видео во Flash MX — это значительный шаг вперед. Если раньше приложение Flash было только инструментом создания анимации для Web, то сейчас оно становится одним из лучших средств работы с видеоплатформами. Теперь любой пользователь, на компьютере которого установлен подключаемый модуль Flash 6, может просматривать как обычные Flash-фильмы, так и фильмы, содержащие материалы реального видео. И все это достигается благодаря использованию одного универсального подключаемого модуля.

### Кодек Soreson Spark

Невидимый пользователю механизм, предназначенный для работы с видео в среде Flash, -- кодек Soreson Spark, разработанный компанией Sorensen Media ([www.sorensen.com](http://www.sorensen.com)). Он выполняет действия компрессии/декомпрессии видеоданных и фактически является частью Flash MX. Кодек Spark способен, не слишком увеличивая размер конечного файла, помещать видеоданные во Flash-фильм. При этом он автоматически сжимает эти данные. Любой видеофайл, который импортируется и вкладывается в документ Flash, должен быть сжат кодеком.

Кодек Spark применяет для видео *временное* сжатие. При таком типе сжатия в видеофайле определяются области, в которых происходят значительные изменения, а каждый кадр кодируется с учетом величины изменений от предыдущего кадра к последующему. Если кадр не содержит таких изменений, в него копируется содержимое предыдущего кадра. Такой тип кодирования иногда называют *межкадровым кодированием*. Если изменения между кадрами значительные, то кодек Soreson Spark вставляет в видеофайл ключевой кадр. Он содержит *полное* изображение кадра и так же, как ключевой кадр Flash-фильма, обозначает «точку» фильма, в которой происходят значительные изменения. Таким образом, для обеспечения во Flash-фильме высокого качества видео кодек Soreson Spark использует комбинацию ключевых кадров и межкадровое кодирование. Эта техника позволяет создавать видеофайлы небольших размеров.

Версия кодека Spark, включенная в состав приложения Flash MX, называется Sorensen Spark Standard Edition (стандартная), но при желании вы можете установить более совершенную версию — Sorensen Spark Pro (профессиональную). Кодек Sorensen Spark Pro входит в состав приложения Sorensen Squeeze, позволяющего сжимать видеофайлы в формат Macromedia Flash Video (FLY). Приложение Squeeze и кодек Spark Pro имеют ряд существенных преимуществ по сравнению со стандартным кодеком, включенным в состав Flash, поскольку позволяют сжимать видеоданные с использованием метода Variable Bit Rate (VBR). Файлы

## Глава 28

# Работа Flash с другими видеоформатами

В последние годы наметилась тенденция к созданию универсальных приложений. Пользователи ждут от новых программ не только выполнения определенных операций, но и корректной работы совместно с другими приложениями. Это повышает эффективность использования программ, а также расширяет их функциональные возможности.

В этом смысле Flash не является исключением. Фактически благодаря популярности Flash и широкому принятию файлового формата SWF данное приложение является сегодня одним из самых переносимых. Возможности программы Flash позволяют не только включать видео- и звуковые эффекты других стандартов :з анимационные или интерактивные фильмы, но также создавать файлы, которые можно использовать в других приложениях.

В данной главе речь пойдет о возможностях Flash при работе с видео. Фильмы Flash могут экспортироваться как видеофайлы, а приложение Flash MX позволяет импортировать видеofilмы и управлять ими как любыми другими элементами Flash-фильма. Видео и Flash-форматы в основе своей различны. В видео не применяется векторная графика, автоматическое построение промежуточных кадров, инструменты рисования и отсутствует интерактивность. Однако фильмы в этом формате имеют свои преимущества. В отличие от Flash-фильма, который похож на мультипликацию, видеofilмы способны демонстрировать реалистичные и живые изображения. Кроме того, видео можно передавать по телевизионным каналам и использовать как часть кинофильмов.

Поскольку развитие Flash продолжается, весьма вероятно, что новые возможности этого приложения не только укрепят его лидирующую позицию среди программ по разработке анимации и интерактивных фильмов для Web, но и позволят Flash стать более важной частью всей медиасреды.

В данной главе будут рассмотрены следующие темы:

- ▶ импорт видеофайлов в среду Flash;
- ▶ кодек Sorenson Spark;
- ▶ использование сценариев ActionScript для управления воспроизведением видео;
- ▶ подготовка компонентов Flash-анимации для видео;
- ▶ работа с видео Flash;
- ▶ звук во Flash;
- ▶ перенос Flash-анимации в приложение, работающее с видео;
- ▶ повторная компоновка анимации в приложении Adobe Premiere;
- ▶ подготовка анимации для переноса на видеоленту.



4. Выберите кадр на временной шкале, содержащий обрамляющую фигуру. На панели Action добавьте действие print O с требуемыми установками.
5. Если панель Properties недоступна, выполните команду Window ▶ Properties.
6. Для того чтобы обозначить фигуру как печатаемую область, укажите на панели Properties в поле Frame Label метку #b.

## Разработки

Вы изучили почти все вопросы, связанные с публикацией фильмов, и уже сейчас можете распространять их в различных форматах. За пределами нашего внимания остались только игровые приставки, которые также могут использоваться для распространения Flash-фильмов.

В качестве примера можно привести создание пользовательского интерфейса популярной игры Star Wars Starfighter компании LucasArt, предназначенной для игровых приставок PlayStation 2 и Xbox. При работе над этим проектом программисты компании, успешно создав интерфейс начала игры, столкнулись с проблемой написания функционального интерфейса для выхода из нее.

Компания LucasArt заинтересовалась разработками небольшой фирмы Secret Level из Сан-Франциско ([www.secretlevel.com](http://www.secretlevel.com)). Ее программисты занимались созданием набора инструментальных средств разработки программного обеспечения, названного Strobe, для интеграции содержимого фильмов Flash в игры PlayStation 2. Оценив возможности Flash, руководство компании LucasArt подписало контракт с компанией Orange Design ([www.orangedesign.com](http://www.orangedesign.com)) на разработку интерфейса выхода из игры, использующего технологию Flash, интегрированную с помощью приложения Strobe. В результате использования Flash был создан прекрасный продукт, демонстрирующий возможности публикации и распространения SWF-фильмов.

Подробную информацию об использовании Flash при создании игры Starfighter вы можете найти в статье по адресу [www.gamasutra.com/features/20010801/corry\\_01.htm](http://www.gamasutra.com/features/20010801/corry_01.htm). Для прочтения статьи вам необходимо бесплатно зарегистрироваться на Web-узле.

## Резюме

В данной главе вы познакомились с различными форматами, в которых может быть опубликован Flash-фильм, и рассмотрели параметры обработки Flash-фильма при создании файлов различных форматов. Затем вы научились осуществлять предварительный просмотр и публикацию фильма, а также узнали о различии между публикацией и экспортированием. В конце главы были рассмотрены вопросы, связанные с созданием фильмов, доступных для людей с физическими недостатками, и фильмов, содержимое которых можно распечатать.

Описанный в данном разделе процесс поможет вам установить такие характеристики фильма, при которых его содержимое будет выводиться на печать как векторная графика. Без установки данных параметров содержимое фильма Flash также можно распечатать, однако качество изображений будет низким — 72 dpi.

Существуют два метода настройки Flash-фильм для вывода его содержимого на печать. Во-первых, можно обозначить определенные кадры на временной шкале, которые будут выведены на печать. Все кадры, не обозначенные как печатаемые, выводиться на печать не будут. Во-вторых, можно обозначить области в пределах определенных кадров, которые будут выведены на печать. Все пространство за пределами определенных зон на печать выводиться не будет.

#### ПРИМЕЧАНИЕ

При использовании команды Print меню браузера процессом печати управлять нельзя.

Прежде чем приступить к процессу конфигурирования фильма, необходимо знать следующее.

- ▶ Все элементы, выбранные для печати, должны быть полностью загружены.
- ▶ Чтобы клип мог быть распечатан, он должен находиться на области действия или на рабочей области и ему должно быть присвоено имя в поле Instance name на панели Properties.
- ▶ Приложения Flash Player версий до 4.025 (Windows) и 4.020 (Macintosh) не поддерживают функцию вывода кадров на печать.

Теперь рассмотрим, каким образом должен быть сконфигурирован Flash-фильм, чтобы его содержимое можно было распечатать.

## Выбор кадров для вывода на печать

Чтобы отметить кадры, которые необходимо распечатать, выполните следующие действия.

1. Выберите на временной шкале кадр, который хотите вывести на печать.
2. Откройте панель Properties, выбрав команду Properties меню Window.
3. В поле Frame Label (Метка кадра) введите символ #p.
4. Для выбора дополнительных кадров повторите шаги 1 и 3.

## Определение зоны для вывода на печать

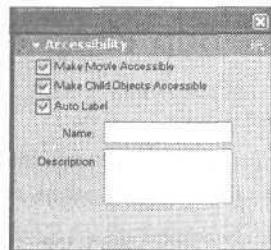
По умолчанию печатаемой областью является область действия, все находящиеся на ней объекты фильма будут выводиться на печать. Изображение любого объекта, выходящего за пределы области действия, не может быть распечатано. Кроме того, вы можете самостоятельно назначить определенную область как печатаемую. Для этого необходимо выполнить следующие действия.

1. Откройте Flash-фильм, содержащий кадры, которые необходимо распечатать.
2. Выберите кадр, который не отмечен как печатаемый с помощью метки #p.
3. На области действия создайте обрамляющую фигуру, ее размер соответствует печатаемой области.

3. Чтобы сделать доступным выбранный объект, установите флажок **Make Object Accessible**.
4. Если выбранный объект является клипом, установите флажок **Make Child Object Accessible**, что позволит сделать доступными все элементы в пределах клипа.
5. Введите имя в поле **Name**. Данное имя будет «прочитано» и озвучено программой считывания содержимого экрана. Учтите, что при создании динамического текста поле **Name** будет недоступным, а в качестве имени будет использоваться сам текст.
6. В поле **Description** введите описание выбранного объекта.
7. Если вы хотите связать комбинацию клавиш с отдельными символами кнопок, клипом или полями ввода текста, введите описание комбинации клавиш в поле **Shortcut** (Клавишная комбинация быстрого вызова). Существуют, однако, некоторые правила ввода описания клавиш, которым нужно следовать.
  - > Используйте сокращения названий клавиш, такие как **Shift**, **Ctrl** или **Alt**.
  - > При вводе букв алфавита используйте символы верхнего регистра.
  - ▷ Между названиями клавиш используйте знак **+** (без пробелов). Например, **Alt+5**, **Ctrl+N** или **Shift+9**.
8. После завершения установки всех параметров доступа к содержимому закройте панель **Accessibility**.

Описанный выше процесс хорошо подходит для того, чтобы управлять доступом к отдельным объектам фильма. Но если нужно сделать доступным для программы считывания содержимого экрана весь фильм, выполните следующие действия.

1. Убедитесь, что на области действия нет выделенных объектов.
2. Выберите команду **Accessibility** меню **Window**.
3. Установите флажок **Make Movie Accessible**. Если вы хотите, чтобы программа считывания содержимого экрана распознавала все доступные объекты в фильме, установите флажок **Make Child Object Accessible**.
4. Введите имя фильма в поле **Name**.
5. В поле **Description** введите описание фильма.
6. Установка флажка **Auto Label** (Автоматическая метка) позволяет использовать в качестве имен объектов текст, содержащийся на кнопке, или поля ввода текста.
7. После завершения закройте панель **Accessibility**.



## Создание фильмов, содержимое которых может быть выведено на печать

Вы уже убедились, что содержимое Flash-фильмов прекрасно выглядит на экране, поскольку изображения в нем создаются на базе векторной графики. Но Flash-фильмы обладают еще одним замечательным свойством — их изображения прекрасно выглядят при выводе на печать.

В последние годы наметилась тенденция к созданию Web-содержимого, доступного людям с различными физическими недостатками. До настоящего времени одной из основных проблем была невозможность интерпретации содержимого Flash-фильмов специальными программами, которые «считывают» информацию с экрана и «пересказывают» ее пользователю, имеющему плохое зрение. Сложность здесь заключается в том, что графическая информация в отличие от текстовой с трудом поддается точному истолкованию, поскольку разные люди могут описывать одно и то же изображение по-разному. В результате программы-интерпретаторы, являющиеся обычными компьютерными приложениями, не в состоянии правильно описать графические данные. Это приводит к тому, что люди с недостатками зрения, не имея точного описания содержимого экрана, могут получить неверную информацию о навигационных схемах, интерпретация которых зависит от правильного восприятия графического интерфейса (кнопок, меню и т. д.).

До появления версии Flash MX фильмы Flash были практически недоступны людям с недостатками зрения. Но теперь появилась возможность добавления информации к определенным элементам фильма, которая затем может быть считана программами-интерпретаторами. Этот способ дает возможность определить, как будут интерпретированы те или иные элементы фильма.

Но тут существуют некоторые условия, соблюдение которых необходимо. Во-первых, чтобы объект был доступен для программ-интерпретаторов, ему *должно* быть присвоено имя в поле Instance name (Имя экземпляра) на панели Properties. То есть доступными для программ-интерпретаторов могут быть только некоторые объекты: динамический текст, поля ввода текста, кнопки, клипы и фильмы целиком.

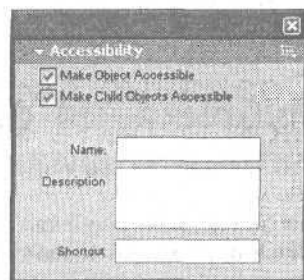
#### ПРИМЕЧАНИЕ-

Чтобы сделать доступным статический текст, вы должны преобразовать его в динамический.

Во-вторых, на компьютере пользователя должна быть установлена операционная система Windows с программным обеспечением для считывания содержимого экрана и подключаемым модулем Flash MX.

Принимая во внимание все вышесказанное, рассмотрим, как сделать содержимое фильма доступным для широкой аудитории пользователей, включая людей с физическими недостатками.

1. Прежде всего, выберите объект, который вы хотите сделать доступным для программы считывания экрана.
2. Откройте панель Accessibility (Доступность), выбрав команду Accessibility меню Window. Если выбранный вами объект нельзя сделать доступным, на панели Accessibility появится сообщение «Current selection cannot have accessibility applied to it» (Выделенный объект нельзя сделать доступным).



храниться в одном каталоге, иначе фильм QuickTime не сможет воспроизвести содержимое Flash-фильма.

## Предварительный просмотр Flash-фильма

После выполнения кропотливой работы по установке всех параметров обработки фильма для его публикации вы, конечно, захотите осуществить предварительный просмотр. Чтобы предварительно просмотреть фильм в любом из форматов, которые были выбраны на вкладке Formats диалогового окна Publish Settings, выберите команду Publish Preview меню File, а затем один из доступных форматов в списке. (Список доступных форматов будет содержать только те из них, которые были выбраны на вкладке Formats.) Flash автоматически загрузит выбранный тип файла.

## Анализ фильма

Для получения графического анализа скорости воспроизведения вашего фильма во время его тестирования выберите, используя команду Test Movie меню Control, команду Bandwidth Profiler меню View.

С помощью данного анализатора вы сможете увидеть размер (в килобайтах) каждого кадра фильма. Чтобы имитировать воспроизведение фильма при загрузке по модемному соединению, при тестировании выберите команду Streaming (Движение в потоке) меню View. Выбор различных параметров модемного соединения для тестирования осуществляется посредством меню Debug (Отладка).

## Публикация Flash-фильма

После установки всех параметров обработки и предварительного просмотра фильм готов к публикации. Чтобы его опубликовать, достаточно выбрать команду Publish меню File. Теперь Flash самостоятельно выполнит необходимые действия по созданию файлов соответствующих форматов. Все создаваемые файлы будут помещены в каталог, содержащий исходный файл с расширением FLA.

Кроме команды Publish меню File содержит команду Export. Отличаются они тем, что при выполнении команды Export вы не устанавливаете параметры в диалоговом окне Publish Settings, а переходите к диалоговому окну Export Movie (Экспортировать фильм), где можно указать тип создаваемого файла и каталог, в который будет помещен данный файл. Недостатком этого способа публикации является отсутствие доступа к параметрам обработки фильма.

## Публикация фильма, доступного для людей с физическими недостатками

Вы уже убедились, что существует большое число файловых форматов, в которых можно опубликовать Flash-фильм. Но существует одна проблема, о которой мы еще не говорили.

3. С помощью списка Layer (Слой) можно управлять в экспортируемом фильме положением объектов Flash-фильма относительно объектов фильма QuickTime.
  - > Выбрав элемент Top (Верх), вы поместите содержимое Flash-фильма на верхнюю дорожку фильма QuickTime.
  - > Если выбран элемент Bottom (Низ), содержимое Flash-фильма помещается на нижнюю дорожку фильма QuickTime.
  - > При выборе элемента Auto (Автоматически) дорожка Flash помещается перед другими дорожками в файле QuickTime, если объекты Flash в самом фильме находятся впереди видеообъектов. Если же объекты в фильме Flash находятся позади видеообъектов, дорожка Flash помещается за всеми другими дорожками файла QuickTime.
4. При установке флажка Use QuickTime Compression (Использовать сжатие QuickTime) параметра Streaming Sound весь аудиопоток Flash-фильма будет экспортирован на звуковую дорожку QuickTime. Причем звук будет повторно сжат с использованием стандартных звуковых параметров QuickTime. Изменить эти параметры можно с помощью диалогового окна Sound Settings (Параметры звука), которое открывается щелчком на кнопке Settings (Параметры). В данном окне можно выбрать алгоритм сжатия, скорость передачи битов, а также установить режим моно или стерео.

---

#### ПРИМЕЧАНИЕ

Вопросы, связанные с использованием звука, подробно описаны в главе 19.

---

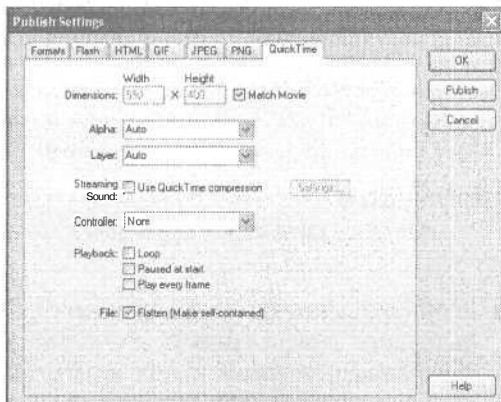
5. При просмотре фильма в формате QuickTime обратите внимание, что в нижней части экрана проигрывателя располагается контроллер, с помощью которого можно управлять воспроизведением фильма и регулировать звук. Раскрывающийся список Controller (Контроллер) позволяет выбрать стандартный контроллер, QuickTime VR-контроллер либо вовсе отказаться от его использования.
6. Если установить флажок Loop (Цикл) параметра Playback (Воспроизведение), фильм будет воспроизводиться циклически.
7. При установке флажка Paused at Start (Остановка в начале) параметра Playback воспроизведение фильма будет остановлено на первом кадре. Для того чтобы начать воспроизведение фильма, необходимо щелкнуть на кнопке контроллера Play (Воспроизведение).
8. Установка флажка Play Every Frame (Воспроизводить каждый кадр) отменяет установленную частоту смены кадров, при этом скорость воспроизведения фильма не сохраняется. Но в этом случае, несмотря на задержки, будет производиться каждый кадр фильма. Обратите внимание, что при установке данного флажка, выключается звуковая дорожка.
9. При установке флажка Flatten (Make Self-Contained) (Создать автономный файл) содержимое Flash-фильма не комбинируется с фильмом QuickTime. Вместо этого содержимое Flash-фильма публикуется в виде отдельного файла, связанного с фильмом QuickTime. Учтите, что все связанные файлы должны

- При выборе пункта Sub для каждого пиксела передается различие между каждым байтом и значением соответствующего байта предыдущего пиксела. Данный алгоритм прекрасно подходит для изображений, содержащих повторяющиеся элементы (например, полосы или клетки) вдоль горизонтальной оси.
- Когда выбран пункт Up, алгоритм сжатия функционирует противоположно предыдущей установке. Данный алгоритм больше подходит для изображений, содержащих повторяющиеся элементы вдоль вертикальной оси.
- При выборе пункта Average для вычисления значения пиксела используются значения двух соседних пикселей. Данный алгоритм применяется для изображений, содержащих повторяющиеся элементы (как по вертикали, так и по горизонтали).
- Если выбран пункт Path, для вычисления значения пиксела используются значения ближайших пикселей, расположенных сверху, слева и справа.

## Параметры Quick Time

Если вы хотите опубликовать Flash-фильм в формате QuickTime (расширение MOV), установите на вкладке Formats диалогового окна Publish Settings флажок QuickTime и перейдите на вкладку QuickTime.

На данной вкладке определяется способ обработки фильма при создании фильмов в формате QuickTime.



1. Параметры Dimensions позволяют установить размеры фильма QuickTime посредством ввода значений (в пикселах) в поля Width и Height. Если вы хотите, чтобы размеры фильма QuickTime соответствовали размерам фильма Flash, установите флажок Match Movie (Соответствовать фильму).
2. Раскрывающийся список Alpha позволяет управлять прозрачностью объектов Flash в фильме формата QuickTime.
  - Когда выбран элемент Auto (Автоматически), Flash-фильм становится прозрачным, если он занимает дорожку, расположенную выше любой другой дорожки фильма QuickTime. Но если Flash-фильм занимает единственную дорожку в фильме QuickTime либо дорожку, расположенную ниже любой другой дорожки, он становится непрозрачным.
  - При выборе элемента Alpha-Transparent Flash-фильм становится прозрачным. В этом случае на экран выводится содержимое дорожек, расположенных ниже дорожки Flash-фильма.
  - Если выбран элемент Copy, Flash-фильм становится полностью непрозрачным. Все дорожки, расположенные ниже дорожки Flash-фильма, блокируются.

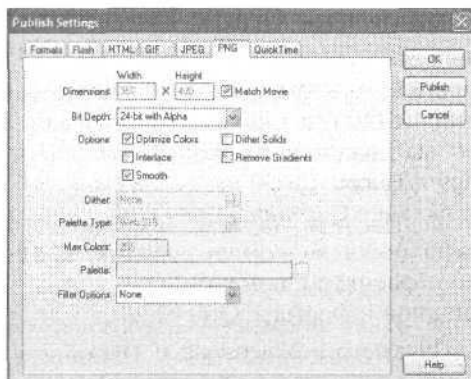
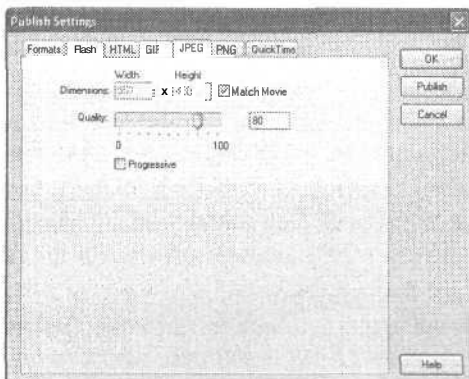
так и в расположенном справа от него поле. Чем меньшее значение вы установите, тем более сжатым будет файл.

- Установка флажка **Progressive** (Прогрессивное) позволяет создавать JPEG-изображение подобное чересстрочному изображению формата GIF.

## Параметры PNG

Если вы хотите опубликовать отдельный кадр фильма как PNG-файл, установите на вкладке **Formats** флажок **PNG Image** и перейдите на появившуюся вкладку **PNG**. Здесь вы сможете установить параметры обработки изображения кадра при создании PNG-файла. По умолчанию Flash экспортирует изображение первого кадра фильма. Но существует возможность так же, как при использовании форматов GIF и JPEG, указать любой кадр, изображение которого вы хотите сохранить в формате PNG. Для этого следует установить метку кадра **#Static** с помощью панели **Properties**.

Установка и действие большинства параметров PNG такие же, как для формата GIF, поэтому за информацией обращайтесь к разделу «Параметры GIF».



На вкладке PNG можно установить дополнительные параметры.

- Раскрывающийся список **Bit Depth** (Глубина цвета в битах) позволяет установить количество цветов в публикуемом PNG-изображении.
  - > При выборе пункта **8-bit** изображение будет иметь только 256 цветов.
  - > При выборе пункта **24-bit** создаваемое изображение может содержать 16,7 миллионов цветов. Размер файла при этом увеличится, но и качество рисунка заметно возрастет.
  - ▷ Если выбрать пункт **24-bit with Alpha**, в создаваемом изображении будут поддерживаться 16,7 миллионов цветов, а также дополнительный 8-битовый канал для обеспечения прозрачности. При создании PNG-файла с таким параметром незаполненные зоны изображения будут прозрачными.
- С помощью раскрывающегося списка **Filter Options** (Параметры фильтров) можно выбрать алгоритм сжатия, который будет использоваться при создании PNG-изображения.
  - > Если выбран пункт **None**, алгоритмы сжатия не применяются, и размер файла значительно увеличивается.



256 цветов, которые приблизительно соответствуют другим цветам. Используйте элементы этого списка для выбора метода, с помощью которого можно комбинировать пиксели при экспортировании фильма в GIF-формат.

- > Выбор элемента None (Нет) позволяет выключать размывку цветов и заменять цвета, не соответствующие элементам базовой палитры, чистыми наиболее близко подходящими цветами из палитры. Благодаря размывке цветов можно добиться уменьшения размера файла, но такое смешивание часто приводит к созданию странных цветов.
  - > При выборе элемента Ordered (Упорядоченный) можно добиться хорошего качества изображения, при этом файл будет иметь сравнительно небольшим размер.
  - > Используя элемент Diffusion (Рассеивание), можно получить наилучшее качество изображения, но размер файла и время его загрузки увеличатся.
10. С помощью списка Palette Type (Тип палитры) можно выбирать тип палитры, которая будет использоваться при публикации GIF.
- > Палитра Web 216 содержит 216 цветов, соответствующих стандартным цветам систем Windows и Macintosh.
  - > При выборе палитры Adaptive (Адаптивная) устанавливается пользовательская палитра, содержащая цвета, фактически присутствующие в изображении. Вы можете ограничить число цветов, введя значение в поле Max Colors.
  - > Палитра Web Snap Adaptive (Адаптивная с привязкой в Web) имеет только одно отличие от палитры Adaptive — цвета в ней конвертируются в цвета палитры Web 216.
  - > Для определения пользовательской палитры выберите палитру Custom (Пользовательская). Щелкните на кнопке с изображением многоточия, расположенной справа от поля Palette (Палитра), и в открывшемся окне Open найдите и выберите файл с расширением АСТ.

## Параметры JPEG

Если вы хотите опубликовать отдельный кадр фильма как JPEG-файл, установите на вкладке Formats флажок JPEG Image и перейдите на появившуюся вкладку JPEG. Здесь вы сможете установить параметры обработки изображения кадра при создании JPEG-файла. По умолчанию Flash экспортирует изображение первого кадра фильма, но посредством установки метки кадра #Static с помощью панели Properties вы можете указать любой кадр, изображение которого хотите сохранить в формате JPEG.

Поскольку для формата JPEG параметры Dimensions устанавливаются и действуют точно так же, как для формата GIF, мы не будем останавливаться на их описании. При необходимости вы можете обратиться к предыдущему разделу.

На вкладке JPEG можно установить дополнительные параметры.

1. Параметр Quality (Качество) позволяет определить степень сжатия, которая будет применяться для изображения в формате JPEG. Значение данного параметра в диапазоне от 0 до 100 можно установить как с помощью бегунка,

При необходимости экспортировать Flash-фильм как GIF-анимацию установите переключатель Playback в положение Animated (Динамический). Чтобы создать анимированный GIF-файл, содержащий только часть фильма, установите метку #First на начальном кадре нужного участка фильма и метку #Last на его последнем кадре.

Когда переключатель Playback установлен в положение Animated, становятся доступными дополнительные положения переключателя: Loop Continuously (Бесконечное воспроизведение) и Repeat (Повторить). При установке переключателя в положение Loop Continuously GIF-анимация будет воспроизводиться циклически. Установив переключатель в положение Repeat, вы можете определить число повторного воспроизведения GIF-анимации.

3. Если установить флажок Optimize Color (Оптимизировать цвет), Flash удалит все неиспользуемые цвета из палитры цветов GIF-файла. Это помогает уменьшить размер конечного файла. Учтите, что при выборе в поле списка Palette Type (Тип палитры) элемента Adaptive (Адаптивная) установка флажка Optimize Color не будет иметь эффекта.
4. При установке флажка Interlace (Чересстрочный) создается чересстрочное изображение в формате GIF. Этот формат позволяет развертывать изображение в окне браузера постепенно со все большей детализацией.
5. Установка флажка Smooth (Сглаживание) приведет к тому, что все элементы публикуемого фильма будут сглажены, но размер конечного файла при этом увеличится.
6. При установке флажка Dither Solids (Размывка цвета) размывка применяется как для чистых цветов, так и для градиентов.
7. Установка флажка Remove Gradients (Удалить градиенты) позволяет преобразовать все градиенты в чистые цвета. Поскольку градиенты не преобразуются в GIF-формат, снятие этого флажка может привести к нежелательным результатам.
8. С помощью раскрывающегося списка Transparent (Прозрачность) можно установить уровень прозрачности фона фильма, а также способ преобразования значения параметра Alpha Flash-фильма.
  - > При выборе пункта Opaque (Непрозрачный) фильм, публикуемый как GIF-файл, будет иметь непрозрачный фон.
  - > Если выбран пункт Transparent (Прозрачный), фильм, публикуемый как GIF-файл, будет иметь прозрачный фон. Учтите, что при выборе этого пункта в сочетании с переключателем Smooth вокруг всех объектов может появиться полупрозрачный ореол.
  - > Для получения возможности управлять прозрачностью отдельных объектов выберите пункт Alpha. В поле Threshold (Пороговая величина) можно ввести значение в диапазоне от 0 до 255. Это значение определит величину, ниже которой все цвета будут полностью прозрачными, а выше — полупрозрачными. Значение 128 соответствует величине Alpha 50 %.
9. Раскрывающийся список Dither позволяет создавать цвета, отсутствующие в текущей палитре. Это достигается путем создания комбинаций пикселей

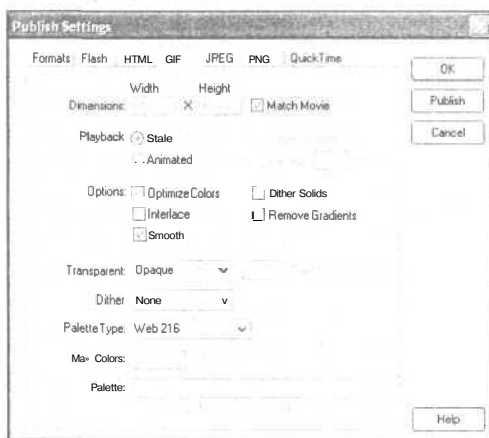
- > Если выбран режим No Border (Без рамки), размер фильма будет увеличен (без искажения) таким образом, чтобы заполнить всю зону, определенную параметрами Dimensions. При этом сохраняется соотношение геометрических размеров фильма. Отличие от предыдущего режима состоит в том, что для сохранения пропорций область фильма может быть установлена больше, чем это определено параметрами Dimensions, поэтому края фильма могут казаться обрезанными.
  - > При выборе режима Exact Fit (Точное соответствие) размер фильма устанавливается точно в соответствии с заданными параметрами Dimensions. При этом пропорциональное соотношение геометрических размеров фильма не сохраняется.
8. Параметры Flash Alignment (Flash-выравнивание) определяют положение фильма в пределах области, заданной в полях Dimensions. Данные параметры устанавливаются с помощью двух раскрывающихся списков — Vertical (По вертикали) и Horizontal (По горизонтали).
  9. Если установить флажок Show Warning Messages (Показывать предупреждающие сообщения), Flash будет выводить предупреждения обо всех конфликтах, возникающих при установке различных параметров публикации.

## Параметры GIF

После установки флажка GIF на вкладке Formats диалогового окна Publish Settings вы получаете доступ к вкладке GIF.

На данной вкладке определяется способ обработки фильма при создании GIF-файлов.

1. Используя поля Dimensions, можно установить размеры изображения для статического или анимированного GIF-файла. Данные размеры устанавливаются посредством ввода значений (в пикселах) в поля Width и Height. Если вы хотите, чтобы размеры GIF-изображения соответствовали размерам Flash-фильма, установите флажок Match Movie (Соответствовать фильму).
2. Переключатель Playback (Воспроизведение) имеет два основных и два дополнительных положения. При установке переключателя в положение Static (Статический) Flash экспортирует изображение одного определенного кадра. По умолчанию как статическое изображение в формат GIF экспортируется первый кадр Flash-фильма. Если вы хотите, чтобы Flash экспортировал изображение другого кадра, добавьте к этому кадру метку #Static, используя панель Properties. (Подробно об установке меток кадров рассказывалось в главе 9.)



- При выборе пункта Auto High (Автоматическое высокое) воспроизведение фильма начинается на высоком уровне качества. Но если компьютер пользователя не может справиться с таким уровнем качества при требуемой скорости воспроизведения, нужно установить качество фильма более низким (вплоть до уровня Low).
  - Если выбран пункт Medium (Среднее), воспроизведение фильма начинается с частичным использованием функции сглаживания — без сглаживания растровых изображений. При этом качество фильма будет выше чем в режиме Low, но ниже чем в режиме High.
  - Выбор пункта High (Высокое) приводит к полному использованию сглаживания в фильме. Если фильм содержит неанимированные растровые изображения, они также сглаживаются. Но это не касается анимированных растровых изображений.
  - Самый высокий уровень качества достигается при выборе пункта Best (Наилучшее). В этом случае фильм воспроизводится с наилучшим качеством, которое только достижимо, без учета возможности компьютера поддерживать требуемую скорость воспроизведения.
5. Раскрывающийся список Window Mode (Режим окна) позволяет установить режим воспроизведения фильмов в окне браузера Internet Explorer 4.0 (или выше) с поддержкой управляющих элементов ActiveX.
- При выборе режима Window (Окно) фильм воспроизводится в прямоугольной области на Web-странице. Важно отметить, что в этом случае скорость воспроизведения фильма будет самой высокой.
  - Выбор режима Opaque Windowless (Непрозрачный) приводит к перемещению элементов Dynamic HTML (в особенности слоев) на задний план. Таким образом, фильм воспроизводится над этими элементами.
  - Для того чтобы сквозь прозрачные области фильма видеть фон HTML-страницы, в которую вложен фильм, следует выбрать режим Transparent Windowless (Прозрачный), но учтите, что при этом снизится скорость воспроизведения фильма.
6. Раскрывающийся список HTML Alignment (HTML-выравнивание) позволяет определить положение фильма на Web-странице. По умолчанию область, в которой воспроизводится фильм, располагается в центре Web-страницы. Выбирая различные элементы списка, можно выровнять фильм по левому, правому, верхнему и нижнему краям в окне браузера.
7. В раскрывающемся списке Scale (Масштаб) содержатся элементы, которые действуют совместно с параметрами, установленными с помощью списка Dimensions.
- Режим Show All (Показывать все), установленный по умолчанию, позволяет увеличить размер фильма (без искажения) в соответствии с установленными параметрами Dimensions, сохраняя при этом соотношение геометрических размеров фильма. При выборе данного режима по краям области, в которой воспроизводится фильм, могут появиться рамки.

о котором речь пойдет позже. Назначение каждого из пунктов списка Dimensions описано далее.

- ▷ Выбор пункта Match Movie (Соответствие фильму) позволяет установить в полях Width и Height значения, соответствующие размерам фильма.
- ▷ Для ввода значения ширины и высоты в пикселах необходимо выбрать пункт Pixels (Пикселы) и ввести значения в поля Width и Height.
- ▷ При выборе пункта Percent (Процент) вы можете ввести в соответствующие поля значения ширины и высоты в процентах. Они будут определять размер окна, в котором происходит воспроизведение фильма, по отношению к основному окну браузера.

### 3. Параметры Playback (Воспроизведение) позволяют управлять воспроизведением фильма.

- ▷ При установке флажка Paused at Start (Остановка в начале) фильм автоматически останавливается на первом кадре. Чтобы продолжить воспроизведение, необходимо щелкнуть на кнопке, содержащей действие playO. В качестве альтернативы можно выполнить щелчок правой кнопкой мыши (Windows) или нажать клавишу Ctrl (Macintosh) и выбрать команду Play (Воспроизведение) в раскрывшемся контекстном меню.
- ▷ Установка флажка Loop (Цикл) применяется для циклического воспроизведения фильма.
- ▷ Если установлен флажок Display Menu (Экранное меню), становится доступным контекстное меню, раскрывающееся при щелчке правой кнопкой мыши (Windows) или щелчке при нажатой клавише Ctrl (Macintosh).
- ▷ При установке флажка Device Font (Шрифт устройства) все используемые в фильме шрифты, не установленные на компьютере пользователя, автоматически заменяются системными сглаженными шрифтами. Данная функция применяется только для фильмов, воспроизводимых на компьютерах с операционной системой Windows, и позволяет улучшить удобочитаемость небольших текстовых надписей и уменьшить общий размер фильма.

### 4. Раскрывающийся список Quality (Качество) позволяет установить качество воспроизведения фильма. Фактически посредством выбора различных элементов списка вы можете устанавливать приоритет качества над скоростью воспроизведения или наоборот.

- ▷ При выборе пункта Low (Низкое) отменяются любые функции сглаживания, и устанавливается приоритет скорости воспроизведения над качеством.
- ▷ Когда выбран пункт Auto Low (Автоматическое низкое), воспроизведение фильма начинается без функции сглаживания. Но если компьютер пользователя может справиться с улучшенным качеством, сохраняя при этом требуемую скорость воспроизведения, следует установить качество фильма более высоким (вплоть до уровня High).

Доступ к этому параметру вы можете получить в диалоговом окне Sound Settings (Звуковые параметры), которое открывается после щелчка на кнопке Set. Подробно об аудиопотоке и озвучивании событий рассказано в главе 21.

- При установке флажка Override Sound Settings (Замена звуковых параметров) все параметры звука, заданные на вкладке Flash диалогового окна Publish Settings (Параметры публикации), автоматически заменяют алгоритмы сжатия, установленные на панели Library.

## Параметры HTML

Если фильм Flash предназначен для распространения в Web, при публикации необходимо создать HTML-файл, в который будет вложен фильм. До недавнего времени для создания HTML-файла использовалась программа Aftershock, но начиная с версии Flash 5 компания Macromedia интегрировала функциональные возможности приложения Aftershock в диалоговом окне Publish Settings (Параметры публикации).

Если снять флажок Flash на вкладке Formats диалогового окна Publish Settings, то флажок HTML также будет недоступен. При установленном флажке HTML одновременно с вкладкой HTML доступной становится и вкладка Flash.

Параметры HTML включают:

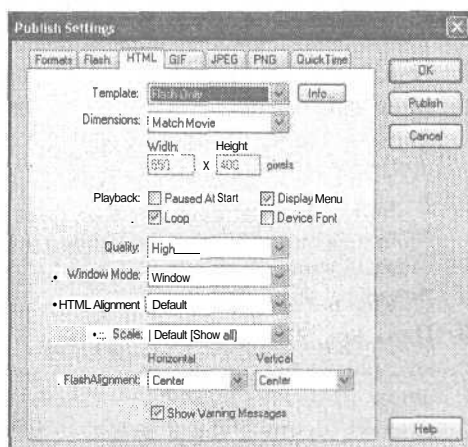
- Раскрывающийся список Template (Шаблон) позволяет выбрать из перечня заранее определенных HTML-шаблонов шаблон, который будет использоваться для воспроизведения Flash-фильма.

В большинстве случаев нет необходимости менять принятый по умолчанию шаблон Flash Only. Но для некоторых других типов файлов (например, QuickTime) лучше подходят другие шаблоны. (Подробную информацию о различных шаблонах вы найдете в справочном разделе Flash MX.)

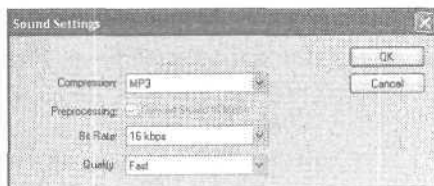
### СОВЕТ

Для получения информации об определенном шаблоне выберите его и щелкните на кнопке Info, расположенной справа от списка Template.

- Элементы раскрывающегося списка Dimensions (Размеры) позволяют управлять в HTML-документе значениями WIDTH (Ширина) и HEIGHT (Высота) в тегах OBJECT и EMBED. Важно понимать, что значения, которые введены в поля Width (Ширина) и Height (Высота), расположенные сразу под списком Dimensions, влияют не на размер Flash-фильма, а на размер области Web-страницы, в которой будет воспроизводиться фильм. Чтобы размер фильма соответствовал размеру данной области, необходимо использовать параметр Scale (Масштаб),



4. При установке флажка **Protect from Import** (Защита от импорта) файл формата SWF будет защищен от обратного импортирования во Flash. Данная функция позволяет защитить фильм от незаконного копирования его содержимого.
5. Установка флажка **Omit Trace Actions** (Пропустить действия trace) позволяет пропускать все содержащиеся в фильме действия trace, которые автоматически открывают окно Output. Подробно о действии trace рассказывалось в главе 18.
6. Установив флажок **Debugging Permitted** (Разрешить отладку), вы позволяете отладчику работать с опубликованным файлом. Кроме того, при установке данного флажка можно осуществлять дистанционную отладку фильма, используя браузер и подключаемый модуль **Flash Debug Player** или **ActiveX**.
7. При установке параметра **Compress Movie** (Сжатие фильма) осуществляется сжатие фильма, уменьшающее время загрузки и размер файла. Но фильм, сжатый таким способом, можно воспроизводить только с использованием приложения **Flash Player** версии 6.
8. В поле **Password** (Пароль) можно указать пароль, который будет использоваться при запуске отладчика или при импортировании файла обратно во Flash. Данное поле доступно, если установлен флажок **Protect from Import** либо **Debugging Permitted**.
9. Параметр **JPEG Quality** (Качество JPEG) позволяет определить степень сжатия, которая будет применяться для всех растровых изображений в фильме. Значение данного параметра в диапазоне от 0 до 100 можно установить как с помощью бегунка, так и в поле, расположенном справа от бегунка. Меньшему установленному значению соответствует более сжатый файл, соответственно, чем меньше размер файла, тем хуже качество фильма. При установке значения данного параметра необходимо выбрать правильное соотношение между качеством фильма и размером файла.
10. Параметр **Audio Stream** (Звуковой поток) предназначен для определения качества и степени сжатия звука. Посредством щелчка на кнопке **Set** вы получаете доступ к диалоговому окну **Sound Settings** (Звуковые параметры), где можете установить звуковые параметры вашего фильма. Раскрывающийся список **Compression** (Сжатие) позволяет установить тип сжатия, используемый для звука. Каждый тип сжатия (MP3, ADPCM, RAW, Disable и Speech) имеет свои параметры, которыми можно манипулировать.



#### ПРИМЕЧАНИЕ

Основные характеристики и параметры звука (сжатие, скорость передачи битов, частота выборки и т. д.) подробно рассмотрены в главе 19, а различные параметры сжатия — в главе 20.

11. Параметр **Audio Event** (Озвучивание событий) позволяет устанавливать различные параметры сжатия для озвучивания событий, используемых в фильме.

При установленном флажке Use default names (Использовать имена по умолчанию) создаваемые файлы будут иметь то же имя, что и файл FLA. После снятия флажка вы получаете доступ к полям, в которых можно изменить имена создаваемых файлов.

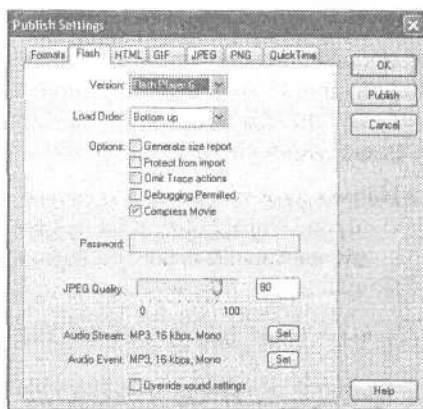
Поскольку для файлов типа Windows Projector и Macintosh Projector не существует отдельных параметров публикации, данные форматы не будут рассматриваться в следующих разделах. Но учтите, что при экспорте фильмов в файлы Windows Projector и Macintosh Projector действуют параметры обработки фильма, установленные на вкладке Flash.

## Параметры Flash

Установив на вкладке Formats диалогового окна Publish Settings флажок Flash (SWF), вы получите доступ к вкладке Flash.

Данная вкладка содержит следующие настройки.

1. В раскрывающемся списке Version (Версия) можно выбрать версию приложения Flash Player (1, 2, 3, 4, 5), с которой будет совместим создаваемый файл. При выборе версии необходимо быть внимательным, поскольку некоторые свойства публикуемого фильма могут не поддерживаться приложениями Flash Player более ранних версий. В большинстве случаев, когда речь идет о совместимости со старыми подключаемыми модулями Flash, проблемы могут возникнуть при использовании сценариев ActionScript.
2. Выбрав элемент раскрывающегося списка Load Order (Порядок загрузки), можно установить порядок загрузки и вывода на экран слоев в каждом отдельном кадре: Bottom Up (Восходящий) и Top Down (Нисходящий). Это особенно важно при загрузке Flash-файла по модемному соединению, поскольку обычно такая загрузка происходит медленно. Вам необходимо установить порядок, когда первыми загружаются слои с самыми важными элементами — большими графическими рисунками, звуками и т. д. Например, представьте, что верхний слой фильма содержит сценарий ActionScript, который загружается очень быстро. В случае установки порядка Top Down может получиться, что сценарий ActionScript загрузится и начнет исполняться до загрузки основных крупных элементов, размещенных на нижних слоях. Но если фильм загружается по скоростному соединению, порядок загрузки слоев не имеет большого значения.
3. Если установить флажок Generate Size Report (Создать отчет об объемах) раздела Options, программа будет создавать текстовый файл SimpleText (Macintosh) или TXT (Windows) с отчетом об объеме анимации и отдельных ее компонентов. Такой отчет полезен для оптимизации фильма.





## Windows Projector

Возможно, Windows Projector представляет наибольший интерес среди доступных форматов. По существу фильм в формате Windows Projector представляет собой исполняемый EXE-файл. Преимущество данного формата состоит в том, что для воспроизведения такого фильма не нужен ни браузер, ни подключаемый модуль Flash, ни соединение с Интернетом. Данный формат прекрасно подходит для распространения фильмов на компакт-дисках и DVD-дисках.

## Macintosh Projector

Данный формат предназначен для использования на компьютерах Macintosh и является эквивалентом формата Windows Projector. Чтобы воспроизвести исполняемый Hqx-файл, не нужен ни Web-браузер, ни подключаемый модуль Flash.

## QuickTime

При публикации Flash-фильмов в формате QuickTime создается файл с расширением MOV, воспроизвести который можно, используя приложение Quick-Time Player. Данный формат позволяет сохранить большую часть интерактивных свойств Flash-фильма. Но для воспроизведения файла с расширением MOV на компьютере пользователя должен быть установлен подключаемый модуль QuickTime.

### ПРИМЕЧАНИЕ

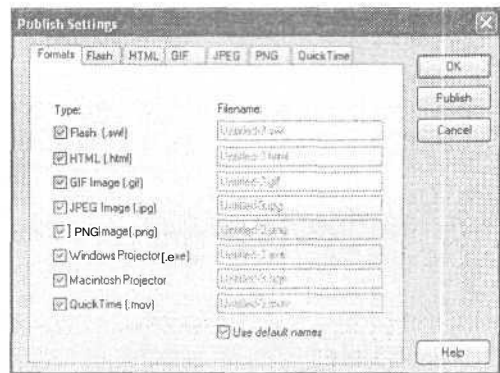
Чтобы опубликовать Flash-фильм в формате QuickTime, установите на своем компьютере приложение QuickTime версии 4.0 или выше.

В фильме формата QuickTime Flash-фильм будет занимать одну дорожку.

## Определение параметров публикации для различных форматов

В данном разделе будут рассмотрены вопросы, связанные с установкой параметров обработки фильма для различных файловых форматов. Установка всех параметров публикации осуществляется с помощью диалогового окна Publish Settings (Параметры публикации), открыть которое можно, выполнив команду File ▶ Publish Settings.

Если на вкладке Formats диалогового окна Publish Settings установить флажок, соответствующий типу какого-либо файла, автоматически появится дополнительная вкладка с названием выбранного типа. Щелкнув на этой вкладке, вы получите доступ к параметрам обработки фильма программой Flash. После установки значений всех параметров обработки Flash-фильма нажмите кнопку ОК.



- ▶ является поточным форматом: показ изображения в окне Web-браузера начинается во время его загрузки;
- ▶ может выводить частично прозрачные изображения, позволяющие видеть фон, поверх которого они помещены;
- ▶ имеет возможность сохранения в одном файле нескольких изображений, что применяется при изготовлении анимации. Когда браузер загружает этот файл, содержащиеся в нем изображения выводятся поочередно, создавая эффект движения.

Вы можете экспортировать Flash-фильм в файл формата Animated GIF для тех пользователей, на компьютере которых не установлен подключаемый модуль Flash. Однако Flash-фильм в формате Animated GIF будет иметь больший размер по сравнению с фильмом, опубликованным в формате SWF. Кроме того, при экспорте в формат Animated GIF все интерактивные свойства Flash-фильма утрачиваются.

## JPEG

Формат JPEG (Joint Photography Experts Group — Объединенная группа экспертов в области фотографии) предназначен для передачи полноцветных изображений с количеством цветов 16,7 млн. В результате, файлы в этом формате имеют больший размер по сравнению с файлами в формате GIF. Размер файла JPEG напрямую зависит от качества изображения — чем выше качество, тем больше размер файла. К сожалению, изображения в данном формате могут быть только статическими: никакой прозрачности, чересстрочности и, безусловно, никакой анимации.

По умолчанию Flash экспортирует в формат JPEG первый кадр фильма как статическое изображение. Но можно экспортировать в JPEG-изображения не только первый, но и другие кадры. Об этом будет рассказано в разделе «Параметры JPEG».

## PNG

Формат PNG (Portable Network Graphics — переносимая сетевая графика), частично разработанный компанией Macromedia и утвержденный в качестве Web-стандарта консорциумом WWW (World Wide Web Consortium), несколько более сложный, чем форматы GIF и JPEG. Разработчики этого стандарта постарались объединить все лучшее из форматов GIF и JPEG. В результате формат PNG может поддерживать изображения с 256 цветами, полутоновые изображения, изображения фотографического качества (содержащие миллионы цветов), а также прозрачность. Проблема при использовании формата PNG состоит в том, что его поддерживают не все браузеры. Изображения PNG-формата частично поддерживаются такими браузерами, как Microsoft Internet Explorer (4.0 или выше) и Netscape Navigator (4.04 или выше). Поскольку PNG является «родным» файловым форматом приложения Fireworks, Flash MX имеет в своем арсенале некоторые специальные инструменты, предназначенные для управления файлами PNG.

Как и формат JPEG, Flash может экспортировать в формат PNG только одиночное статическое изображение.

фильм в соответствии с установленными параметрами. Действия, необходимые на первом этапе, мы рассмотрим после описания типов форматов, а о втором этапе расскажем в разделе «Публикация Flash-фильма».

## Форматы фильмов

При публикации фильмов Flash можно создавать различные типы файлов, соответствующие определенным форматам. Каждый формат имеет свои достоинства и недостатки, поэтому для правильного выбора вы должны знать особенности каждого такого формата.

### Flash

В большинстве случаев при публикации фильмов вы будете создавать файлы с расширением SWF. Этот формат принят во Flash по умолчанию и является единственным, где реализована полная поддержка сценариев ActionScript и эффектов анимации. Для воспроизведения файлов с расширением SWF на компьютере пользователя должен быть установлен подключаемый модуль Flash. Поскольку в других форматах, таких как QuickTime, доступны не все свойства Flash-фильмов, мы рекомендуем для полной реализации возможностей программы применять именно формат SWF.

Единичные файлы с расширением SWF можно воспроизводить с помощью приложения Flash Player. Но чтобы распространять фильмы в Web, когда для просмотра фильма используется окно браузера, необходимо кроме SWF-файла создать HTML-файл, в который будет вложен Flash-фильм. Данный процесс будет подробно рассмотрен в разделе «Параметры HTML».

### HTML

Чтобы просмотреть SWF-файлы с помощью Web-браузера, необходимо создать HTML-файл, в который должен быть вложен Flash-фильм. Файл HTML служит для активизации фильма, спецификации установок браузера и в определенной степени для определения того, как будет выглядеть фильм. Таким образом, если SWF-файл предназначен для распространения в Web, он должен публиковаться вместе с HTML-файлом.

Приложение Flash Player, которое устанавливается на компьютер пользователя при подключении модуля Flash, может воспроизводить файл с расширением SWF без помощи HTML-страницы. Поэтому HTML-файл необходим только в тех случаях, когда зритель собирается просматривать фильм в окне браузера.

### GIF

Формат GIF (Graphic Interchange Format — формат обмена графическими данными) разработан в конце 80-х годов компанией CompuServe и получил широкое распространение как основной формат передачи графических данных в Web. Изображение в этом формате может содержать всего 256 цветов. И хотя качество такого изображения не слишком высокое, формат GIF обладает некоторыми преимуществами, главное из которых — **небольшой** размер файлов сравнительно с другими графическими форматами (например, JPEG). Кроме того, GIF-формат

- ▶ поддерживает чересстрочную развертку, то есть развертка изображения происходит в окне браузера постепенно со все большей детализацией;

## Глава 27

# Публикация Flash-фильмов

Любой фильм создается для показа зрительской аудитории. Чтобы донести фильм до зрителя, нужно представить его в соответствующем формате.

В данной главе будут рассмотрены вопросы, касающиеся преобразования файлов с расширением FLA (формат Flash-файлов по умолчанию) в формат, подходящий для распространения в Web или посредством других носителей. Этот процесс называется *публикацией*. Публикация фильма не является сложной, но вам потребуются специальные знания о достоинствах и недостатках того или иного формата представления фильма. Кроме того, правильная настройка параметров обработки фильма перед его публикацией в значительной степени поможет улучшить восприятие конечного продукта.

В настоящей главе рассматриваются следующие темы:

- ▶ установка параметров обработки Flash-фильма при его публикации;
- ▶ предварительный просмотр Flash-фильма;
- ▶ оптимизация Flash-фильма;
- ▶ публикация Flash-фильма;
- ▶ публикация фильма, доступного для людей с физическими недостатками;
- ▶ создание фильма, содержимое которого может быть выведено на печать.

## Установка параметров обработки Flash-фильма при его публикации

В большинстве случаев Flash-фильмы публикуются как файлы с расширением SWF. Фильмы данного формата можно просматривать с помощью браузеров при установленном подключаемом модуле Flash. Однако Flash поддерживает создание фильмов и в некоторых других форматах, что позволяет охватить более широкую аудиторию зрителей. Например, **неинтерактивный** фильм для Web можно опубликовать в форматах SWF и QuickTime. При этом достаточно, чтобы компьютер пользователя поддерживал хотя бы один из этих форматов. Еще одним достоинством Flash является **возможность** одновременного создания нескольких файлов различных форматов. Кроме того, определяя параметры обработки фильма для преобразования в другой формат, вы можете заранее установить визуальные характеристики и свойства фильма.

Особенности различных форматов будут рассмотрены в последующих разделах, после чего мы опишем индивидуальные установки параметров публикации фильма для каждого формата.

Что касается процесса публикации фильма, то он включает два основных этапа. Это, во-первых, установка значений параметров обработки фильма при публикации, а во-вторых, отправка сообщения Flash о необходимости опубликовать

# ЧАСТЬ 8

---

## Публикация и распространение Flash-фильмов

*Ваших знаний и опыта уже достаточно для того, чтобы создавать различные изображения и анимацию, а также интерактивные фильмы с использованием сценариев языка ActionScript. Но для создания конечного продукта этого мало, ведь нужно еще сделать вашу анимацию доступной широкой аудитории. Цель любого Flash-проекта будет достигнута лишь тогда, когда итог работы смогут увидеть и оценить зрители.*

*В этой части книги речь пойдет о публикации и распространении Flash-фильмов. Прежде всего, мы расскажем, как конвертировать файл с расширением FLA в один из форматов, в котором можно распространять Flash-фильмы (основным среди них является формат SWF). Далее будут рассмотрены вопросы, связанные с использованием во Flash-фильмах фильмов других видеоформатов. Наконец, вы узнаете обо всех деталях, касающихся распространения Flash-фильмов на компакт-дисках.*

|                                                               |     |
|---------------------------------------------------------------|-----|
| <b>Глава 27.</b> Публикация Flash-фильмов.....                | 554 |
| <b>Глава 28.</b> Работа Flash с другими видеоформатами.....   | 574 |
| <b>Глава 29.</b> Публикация Flash-фильма на компакт-диск..... | 600 |

13. Теперь, когда логотип установлен в нужную позицию, пришло время использовать панель свойств, чтобы сделать окончательные настройки. Выберите Flash-фильм. Перейдите в панель свойств (если эта панель не открыта, откройте ее выбором команды Properties в меню Window).
14. Расширьте панель свойств, щелкнув на маленькой направленной вниз стрелке, которая расположена в нижнем правом углу панели.
15. В списке Quality (Качество) выберите элемент High (Высокое).
16. Установите флажок Autoplay. В этом случае при открытии HTML-страницы фильм будет воспроизведен автоматически.

---

**ПРИМЕЧАНИЕ**

Вы можете также установить флажок Loop (Повтор), хотя в данном случае задавать циклическое повторение нет необходимости, поскольку анимация находится в клипе, а он по умолчанию повторяется автоматически.

---

17. В списке Scale (Масштаб) выберите элемент Default (Show All).
18. Для предварительного просмотра фильма нажмите кнопку Play.
19. Теперь активизируйте команду File ▶ Save, чтобы сохранить документ. В открывшемся окне Save As укажите папку, куда нужно записать HTML-файл (он должен находиться в той же папке, что и файл retro\_logo.swf), введите имя retro\_splash в поле Name (Имя) и нажмите кнопку Save.

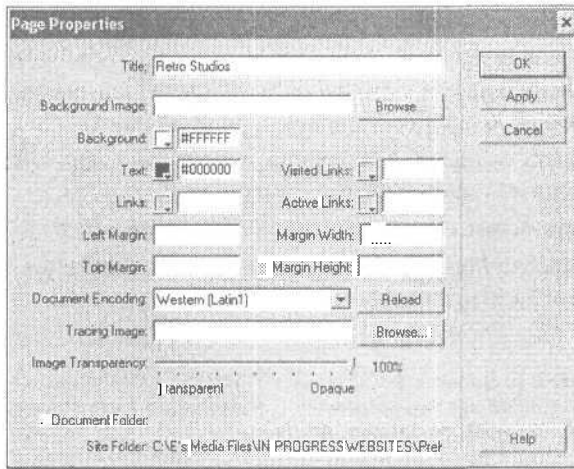
---

**ПРИМЕЧАНИЕ**

Если вы сохраните HTML-документ не в той же папке, где находится SWF-файл, то при отображении документа браузер не сможет найти SWF-файл, и его содержимое не отобразится на экране.

---

20. Наконец, чтобы просмотреть получившийся логотип в Web-браузере, нажмите клавишу F12 или воспользуйтесь командой File ▶ Preview in Browser (Файл ▶ Предварительный просмотр в браузере).



6. В окне Page Properties нажмите кнопку Apply (Применить), а затем кнопку OK.
7. Поместите логотип в документ Dreamweaver. Для этого активизируйте команду Insert ► Media ► Flash (Вставка ► Носитель ► Flash).

### СОВЕТ


Вместо указанной команды можно использовать кнопку Insert Flash (Вставить Flash), расположенную в разделе Common (Общий) палитры Objects (Объекты). Эта палитра открывается после выбора в меню Window команды Objects.

8. Найдите файл `retro_logo.swf` в открывшемся окне Select File (Выбор файла), выделите его и щелкните на кнопке Select (Выбрать).
9. В ответ на запрос программы Dreamweaver, использовать ли для Flash-файла относительный путь, нажмите кнопку OK, а в ответ на запрос, следует ли копировать Flash-файл на локальный узел, нажмите кнопку No.

### ПРИМЕЧАНИЕ

Пусть вас не беспокоит, что после помещения Flash-файла в документ Dreamweaver он будет отображаться как серый квадрат с пиктограммой Flash в середине. Вы сможете выполнить предварительный просмотр анимации в самой программе Dreamweaver; как это сделать, мы расскажем дальше. После помещения SWF-файла в документ Dreamweaver необходимо разместить этот логотип так, чтобы Web-страница имела привлекательный вид. Размер логотипа можно изменить перетаскиванием правого нижнего угла серого квадрата, изображающего импортированный SWF-файл.

10. Щелкните чуть правее Flash-файла (чтобы он не оказался выбранным). Затем откройте панель свойств, активизировав в меню Window элемент Properties.
11. Нажмите кнопку Align Center (Выровнять по центру) — среднюю из трех кнопок выравнивания, расположенных в верхнем правом углу панели свойств.
12. Щелкните чуть левее файла и три или четыре раза нажмите клавишу Enter/Return. Таким образом вы передвинете Flash-фильм немного вниз, чтобы он расположился посередине страницы.

24. Откройте окно библиотеки (если оно не открыто), выбрав в меню Window (Окно) элемент Library (Библиотека).
25. Затем выделите первый ключевой кадр в слое background и перетащите символ background из библиотеки на область действия.
26. Выберите первый ключевой кадр в слое 3D Animation и перетащите клип 3D object из библиотеки на область действия. Разместите его, как показано на рисунке. 
27. Возвратитесь на главную временную шкалу, выберите первый ключевой кадр ее единственного слоя и перетащите клип 3D animation из библиотеки на область действия. Установите логотип посередине области действия.
28. Протестируйте созданный фильм, выбрав команду Control ▶ Test Movie.
29. Наконец, перед тем как посредством программы Dreamweaver поместить логотип в HTML-документ, необходимо экспортировать результат работы в SWF-файл. Для этого активизируйте команду File ▶ Export Movie (Файл ▶ Экспортировать фильм).

#### СОВЕТ

Хотя это может показаться вам очевидным, напоминаем, что формат FLA не воспринимается Web-браузером. Поэтому нужно экспортировать анимацию в формат SWF.

30. Введите имя retro\_logo в поле Name открывшегося окна Export Movie, укажите папку, в которой нужно сохранить файл и нажмите кнопку Save.
31. В открывшемся окне Export Flash Player (Экспорт для проигрывателя Flash) оставьте все настройки по умолчанию и нажмите кнопку ОК.

## Вставка анимации в документ Dreamweaver



Теперь, когда вы создали анимированный трехмерный логотип и экспортировали его из Flash в формате SWF, можно поместить его в HTML-файл при помощи программы Dreamweaver.



Демонстрационная версия программы Dreamweaver находится на прилагаемом компакт-диске.

1. Запустите программу Dreamweaver.
2. Активизируйте команду Modify ▶ Page Properties (Модификация ▶ Параметры страницы), чтобы открыть окно Page Properties. Измените в этом окне цвет фона так, чтобы он соответствовал цвету фона анимации.
3. В поле Title (Заголовок) введите заголовок Web-страницы Retro Studios.
4. Щелкните на образце цвета Background (Фон) и в раскрывшейся палитре цветов выберите значок системной палитры (маленький значок с цветовым кругом в верхнем правом углу палитры).
5. В открывшемся окне введите значение 153 в поля Red, Green и Blue и щелкните на кнопке ОК.



8. Выберите только что созданный текст и откройте панель свойств командой Windows ▶ Properties.
9. На панели свойств выберите шрифт Eight Track Program 3 в списке Font.
-  Шрифт Eight Track Program 3 создан компанией Fontalicious ([www.fontalicious.com](http://www.fontalicious.com)). Поскольку этот шрифт следует установить на компьютер, мы поместили его на компакт-диск в каталог Hand On 7.
10. Введите для размера шрифта значение 35 в поле Font Size (Размер шрифта).
11. В поле Character Spacing (Разрядка символов) введите 1.
12. Задайте черный цвет символов текста.
13. Включите стили Bold (Полужирный) и Italic (Курсив).
14. После задания параметров оформления текста щелкните на области действия вне этого текста, чтобы отменить выделение.
15. Выберите инструмент Arrow и разместите текст, как показано на рисунке. 
16. После создания статических элементов логотипа пришло время добавить к нему SWF-файл с анимацией, который вы экспортировали из программы Amorphium Pro. Для этого перейдите на главную временную шкалу.
17. Активизируйте команду Insert ▶ New Symbol (Вставка ▶ Новый символ). В поле Name открывшегося окна Create New Symbol введите имя 3D object, установите переключатель Movie Clip и нажмите кнопку ОК. Программа Flash перейдет в режим редактирования символа.
18. Активизируйте команду File > Import (Файл ▶ Импортировать).
19. В открывшемся окне Import (Импорт) выберите сохраненный ранее файл 3D\_object.swf и нажмите кнопку Open. Содержимое SWF-файла будет импортировано в виде покадровой анимации, где каждый кадр соответствует кадру оригинального файла Amorphium Pro.

## ПРИМЕЧАНИЕ

Трехмерная анимация, которую вы импортируете, помещается не в центре окна. Поскольку это анимация является покадровой, очень сложно изменить ее положение, ведь это нужно сделать для каждого кадра. Однако смещение анимации при редактировании символа практически не повлияет на фильм, просто регистрационная точка будет расположена не в центре окна.

20. Теперь пришло время скомбинировать символ заднего плана с анимацией, чтобы затем разместить его на главной временной шкале для создания окончательного варианта логотипа. Активизируйте команду Insert ▶ New Symbol.
21. В поле Name открывшегося окна Create New Symbol (Создание нового символа) введите имя logo, установите переключатель Movie Clip и нажмите кнопку ОК.
22. В режиме редактирования созданного символа присвойте существующему слою имя background.
23. Добавьте еще один слой и назовите его 3D Animation.

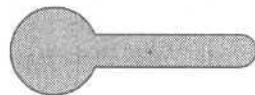
После этого начнется формирование SWF-файла кадр за кадром. В зависимости от производительности компьютера данный процесс займет некоторое время. Когда программа Amorphium Pro завершит создание фильма, она воспроизведет его в окне предварительного просмотра.

16. Закройте окно предварительного просмотра.
17. Активизируйте команду Project ▸ Quit (Проекты ▸ Выйти), завершив таким образом работу программы Amorphium Pro.

## Работа с анимацией во Flash

После экспорта анимации из программы Amorphium Pro добавим некоторые элементы к логотипу в среде Flash.

1. Запустите Flash.
2. Активизируйте команду Modify ▸ Document (Модификация ▸ Документ).
3. В открывшемся окне Document Properties (Параметры документа) задайте размеры области действия равными 550x300 пикселей. Затем щелкните на образце Background Color и выберите для фона серый цвет #999999.
4. Далее начните создавать основную фигуру на заднем плане. Для этого активизируйте команду Insert ▸ New Symbol (Вставка ▸ Новый символ), вследствие чего откроется окно Create New Symbol (Создание нового символа). В поле Name введите имя background, установите переключатель Graphic (Графический) и нажмите кнопку ОК.
5. В режиме редактирования символа используйте инструменты рисования и раскраски для создания фигуры, подобной той, что показана на рисунке (убедитесь, что фигура занимает отдельный ключевой кадр).



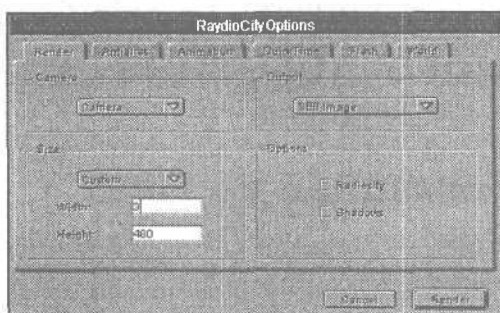
Не имеет значения, какой инструмент будет применяться для создания фигуры. Например, вы можете нарисовать закругленный прямоугольник и скомбинировать его с эллипсом или создать эту фигуру при помощи инструмента Реп. Самое главное, чтобы цвет заливки фигуры был следующим: значение красной составляющей — 255, зеленой — 152, а синей — 0. Также необходимо, чтобы линии, образующие контур, имели черный цвет и толщину 3 пикселя. Размеры фигуры должны быть приблизительно такими, как на предыдущем рисунке.

### ПРИМЕЧАНИЕ

Можете открыть готовый файл `retro_logo fla`, который находится в папке Hands On 7 на прилагаемом компакт-диске, и скопировать оттуда символ заднего плана.

6. После создания фигуры заднего плана добавьте еще один слой и назовите его text.
7. Выделите первый кадр слоя text, выберите инструмент Text, щелкните в области действия и введите текст `retro-labs`. (Пока можете поместить его в любом месте, а передвинете позже.)

- Щелкните в любом месте области Camera (Perspective), чтобы открыть окно RaydioCity Options. В нем нужно задать некоторые параметры экспортируемого SWF-файла.



- На вкладке Render в области Size выберите размеры 128x128.
- В списке Camera выберите элемент Camera.
- Затем в области Output (Вывод) выберите Flash Animation.
- Перейдите на вкладку Flash и в списке Lines (Линии) выберите элемент Outlines (Контуры).
- Затем в списке Polygons (Многоугольники) выберите элемент Average (Обычный).
- Теперь зададим цвета для экспорта в SWF-файл. Сначала щелкните на маленькой пиктограмме с цветным кругом, расположенной в нижнем левом углу образца Global Ambient, который находится в нижнем правом углу вкладки Flash окна RaydioCity Options.

#### СОВЕТ -

Цвет образца Global ambient — это цвет источника света, освещающего все объекты сцены. Можно выключить этот источник, но при этом все объекты анимации или изображения станут невидимыми, поскольку сцена будет залита черным цветом. Поэтому выбирайте не очень темный и не очень светлый цвет источника света.

- В открывшемся окне Color Picker (Палитра цветов) введите значение 128 в поля Red, Green и Blue, а затем нажмите кнопку ОК.
- Теперь следует задать цвет фона. Сначала щелкните на маленькой пиктограмме с цветным кругом, расположенной в нижнем левом углу образца Background Color (Цвет фона).
- В открывшемся окне Color Picker введите значение 255 в поля Red, Green и Blue, а затем нажмите кнопку ОК.
- После задания параметров на вкладке Flash нажмите кнопку Render (Визуализировать).
- В открывшемся окне Save As укажите папку, в которой хотите сохранить файл, введите имя 3D-object в поле Name (Имя) и нажмите кнопку Save (Сохранить).

## Практикум 7

# Создание анимированного трехмерного логотипа

Седьмая часть этой книги была посвящена использованию Flash совместно с другими программами. В данном практическом задании вы сможете применить полученные знания, чтобы создать анимированный трехмерный логотип Flash для некой студии Retro Studios. Начнем с экспорта трехмерной модели, созданной программой Amorphium Pro, в SWF-файл, который затем будет импортирован во Flash для добавления к логотипу некоторых графических элементов. Затем этот анимированный логотип будет помещен в HTML-документ, созданный программой Dreamweaver.

### ПРИМЕЧАНИЕ

Разумеется, создание трехмерного логотипа с анимацией — это сложный процесс, и многие его этапы выходят за рамки данного практикума. Если вы хотите получить больше информации о программе Amorphium Pro, загрузите ее полное описание в PDF-формате с Web-страницы компании Electric Image по адресу [www.electricimage.com/amorphium/manual.html](http://www.electricimage.com/amorphium/manual.html).



Демонстрационные версии программ Amorphium Pro и Macromedia Dreamweaver находятся на прилагаемом компакт-диске. Кроме того, на нем в каталоге Hands On 7 хранятся все файлы, представленные в этом практическом задании, и готовые файлы (вы сможете увидеть, как будет выглядеть проект, хотя для нас гораздо важнее процесс разработки, чем ее результат).

## Экспорт трехмерной анимации

Начнем с работы в программе Amorphium Pro.

1. Запустите Amorphium Pro. Когда появится окно Save As (Сохранить как), нажмите кнопку Cancel (Отмена).
2. Теперь активизируйте команду Open Project (Открыть проект) меню Project (Проект), вследствие чего откроется окно Open (Открыть). Перейдите в папку Hands On 7 на прилагаемом компакт-диске, выберите файл logo.cmf и нажмите кнопку Open.

### СОВЕТ

Если вы уже работали с каким-либо документом в Amorphium Pro, он автоматически откроется после запуска программы. Закройте его перед созданием нового документа, активизировав команду Project > Close (Проект > Закрыть).

3. После открытия трехмерного анимированного логотипа нажмите на кнопку RaydioCity, которая расположена в нижней части палитры Composer (Композиция).

## Разработки

Поскольку Flash относительно недавно используется в области трехмерного моделирования, количество интересных разработок ограничено. Одна из них — Web-страница Titoonic (рис. 26.6).

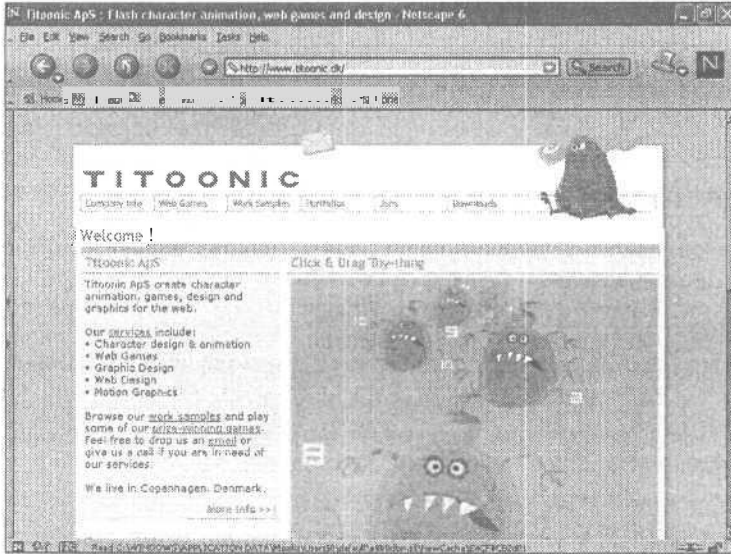


Рис. 26.6. Web-страница Titoonic

Эта Web-страница, появившаяся в Интернете в августе 2000 года ([www.titoonic.com](http://www.titoonic.com)), принадлежит датской компании, специализирующейся на выпуске Web-мультимедиа и 3D-продукции: справочников, игр, поздравительных открыток и т. п.

Созданная специалистами, выросшими на классической анимации и иллюстрации, Titoonic своим стилем и неотразимым дизайном расширяет привычные рамки трехмерного мира Flash.

## Резюме

Глава посвящена вопросам работы с трехмерной графикой в среде Flash.

Инструменты Flash не приспособлены к созданию полноценных трехмерных объектов. Однако, зная законы перспективы, а также приемы работы с источниками света и языком ActionScript, в этой программе можно симитировать трехмерное пространство.

Кроме того, экспорт объектов в среду Flash поддерживают многие современные программы трехмерного моделирования: Swift 3D, Amari 3D, Vecta3D, Poser и Amorphium Pro.

Завершает главу обзор программ, которые не оснащены средствами экспорта во Flash, но, возможно, являются будущим Flash в мире трехмерного мультимедиа.

Carrara Studio относится к программам среднего уровня сложности. Одна из наиболее интересных ее особенностей — интуитивный интерфейс, разделенный на «комнаты», каждая из которых имеет свой набор инструментов. Всего насчитывается пять комнат: Assemble, Storyboard, Model, Texture и Render (пиктограммы для получения доступа к ним расположены в правом верхнем углу окна программы). В одной комнате создается модель, в другой на нее наносятся текстуры, в третьей выполняется рендеринг и т. д.

Основной является комната Assemble. В ней вы можете позиционировать и анимировать модели, размещать источники света и камеры. Наиболее интересной особенностью этой комнаты является трехмерная рабочая область, где видны оси X, Y и Z. Любые импортированные модели отображаются как трехмерные объекты с проекциями на плоскостях, образуемых осями координат. Эти проекции позволяют разместить объект в трехмерном пространстве, их также можно использовать для выделения, перемещения и масштабирования объекта.

Carrara поддерживает несколько систем моделирования, каждая из которых позволяет создавать объекты, используя сплайны, вершины, текст и технологию MetaBalls. В Carrara Studio полностью поддерживается логическое моделирование. Если вы расположены к математике, то можете создавать модели с помощью математических формул. Программа предоставляет полный набор модификаторов для изменения формы и пути движения объекта.

Carrara Studio поставляется с полноценно функционирующей версией Amari 3D5, поэтому ее файлы можно сначала экспортировать в формат Amari 3D, а затем — в формат SWF. Наконец, Carrara Studio предоставляет возможность экспорта файлов в 3DS, что делает возможным их открытие в Vecta3D.

## Maya

Maya, продукт канадской компании Alias|Wavefront ([www.aliaswavefront.com](http://www.aliaswavefront.com)), является предком всех программ трехмерного моделирования. Эту программу можно смело назвать вершиной 3D-индустрии. Вы не найдете более мощной, а также более сложной и дорогостоящей программы.

Программа Maya используется главным образом при создании художественных фильмов. Нельзя утверждать, что дизайнеры-индивидуалы или небольшие творческие коллективы не согласились бы выбрать для работы эту программу. Этот инструмент недоступен им по причине высокой стоимости — 16000 долларов. Каких результатов позволяет добиться Maya, можно увидеть в фильмах «Гринч — похититель рождества», «Мумия», «Мумия возвращается», «Звездные войны» и «Враг у ворот».

Казалось бы, одно упоминание об этой программе звучит несерьезно. Стоит ли говорить о ее совместном использовании с Flash? И да и нет. Поскольку Maya не поддерживает экспорт во Flash, компания Alias|Wavefront выпустила приложение, которое конвертирует файлы из Maya в Shockwave 3D (3D-формат для Macromedia Director 8.5 Shockwave Studio). Этот шаг компания сделала, чтобы обеспечить подготовку высококачественной видеопродукции для Интернета. Роль посредника, несомненно, открывает перед Flash блестящие перспективы.

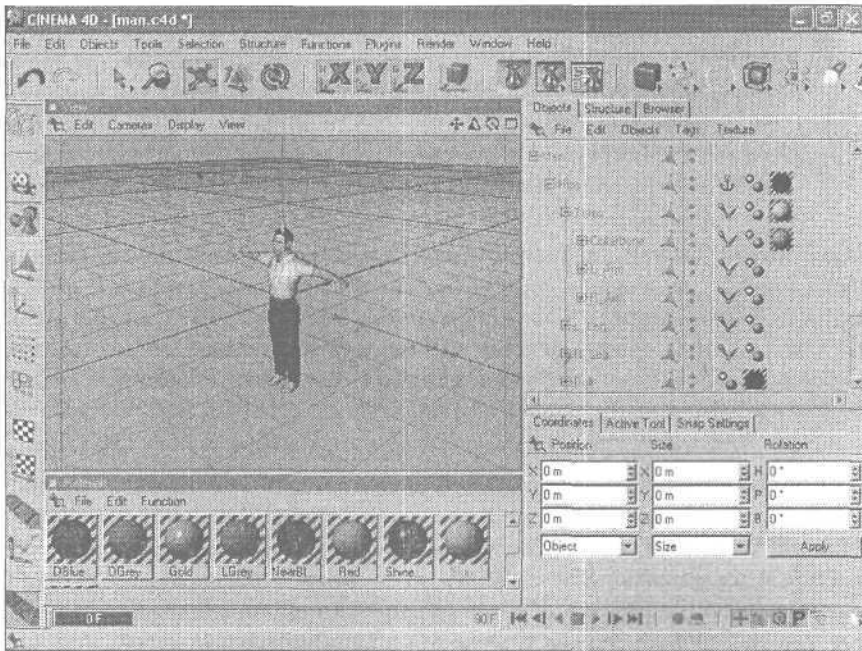


Рис. 26.5. Интерфейс программы Cinema 4D

Cinema 4D снабжена сложной системой теней и материалов, средствами поддержки различных схем освещения и инструментами анимации. В Cinema 4D интегрирован редактор BodyPaint, позволяющий раскрашивать любой трехмерный объект в интерактивном режиме.

## Lightwave 3D

Программа LightWave 3D, разработанная компанией NewTek ([www.newtek.com](http://www.newtek.com)), поражает широтой возможностей. Прежде всего следует отметить поверхности, разделяемые в реальном времени, многочисленные инструменты для работы с текстурами, 3D-ластики (caustics), элементы трехмерного изображения (HyperVoxels), средства настройки рассеянного света и трассировки лучей, а также исключительно мощную систему анимации.

Компания Electric Rain (создатель Swift 3D) разработала дополнение для LightWave 3D — Swift 3D LW. Это средство позволяет пользователям LightWave 3D осуществлять прямой экспорт во Flash.

## Carrara Studio

История программы Carrara Studio напоминает путь, пройденный приложением Poser 4.03. Созданное компанией MetaCreation приложение Carrara оказалось на грани исчезновения, но, к счастью, его разработчики основали свою компанию — Eovia — и занялись продвижением продукта. Сначала они подготовили программу модернизации, исправляющую в Carrara мелкие ошибки, а затем, в июне 2001 года, выпустили в свет приложение Carrara Studio.

поддерживающего экспорт во Flash, а Flash-дизайнеры, готовые уделить достаточно времени изучению профессиональной программы трехмерного моделирования, будут разочарованы отсутствием в ней возможности экспорта в среду Flash. Выход из этой ситуации есть. Наиболее примечательное свойство описанной выше программы Vesta3D — это способность открывать файлы, созданные в программе 3ds max, и экспортировать их во Flash. Кроме того, многие компании разрабатывают дополнения, которые позволяют вашей любимой программе трехмерного моделирования выполнять экспорт в формат SWF.

В этой связи представляем серию мощных программ трехмерного моделирования тем, кто желает стать «трехмерным» Flash-гуру.

## 3ds max

Наибольшее распространение на рынке программ трехмерного моделирования получила 3ds max. Сейчас это ведущий инструмент трехмерного моделирования. Очень сложно найти разработчика игр, который не использовал бы это приложение.

Программа 3ds max поддерживает все методы моделирования. В частности, существует возможность работы с NURBS, многоугольниками, примитивами, разнообразными эффектами, материалами, освещением, макроязыком. Предусмотрена возможность пакетного выполнения команд пользовательского интерфейса.

Для лиц, заинтересованных в использовании трехмерных моделей в среде Flash, компания Electric Rain (создатель Swift 3D) разработала дополнение Swift 3D Max, обеспечивающее прямой экспорт во Flash.

В августе 2001 года компании Macromedia и Discreet стали партнерами. Цель этой акции — ускорение развития интерактивного трехмерного содержимого Web. В рамках этой программы было выпущено приложение Director, последняя версия которого обладает мощными инструментами создания интерактивного трехмерного Web-содержимого.

---

### ПРИМЕЧАНИЕ

Информация о программах Director и 3ds max опубликована на Web-странице [www.macromedia.com/director](http://www.macromedia.com/director).

---

## Cinema 4D

Программа Cinema 4D от компании Maxon Computer ([www.cinema4d.com](http://www.cinema4d.com)) является одной из наиболее известных в мире трехмерного моделирования. Это средство отличается поразительной мощностью, что позволяет ему конкурировать с 3ds max и LightWave 3D.

Интерфейс Cinema 4D легко поддается настройке, благодаря чему пользователи могут создать максимально удобную для себя рабочую среду (рис. 26.5). Программа поддерживает многочисленные системы моделирования, в том числе HyperNURBS (вариант NURBS), моделирование с использованием параметрических примитивов и технологию MetaBalls.



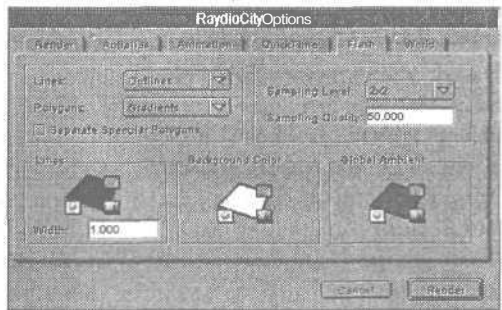
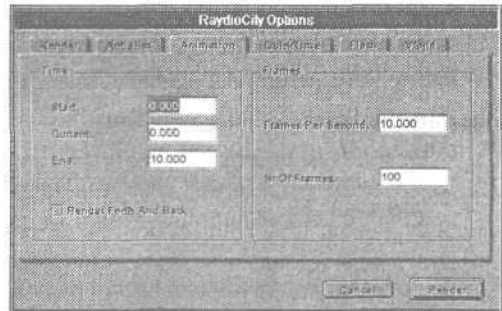
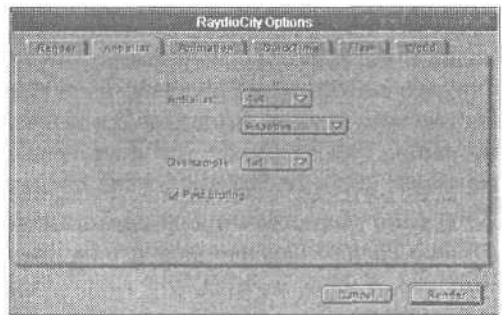
14. Щелкните на вкладке Flash.
15. В списке Lines выберите одно из следующих значений.
  - ▷ None — линии отсутствуют как по краям многоугольников, так и в местах пересечения объектов.
  - > Outlines — отображение линий на пересечении объектов.
  - > Wireframe — представление объекта в виде проволочного каркаса.
16. В списке Polygons определите способ построения многоугольников, изображающих поверхности объектов.
  - > None — многоугольники отсутствуют.
  - ▷ Cartoon — каждый многоугольник экспортируется со сплошной заливкой.
  - > Gradients — в выходном файле многоугольникам будет присвоена градиентная заливка для имитации освещения объектов.
17. Щелкните на образце цвета Lines, чтобы выбрать цвет для линий в итоговом файле.
18. Выберите цвет освещения в области Global Ambient. Если цвет отличается от черного, в экспортированном файле скрытые от света участки модели будут темнее, чем освещенные, и приобретут выбранную окраску.
19. По умолчанию в любом экспортируемом файле цвет фона черный. Чтобы изменить его, воспользуйтесь элементом Background Color.
20. Вы можете задать качество экспортируемого Flash-файла в числовом выражении. Для этой цели предназначено поле Sampling Quality. Чем больше это значение, тем выше качество и значительнее объем файла.
21. Завершив установку параметров, нажмите кнопку Render, которая находится в нижнем правом углу диалогового окна RaydioCity Options.
22. Когда откроется окно Export, введите имя файла в поле Name, укажите, где сохранить файл, и щелкните на кнопке Save.

## Работа с программами, не поддерживающими экспорт в формат SWF

Трехмерные модели использовались на телевидении, в игровой индустрии, кино, архитектуре и промышленном дизайне еще до появления Flash. По этой причине многие мощные программы трехмерного моделирования не обладают функциями экспорта в среду Flash. Большинство программ, поддерживающих экспорт во Flash, включая те, которые мы рассмотрели, предназначены для пользователей с небольшим опытом трехмерного моделирования и, следовательно, далеки от желаемой функциональной завершенности.

Таким образом, для пользователей Flash, создающих трехмерную анимацию, установлены жесткие рамки. Искушенные разработчики трехмерных моделей вряд ли захотят отказаться от любимой программы в пользу примитивного приложения,

3. В списке Output отметьте пункт Flash.
4. В списке Size укажите размер экспортируемого файла. Значения могут быть как стандартными, так и пользовательскими. Во втором случае они вводятся в поля Width и Height. Разумеется, чем значительнее размеры сцены, тем больше объем файла.
5. Флажок Radiosity оставьте отключенным. Он задает дополнительное затенение, которое не действует при экспорте в SWF.
6. Если вы хотите, чтобы тени, используемые в сцене, присутствовали в итоговом файле, установите флажок Shadows.
7. Перейдите на вкладку Antialias, чтобы определить параметры сглаживания.
8. Выберите в области Antialias один из приемов: Adaptive (угловатое изображение) или Oversampling (мягкие переходы в изображении).
9. Чтобы экспортированное изображение получилось слегка расплывчатым, активизируйте флажок Post blurring.
10. Если вы экспортируете анимацию, перейдите на вкладку Animation. Здесь можно установить некоторые параметры анимации в выходном файле.
11. Введите значения длины экспортируемой анимации в поля Start, Current и End. Это числовое выражение информации, которая графически представлена на временной шкале.
12. Задайте частоту смены кадров в поле Frames per Second. Чем больше это значение, тем выше качество фильма. Но, к сожалению, размер файла также заметно увеличивается.
13. В случае необходимости укажите точное количество кадров в поле Nr of Frames. Amorphium Pro автоматически рассчитает частоту кадров, исходя из введенного вами значения и разности между конечным (поле End Time) и начальным (поле Start Time) временем. Например, если Start Time равно 0, End Time — 10, а Number of Frames — 200, частота кадров составит 20.



В главной комнате — Tools — объекты формируются по образцу с помощью различных кистей. В других комнатах, например в FX, осуществляется деформация трехмерных объектов посредством специальных инструментов, таких как Bend (Изгиб), Twirl (Завиток) и т. д.

Интересны инструменты, предоставляемые в комнате Biospheres, которые идеально подходят для создания объектов, похожих на молекулы.

#### ПРИМЕЧАНИЕ —

«Биосферы» можно сравнить с круглыми капельками ртути, которые взаимодействуют различным образом.

Наряду с инструментами моделирования программа Amorphium Pro оснащена мощной системой редактирования материалов и богатой библиотекой текстур, которые можно перетаскивать на поверхность модели. В арсенале Amorphium Pro имеются также инструменты анимации, работающие по распространенной методике, суть которой — ввод ключевых кадров во временную шкалу. Этой технологией легко овладевают даже начинающие пользователи.

Программа Amorphium Pro предоставляет неординарные возможности трассировки лучей и создания освещения. Наконец, Amorphium Pro позволяет экспортировать модели и анимацию во Flash-файлы.



Демонстрационная версия Amorphium Pro находится на прилагаемом компакт-диске.

Как и в предыдущих разделах, мы ограничимся рассмотрением только тех средств Amorphium Pro, которые служат для экспорта моделей и анимации во Flash-файлы.

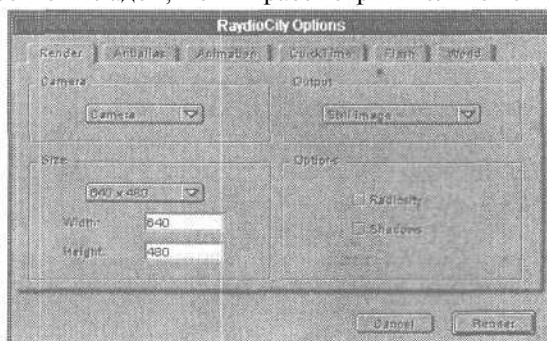
#### ПРИМЕЧАНИЕ

Если вы заинтересовались возможностями Amorphium Pro, почитайте разделы справочной системы, а также посетите Web-страницу компании Electric Image ([www.electricimage.com](http://www.electricimage.com)).

Когда анимация, предназначенная для экспорта в программу Flash, готова, следует задать параметры для выходного файла. Откройте окно RaydioCity Options, нажав кнопку, которая находится в нижней части панели инструментов Composer. Затем выполните щелчок на рабочей области.

В окне вы увидите большое количество вкладок, но мы рассмотрим только те из них, которые связаны с процедурой экспорта.

1. В окне RaydioCity Options перейдите на вкладку Render.
2. В списке Camera выберите камеру, которую надлежит использовать для построения изображения. Возможны такие варианты: Top, Right, Front и Isometric. Допускается «установка» собственных камер.



## Amorphium Pro

На заре трехмерного моделирования специальными программами приходилось управлять на интуитивном уровне. Часто они имели настолько сложный интерфейс, что приводили неискушенных пользователей в замешательство. В результате моделирование с использованием многоугольных каркасов занимало продолжительное время. Кроме того, программы трехмерного моделирования не рассчитаны на специалистов в таких традиционных областях, как скульптура, полиграфия, живопись и керамика. А ведь многие художники видят в мире трехмерного моделирования новую форму выражения своих идей. Насколько желанной для поклонников «осязаемого» искусства является программа, не требующая затрат времени на изучение трехмерного моделирования и громоздких геометрических построений! К счастью, альтернатива инженерному моделированию существует. Это — программа Amorphium Pro от компании Electric Image ([www.electricimage.com](http://www.electricimage.com)). Речь идет не об обычной программе трехмерного моделирования. В Amorphium Pro процесс создания объектов напоминает лепку скульптур из глины. Программа позволяет манипулировать трехмерными моделями, нажимая на них, растягивая, искривляя, трансформируя и раскачивая их. Все изменения происходят в реальном времени. Приложение Amorphium Pro предлагает совершенно новый подход к моделированию трехмерных объектов, что делает его уникальным.

Интересен интерфейс программы, разделенный на отдельные «комнаты» (рис. 26.4), доступ к которым можно получить с помощью меню, расположенного в верхней части главного окна. Каждая комната содержит набор специфических инструментов, предназначенных для решения конкретной задачи.

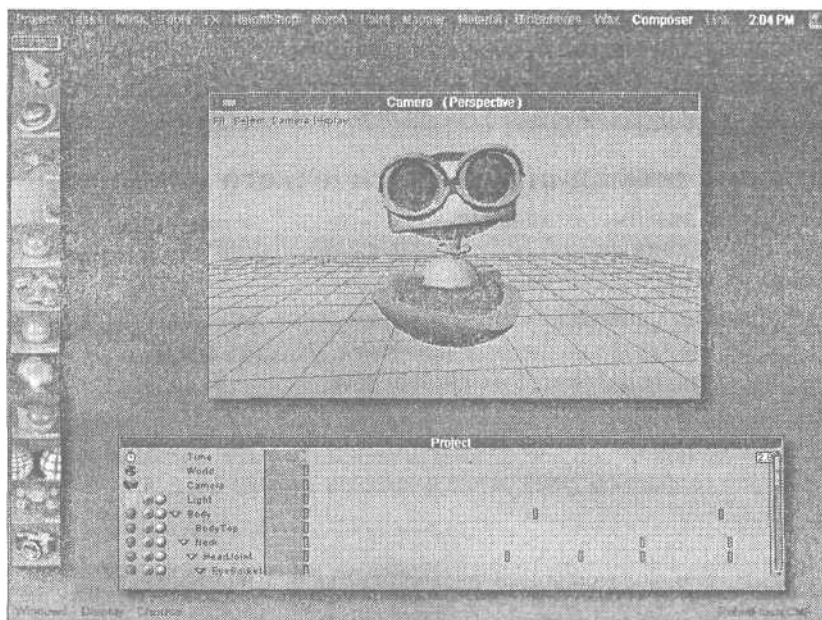


Рис. 26.4. Интерфейс программы Amorphium Pro

## 4. Нажмите одну из кнопок Animation Preview:

- ▷ Play Once — активизация режима однократного воспроизведения анимации при предварительном просмотре.
- ▷ Loop — *бесконечное* воспроизведение анимации в режиме предварительного просмотра.
- > Play Back and Fourth — переход в режим предварительного просмотра, в котором анимация воспроизводится от начала до конца, а затем от *конца* до начала.

Для предварительного просмотра анимации в рабочей области нажмите кнопку Preview Animation. Следует учитывать, что во время предварительного просмотра анимация воспроизводится как совокупность точек, а не так, как она будет выглядеть в SWF-файле.

## Параметры рендеринга

Каким образом будет осуществлен рендеринг, зависит от следующих установок.

- ▶ Render Outlines. Если активизирована эта кнопка, экспортируемая модель будет представлена как *совокупность* линий.
- ▶ Render fills. При нажатии указанной кнопки экспортируемая модель состоит только из заливок.
- ▶ Render lines and fills. Выбор данной кнопки обеспечивает вывод и линий и заливок экспортируемой модели.
- ▶ Perspective. В этом поле указывается, по каким законам перспективы должно строиться результирующее изображение. При вводе значения 1 применяется перспектива, как в объективе фотоаппарата, а при указании значения 0 — изометрическая перспектива (установка по умолчанию).
- ▶ Intersection. На экран выводятся линии пересекающихся поверхностей.
- ▶ Surface Detail. В модели видны все детали поверхности.

## Построение анимации или статического изображения

Подведем итоги: мы импортировали модель в программу Vecta3D, задали для нее стиль, определили способ построения изображения. Теперь можно приступить к формированию SWF-файла.

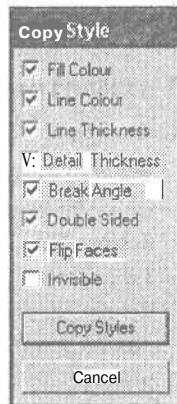
- ▶ Для построения анимации нажмите кнопку Save Animation to Disk. Когда появится окно Save As, введите имя в поле Name, укажите место, в котором *нужно* сохранить файл, и щелкните на кнопке Save.
- ▶ Чтобы сформировать статическое изображение, нажмите кнопку Save Still. В открывшемся окне Save As введите имя в поле Name, укажите место, в котором следует сохранить файл, и щелкните на кнопке Save.

### СОВЕТ

Для предварительного просмотра изображения нажмите кнопку Preview Still, которая расположена в области Single Frame окна программы.

обе стороны многоугольников. К сожалению, при этом значительно увеличивается продолжительность процесса построения.

- ▶ **Flip Faces.** В результате активизации этого флажка все многоугольники выбранного объекта «выворачиваются» лицевыми поверхностями внутрь (и наоборот).
- ▶ **Invisible.** Объекты, для которых активизирован этот флажок, в сформированном SWF-файле не отображаются.
- ▶ **ReRender.** Элементы управления Surface Detail, Flip Faces и Invisible связаны с процессом формирования SWF-файла. Это означает, что их действие проявляется только по выполнении рендеринга, в режиме предварительного просмотра или в случае экспорта файла. Если вы хотите увидеть изменения в окне Styles, нажмите кнопку ReRender.
- ▶ **Copy Styles.** Это средство мгновенного копирования стилей из одного объекта в другой. Выберите объект, стиль которого нужно скопировать, а затем, удерживая нажатой кнопку Ctrl, щелкните на объекте, которому хотите назначить этот стиль, и нажмите кнопку Copy Styles. В появившемся окне укажите стиль, подлежащий копированию, и еще раз нажмите кнопку Copy Styles.
- ▶ **Global Intersection Style Color.** В Vecta3D любой паре пересекающихся линий присваивается один и тот же набор параметров. Чтобы установить их цвет, щелкните на образце Global Intersection Style Color и выберите (или синтезируйте) новый цвет.
- ▶ **Global Intersection Style Thickness.** В поле Thickness определяется толщина пересекающихся линий.



Завершив работу с окном Styles, закройте его. Сразу после этого все установки, выполненные в данном окне, будут применены к модели. Таким образом, нажимать кнопку Apply не обязательно.

## Создание простой анимации

В разделе «Импорт последовательности файлов с трехмерными изображениями» мы создавали анимацию, импортируя последовательность файлов. Кроме того, Vecta3D предоставляет инструменты для создания простой анимации вращения.

1. Загрузите модель в рабочую область.
2. В области Animation окна программы введите значение (в градусах) в поле: Object Rotation или World Rotation. В первом случае объект будет вращаться вокруг соответствующей оси, проходящей через его центр, а во втором — вокруг соответствующей мировой оси (которая, разумеется, проходит через центр вашей рабочей области).
3. Укажите в поле Frames, сколько кадров использовать в цикле вращения. Чем больше кадров в такой анимации, тем медленнее вращается объект.

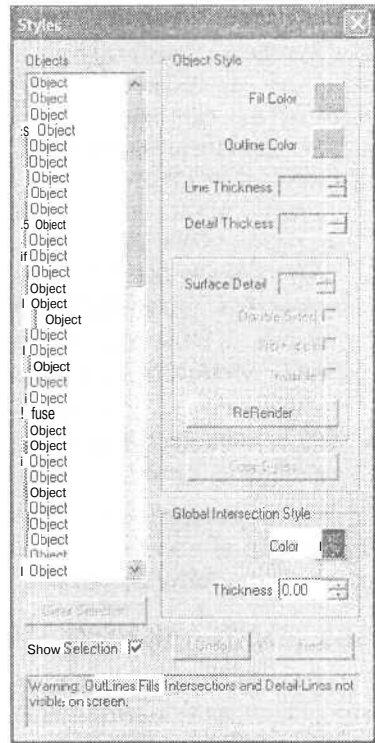
точки вращения позволяет инструмент Edit Registration Center. Если активизировать указанный инструмент, появится красная стрелка — *регистрационный центр*. Перетащите модель в другую позицию. Таким образом вы измените позицию регистрационного центра.

- ▶ **Reset Model Position.** Для восстановления изначальной позиции импортированной модели достаточно нажать кнопку Reset Model Position.
- fr- **Save View.** Благодаря этому инструменту можно сохранить в V3D-файле текущие значения масштаба и ориентации модели.
- ▶ **Load View.** Данный инструмент позволяет применить сохраненные в V3D-файле значения позиции, масштаба и ориентации.

## Работа с окном Styles

Доступ к окну Styles предоставляется при нажатии кнопки Toggle Styles Window в секции View. В этом окне управляют цветом и стилями линий Flash-файла при экспорте.

- ▶ **Objects.** Список Objects занимает левую часть окна Styles. В нем перечисляются отдельные объекты импортированной трехмерной модели. Имена присваиваются им при создании объекта в исходной программе. Выделив один из объектов, вы сможете задать его индивидуальные параметры.
- ▶ **Object Style Fill Color.** Этот элемент предназначен для изменения цвета заливки выбранного объекта. Операция проста: щелкните на образце Fill Color и выберите (или синтезируйте) новый цвет.
- ▶ **Object Style Outline Color.** Данный элемент идентичен предыдущему, но служит для изменения цвета контура.
- ▶ **Line Thickness.** В этом поле определяется толщина контурных линий объекта (в пикселах). Возможны два варианта: ввод значения с клавиатуры и пошаговое изменение значения посредством кнопок со стрелками.
- ▶ **Detail Thickness.** Введя значение (в пикселах) в поле Detail Thickness, вы измените толщину внутренних деталей выбранного объекта.
- ▶ **Surface Detail.** В этом поле указывается, насколько детализированным должно быть изображение поверхности в окончательном Flash-файле. При вводе значения 0 детали не воспроизводятся.
- ▶ **Double Sided.** В большинстве случаев многоугольники, из которых строятся трехмерные объекты, односторонние. При формировании SWF-файла это иногда приводит к ошибкам. Установив флажок Double Sided, вы будете видеть

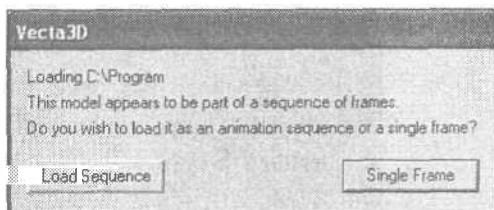


**ПРИМЕЧАНИЕ** —

Каждому импортированному файлу последовательности соответствует отдельный кадр во Flash-анимации.

Далее рассказывается, как осуществляется импорт последовательности файлов.

1. Активируйте команду File ▶ Open.
2. Когда откроется окно Open, найдите последовательность файлов.
3. Выберите первый файл последовательности (например, circle\_00.3ds) и нажмите кнопку Open.
4. В появившемся окне щелкните на кнопке Load Sequence.

**СОВЕТ**

Для импорта отдельного файла последовательности нажмите кнопку Single Frame, а не Load Sequence.

## Управление трехмерной моделью с помощью инструментов просмотра

Позиция и ориентация импортированной модели могут быть изменены. Соответствующие инструменты объединены в группу View и расположены в нижнем левом углу окна программы. Ознакомимся с каждым из них.

- ▶ Spin. Инструмент Spin позволяет вращать модель относительно любой оси при условии, что область выполнения этой операции не ограничена искусственным образом. Вращение осуществляется методом перетаскивания объекта мышью.

**СОВЕТ**

При работе с инструментом Spin для активизации режима масштабирования следует удерживать нажатой клавишу Ctrl.

- ▶ Zoom. С помощью инструмента Zoom вы можете приближать к модели камеру и отдалять ее. Работа с данным инструментом ведется следующим образом. Активируйте инструмент Zoom в области View, установите указатель на модели и перемещайте его. В случае приближения указателя к верхней части модели изображение дается мелким планом, иначе — крупным планом.

**СОВЕТ**

При работе с инструментом Zoom для активизации режима Spin следует удерживать нажатой клавишу Ctrl.

- ▶ Edit Registration Center. Как и во Flash, каждая модель в Vecta3D имеет центральную точку, вокруг которой осуществляется вращение. Изменять позицию



линий, геометрических параметров и качества изображения. Программа обладает набором инструментов для управления позицией модели при экспорте и даже несколькими инструментами анимирования.

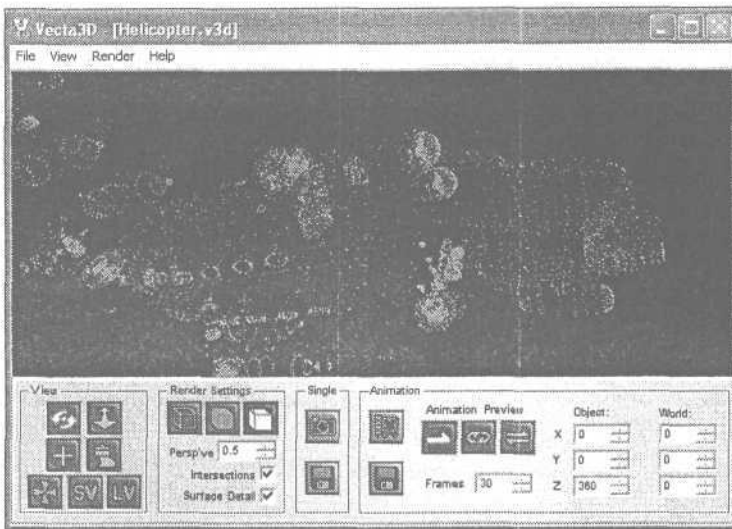


Рис. 26.3. Интерфейс программы Vecta3D

## СОВЕТ

Инструменты анимирования, предлагаемые программой Vecta3D, довольно примитивны. Поэтому для выполнения каких-либо сложных операций обращайтесь к своей программе трехмерного моделирования.

## Импорт 3D-файла

Импорт модели — это первый шаг на пути ее преобразования в SWF-файл.

1. Активизируйте команду File ▶ Open.
2. В окне Open укажите тип файла в списке Files of Type.
3. Выделите имя файла.
4. Нажмите кнопку Open.

Вы, наверное, заметили, что в меню File также есть команда Import. Она отличается от команды Open тем, что позволяет импортировать EPS- и AI-файлы.

## Импорт последовательности файлов с трехмерными изображениями

Хотя инструменты для работы с анимацией, предоставляемые Vecta3D, являются довольно примитивными, вы можете использовать их для экспорта высококачественной сложной анимации в формат SWF. Данный процесс включает такие операции: создание анимации в программе трехмерного моделирования, экспорт ее в виде последовательности файлов и формирование на их основе Flash-фильма с помощью Vecta3D.

Для описания всех возможностей программы Amari 3D 6.1 потребовалась бы целая книга. Поэтому мы поговорим лишь о главном на данный момент — об экспорте модели из Amari 3D в файл SWF.

### СОВЕТ

Если вы заинтересовались возможностями Amari 3D, почитайте разделы справочной системы, а также посетите Web-страницу компании Eovia ([www.eovia.com](http://www.eovia.com)).

Итак, чтобы экспортировать анимацию из Amari 3D в файл SWF, выполните следующие действия.

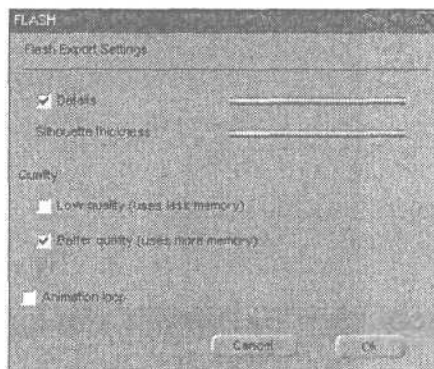
1. Активизируйте команду File ▶ Export ▶ Flash.
2. В окне Flash Export Settings задайте параметры экспорта.

▷ Установите флажок Details и с помощью соседнего ползунка выберите необходимую степень сглаживания контуров.

▷ Посредством ползунка Silhouette Thickness определите толщину контуров объектов.

▷ Воспользовавшись флажками Low Quality и Better Quality, определите качество (и, соответственно, размер) SWF-файла.

▷ Если требуется задать циклическое повторение Flash-анимации, активизируйте флажок Animation Loop.



3. Щелкните на кнопке ОК.

4. В окне Save As укажите место хранения SWF-файла и нажмите кнопку Save.

## Vecta3D

Vecta3D, продукт компании Ideaworks3D ([www.ideaworks3D.com](http://www.ideaworks3D.com)), — это одно из первых средств экспорта трехмерных изображений в среду Flash. Vecta3D представляет собой программу преобразования, которой передаются трехмерные модели, подготовленные в одной из программ моделирования (3ds max или DXF), и результатом работы которой являются Flash-файлы (рис. 26.3). Возможно и обратное преобразование EPS- и AI-файлов в трехмерные объекты. Такой процесс называется *экструзией*.

### ПРИМЕЧАНИЕ

Не стоит беспокоиться из-за того, что Vecta3D импортирует только 3DS- и DXF-файлы. Большинство профессиональных разработчиков трехмерных моделей экспортируют их в формат 3DS или DXF.

Наряду с импортом и экспортом Vecta3D позволяет выполнять такие операции, как изменение цветов трехмерной модели, а также определение *стиля* и *толщины*

трехмерного моделирования, рассчитана на профессионалов именно в этой области. Это средство **намного** более высокого уровня, чем, скажем, Swift 3D. Апарі 3D повсеместно используется для трехмерного моделирования архитекторами, дизайнерами, инженерами, художниками, а также в интерактивной среде, на телевидении и при создании фильмов.

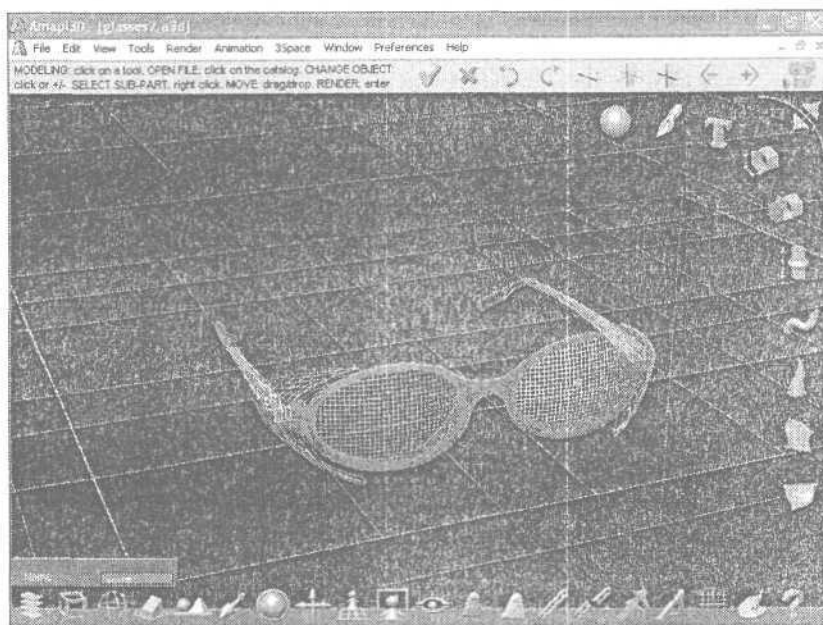


Рис. 26,2. Необычный интерфейс программы Апарі 3D

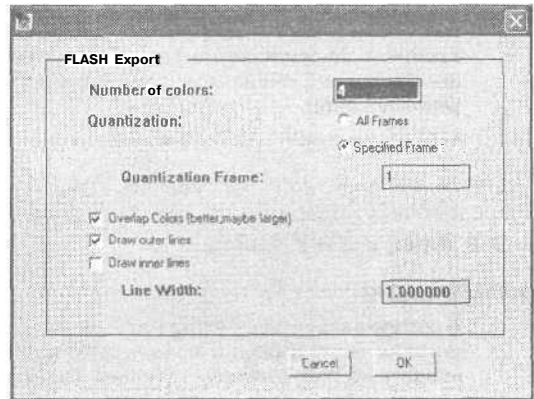
Большим достоинством программы Апарі 3D является набор инструментов, основанных на гибридной технологии, которая позволяет пользователям работать как с NURBS (Nonuniform Rational B-Splines — неравномерные рациональные би-сплайны), так и с многоугольными моделями. Благодаря гибридной технологии инструменты конструирования и моделирования способны адаптироваться к типу обрабатываемого объекта.

К числу наиболее интересных средств Апарі 3D принадлежат такие инструменты моделирования, как двумерные и трехмерные примитивы, инструменты конструирования поверхностей, инструменты для глобальных и локальных деформаций, разрезающие и сглаживающие инструменты, а также уникальные ассистенты рисования в трехмерном пространстве. Апарі 3D — не только программа моделирования трехмерных тел. Она снабжена сложной системой формирования эффектов освещения (называемых *трассировкой лучей*), а также мощным редактором материалов, подобием ретушера. В последней версии Апарі 3D (версия 6.1) появилась возможность SWF-экспорта.



Демонстрационная версия Апарі 3D 6.1 находится на прилагаемом компакт-диске.


- ▷ Область Frame Rate содержит два переключателя, назначение которых – определение частоты смены кадров (единица измерения — кадров в секунду). Вы можете принять текущее стандартное значение (Use Movies frame rate) или ввести собственное в расположенное рядом поле (Use this frame rate).
  - ▷ По умолчанию в разделе Time Span диалогового окна Make Movie указывается фактическая длина фильма. Вы всегда можете изменить ее, введя новое значение в формате *часы:минуты:секунды.кадры*. Этот элемент управления полезен в тех случаях, когда требуется экспортировать только часть анимации.
5. Щелкните на кнопке Flash Settings в нижнем левом углу диалогового окна Make Movie.
  6. В результате откроется окно, где можно произвести следующие установки.
    - ▷ В поле Number of Colors задается число цветов в экспортируемом файле. Ваш выбор должен зависеть от требуемого качества файла. Не забывайте, что при увеличении количества цветов файл увеличивается в объеме.
    - ▷ Активизация флажка Overlap Colors приводит к тому, что сначала Flash закрашивает модель цветом, который наиболее часто используется, а затем наносит цвета в порядке снижения частоты применения. Данный флажок позволяет значительно повысить качество изображения, правда одновременно увеличивается и размер файла.
    - ▷ В случае установки флажка Draw outer lines модель экспортируется с линиями вокруг всего силуэта. Толщина линии указывается в поле Line Width (в пикселах).
    - ▷ При активизации флажка Draw inner lines Flash рисует линии (толщина которых также задается в поле Line Width) вокруг каждого цветового слоя.
  7. Щелкните на кнопке ОК для выхода из окна Flash Export.
  8. Нажмите кнопку ОК в окне Make Movie.
  9. Когда появится окно Save File As, укажите место хранения SWF-файла и нажмите кнопку Save.



## Аmarі 3D

С самого начала программа Amari 3D отличалась нестандартностью. Даже ее интерфейс необычен: инструменты расположены дугой в правой части окна (рис. 26.2). Данная программа, продвижением которой в настоящее время занимается компания Eovia ([www.eovia.com](http://www.eovia.com)), хотя и не похожа ни на одно приложение

В 2001 году с целью расширения граничных возможностей трехмерного моделирования компания Curious Labs ([www.curiouslabs.com](http://www.curiouslabs.com)) выпустила программу Poser Pro Pack. Эта программа, продававшаяся отдельно от Poser 4, была снабжена огромным количеством расширений, улучшений и дополнений, которые значительно повысили ее мощь. Говоря о характеристиках Pro Pack, следует отметить улучшенный интерфейс, дополнения, позволяющие работать совместно с 3ds max и LightWave, макроязык, основанный на языке Python, и улучшенные средства сжатия файлов. Особый интерес представляет функция экспорта моделей и анимации во Flash.

 Демонстрационная версия Poser 4 находится на прилагаемом компакт-диске.

#### ПРИМЕЧАНИЕ

Демонстрационную версию программы Poser Pro Pack можно загрузить с Web-страницы компании Curious Labs ([www.curiouslabs.com/products/proPack/html/proPackDemoDownload.html](http://www.curiouslabs.com/products/proPack/html/proPackDemoDownload.html)).

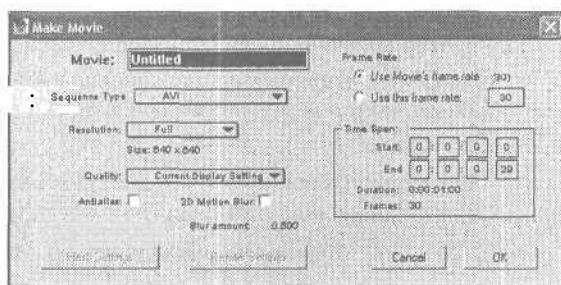
Описать возможности программы Poser в одной главе невозможно, поэтому мы сосредоточим внимание на процессе экспорта трехмерных персонажей, созданных в Poser, в SWF-файлы.

#### ПРИМЕЧАНИЕ

В справочной системе Poser Pro Pack насчитывается 180 страниц, на которых подробно описаны инструменты и возможности программы. Весьма полезную документацию вы найдете на Web-странице компании Curious Labs ([www.curiouslabs.com](http://www.curiouslabs.com)).

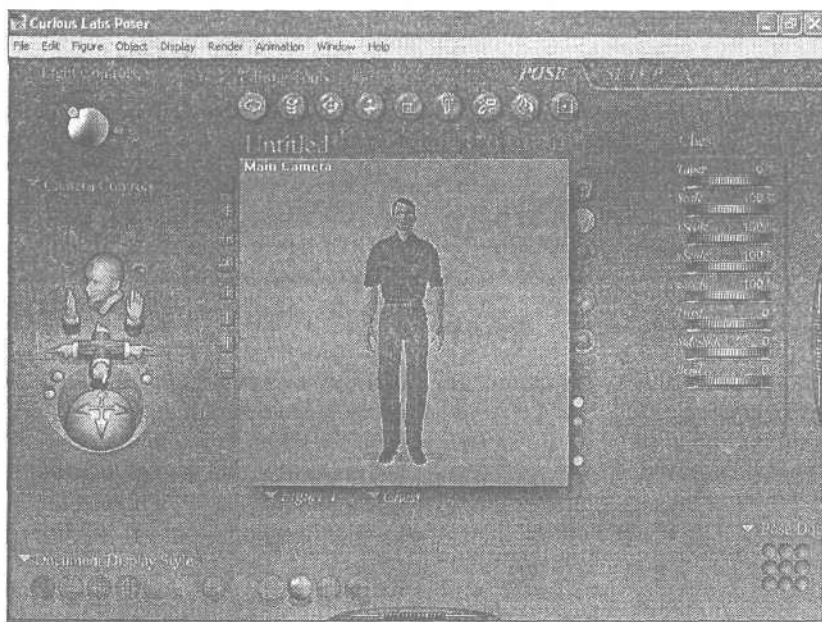
Как и другие программы, Poser Pro легко выполняет экспорт в SWF-файлы. Однако у этого процесса есть ряд особенностей, знание которых необходимо для получения хороших результатов.

1. Откройте файл, который вы хотите экспортировать в программу Flash, и активизируйте команду Animation ▶ Make Movie, вследствие чего откроется окно Make Movie.
2. Введите имя фильма в поле Movie.
3. В списке Sequencer Type выберите элемент Macromedia Flash (.swf).
4. Далее вы можете установить самые разнообразные параметры.
  - > В списке Resolution определяется разрешение (Current, Full, Half или Quarter). Текущее значение отображается под указанным полем.
  - > В списке Quality на выбор предоставляются два варианта: Current Display Settings и Current Render Settings. Для настройки параметров рендеринга выберите Render ▶ Render Options.



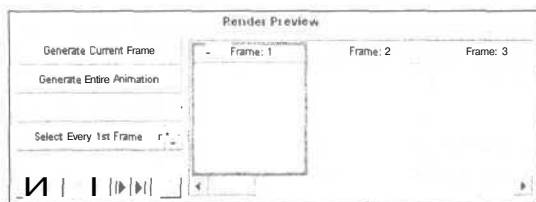
Poser представляет собой удивительное сочетание привлекательного интерфейса и полезных функций. В этой программе вы сможете создавать статические изображения, фильмы и трехмерные объекты, используя компоненты богатейшей коллекции полностью управляемых, сочлененных моделей людей и животных. Элементы из библиотек поз, выражений лица, движений рук, одежды, антуража и анимации могут быть быстро включены в вашу модель благодаря уникальному интерфейсу программы Poser (рис. 26.1). Для управления каждой неделимой частью персонажа разработаны интерактивные элементы, которые обеспечивают значительный контроль над окончательным видом модели. Используя эти элементы, можно решать даже такие задачи, как плавная смена выражений лица и изменение пропорций тела. Рассчитанная на новичков и пользователей со средним уровнем подготовки, программа Poser предоставляет богатый инструментарий для создания изображений человека и животных и их анимирования.

Разработанная компанией MetaCreation, автором замечательных продуктов Ray Dream, Infini-D, Bryce и Painter, программа Poser оказалась под угрозой остаться незамеченной из-за внезапного решения компании отказаться от всех проектов, связанных с созданием программ обработки двух- и трехмерной графики. К счастью огромного количества пользователей Poser, изначальная команда разработчиков решила заняться продвижением этой программы. Как это обычно происходит при таких обстоятельствах в компьютерном мире, была сформирована новая компания, получившая название Curious Labs, которая занималась реализацией Poser. Вскоре увидела свет усовершенствованная версия Poser — 4.03. Программа стала более удобной в использовании и стабильной в работе.



**Рис. 26.1.** Уникальный интерфейс программы Poser является весьма удобным и эргономичным.

15. Тип обводки можно выбрать в списке Edge Type:
  - > Outlines — сплошной линией обводятся края всех объектов.
  - > Entire Mesh — сплошной линией обводятся края всех многоугольников в составе каждого трехмерного объекта, что создает эффект каркаса.
16. Если края задних граней трехмерных объектов должны быть видимыми, установите флажок Include Hidden Edges.
17. В результате активизации флажка Include Detail Edges становится доступным поле Detail Edge Angle. При необходимости укажите в нем количество внутренних краев, которые будут отображаться в экспортированном SWF-файле.
18. Выберите толщину линии в списке Line Weight
19. Для определения цвета линии выполните двойной щелчок в поле Line Color, вследствие чего откроется окно Color Picker, где можно выбрать или создать нужный цвет.
20. Перед экспортом необходимо выполнить рендеринг анимации. Чтобы произвести указанную операцию для всей анимации, нажмите кнопку Generate Entire Animation. Если же действие этой операции должно распространяться на отдельный кадр, выберите его на панели Render Preview и щелкните на кнопке Generate Current Frame.



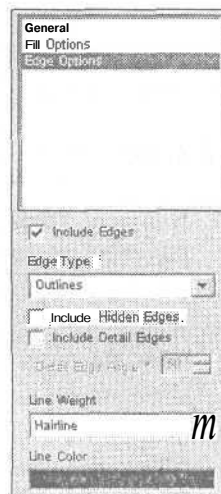
21. По завершении рендеринга нажмите кнопку Export Entire Animation, чтобы экспортировать анимацию целиком, или кнопку Export Current Frame, чтобы экспортировать отдельный кадр, выбранный на панели Render Preview.
22. В диалоговом окне Export Vector File укажите место хранения SWF-файла и его имя, после чего щелкните на кнопке Save.



## Poser и Poser Pro Pack

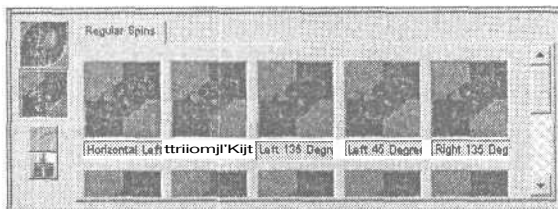
Создание реалистичных персонажей — одна из сложнейших задач трехмерного моделирования. Будь то человек, собака или трехголовый пришелец с планеты Хузах, моделирование органических объектов является невероятно сложным. Вы не только должны хорошо знать программу моделирования, с которой работаете, но и располагать информацией об анатомическом строении тела, принципах биомеханики и движения. Без этого невозможно создавать модели на профессиональном уровне. Быстро приобрести соответствующие навыки и знания невозможно. Поэтому хорошо было бы иметь программу, в которой при создании органических моделей не требуются знания в области анатомии и других наук. Такая программа существует. Это — Poser от компании Curious Labs.

7. При желании установите флажок **Combine Edges and Fills**. В результате контур и заливка объекта будут восприниматься как единое целое (что напоминает результат выполнения операции группировки). Если не сделать этого, экспортированный SWF-файл будет содержать объекты, контурами и заливками которых можно управлять независимо.
8. Теперь выберите группу параметров **Fill Options** на панели **Property**.
9. Для того чтобы объекты в экспортированном SWF-файле имели заливку, активизируйте флажок **Fill Object**.
10. Выберите тип заливки в списке **Fill Type**. Нужно учитывать, что вместе с количеством цветов увеличивается и размер SWF-файла. От представленных в этом списке параметров зависит, как заливка будет выглядеть в экспортированном SWF-файле. Ознакомимся с ними.
  - ▷ **Cartoon Single Color Fill**. Весь объект окрашивается одним цветом.
  - ▷ **Cartoon Average Color Fill**. Многоугольники, образующие поверхность, в экспортированном SWF-файле получают одинаковый цвет.
  - ▷ **Cartoon Two Color Fill**. Как и в предыдущем случае, отдельные многоугольники раскрашиваются одинаково, если они принадлежат одной поверхности, но теперь уже двумя цветами.
  - ▷ **Cartoon Four Color Fill**. То же, что и в предыдущем случае, но при раскрашивании применяются четыре цвета.
  - ▷ **Cartoon Full Color Fill**. Каждому многоугольнику присваивается отдельный цвет.
  - ▷ **Area Gradient Shading**. Программа ищет группы многоугольников, расположенных на одной поверхности, и заполняет их радиальным градиентом. Это создает эффект отражения света.
  - ▷ **Area Gradient Shading**. В результате выбора этого значения становится доступным флажок **Include Specular Highlights**. Если активизировать его, программа будет искать поверхности, расположенные перпендикулярно к источнику света, и назначать им более светлый цвет. Будьте осторожны! Этот процесс сопровождается значительным увеличением размера файла.
11. Если вы хотите, чтобы экспортированная сцена содержала тени, установите флажок **Include Shadows**.
12. В списке **Specular / Shadow Light** выберите источник света, с помощью которого будут созданы тени в экспортированном файле. К сожалению, с этой целью можно использовать только один источник.
13. Теперь выберите на панели **Property** группу параметров **Edge Options**.
14. Если объекты в экспортированном SWF-файле должны иметь обводку, активизируйте флажок **Include Edges**.





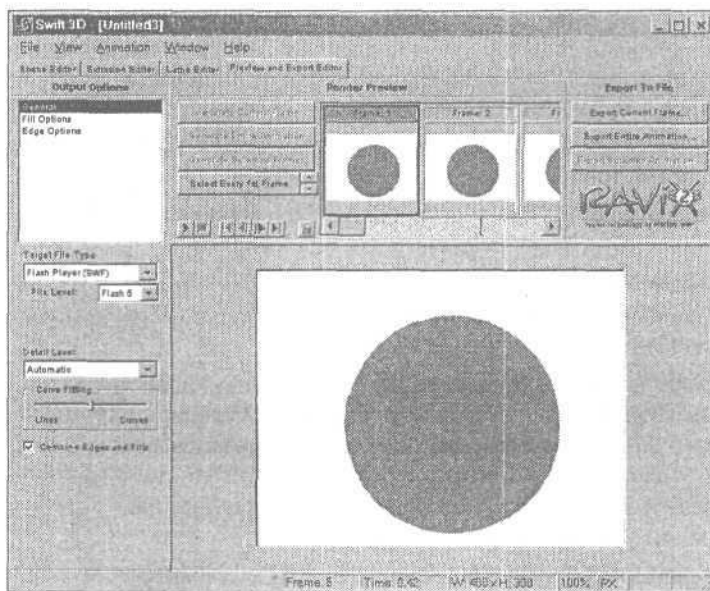
3. В результате откроется палитра Animation, содержащая единственную вкладку Regular Spins. Выберите на ней одну из анимационных заготовок.
4. Перетащите анимацию из палитры на выбранный объект.
5. Для тестирования анимации нажмите кнопку Play Animation, которая находится ниже временной шкалы.



## Экспорт в SWF

Наша модель готова: мы выбрали для нее цветное решение и анимировали. Следующий шаг — экспорт в формат SWF.

1. Перейдите на вкладку Preview and Export Editor окна программы, щелкнув на соответствующей закладке.



2. Выберите группу параметров General на панели Property.
3. В списке Target File Type задайте формат файла, в который должен быть экспортирован проект. В нашем случае — это Flash Player (SWF).
4. Укажите версию Flash для SWF-файла в списке File Level.
5. Выберите уровень детализации для экспортируемого файла в списке Detail Level. Помните: чем выше уровень детализации, тем больше размер файла. Соответственно, чем ниже этот уровень, тем быстрее формируется SWF-файл.
6. Определите, какую позицию должен занимать ползунок на регуляторе Curve Fitting. При смещении ползунка к метке Curves программа сглаживает линии объектов.

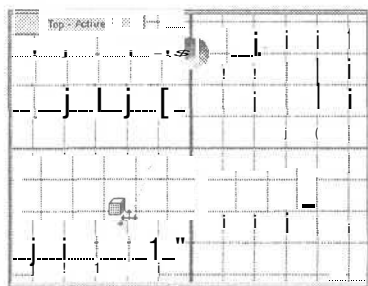
Удалить источник света со сцены несложно: достаточно выделить его и нажать клавишу Delete.



### Управление позицией источника света

Управлять позицией источника света легко. Если источник света связан с трекболом, надлежит сначала выделить источник на трекболе, а затем перемещать сам трекбол. Как и при управлении объектами, вы можете использовать кнопки Lock Axis, расположенные слева от трекбола, чтобы ограничить область распространения света. Наверное, вы уже обратили внимание на то, что в процессе перемещения трекбола, регулирующего освещение, сцена моментально меняется, чтобы соответствовать новой схеме освещения.

Если источник света добавлен непосредственно в сцену, ваши действия сводятся к перемещению символа источника в новую позицию при нажатой левой кнопке мыши (вид указателя при этом меняется).



В какой бы позиции вы ни разместили источник света, связанный с точкой вращения объекта, он по-прежнему останется связанным с указанной точкой линией, вдоль которой направляется свет.

### Создание анимации

Swift 3D является не только программой трехмерного моделирования, но и анимационной программой. Она позволяет создавать настоящие трехмерные анимационные ролики и экспортировать их в формат SWF.

Как и в среде Flash, при подготовке анимации с помощью Swift 3D используются ключевые и текущие кадры. Правда приемы работы с ними специфичны для каждой программы. В Swift 3D существует библиотека с заготовками анимации, которые можно назначать объектам сцены. Такие *перетаскиваемые анимации* являются простым средством создания сложных анимационных эффектов. Давайте проверим сказанное на практике.

1. Выделите анимируемый объект.
2. Нажмите кнопку Show Animation, расположенную в нижнем правом углу окна программы (под кнопкой Show Materials).

**ПРИМЕЧАНИЕ-**

При работе с текстом или сложными импортированными моделями материал требуется назначать каждой части объекта в **отдельности**. В случае текста это означает, что операцию нужно выполнить как для лицевой стороны текста, так и для скоса.

**Работа со светом**

Одним из важнейших параметров любой трехмерной модели является освещение. При правильном освещении даже заурядная модель приобретает привлекательный облик, хотя, к сожалению, возможности работы со светом в Swift 3D ограничены.

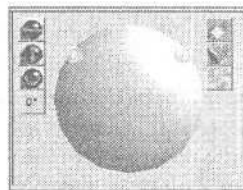
В Swift 3D различают два вида источников света: световую точку и световое пятно. Световая точка — это световой фрагмент компьютерной графики с одинаковым излучением во всех направлениях. Световым пятном называется источник, излучающий конический пучок световых лучей. Источники могут быть добавлены в сцену непосредственно или с помощью трекбола.

**Добавление и удаление источников света с помощью трекбола**

Трекбол, предназначенный для управления источниками света, размещен в нижней части окна программы, в области Select/Move Light. Справа от трекбола находятся кнопки, которые позволяют, в частности, добавлять в сцену любое количество источников — как световых точек (кнопка Create Trackball Point Light), так и световых пятен (кнопка Create Trackball Spot Light).

Когда вы добавляете источник света, он появляется на трекболе.

Для удаления источника света выберите его на трекболе и нажмите кнопку Remove Trackball Light, которая расположена под кнопками добавления источников.

**Источники света в сцене**

Источник света может быть добавлен непосредственно в сцену. Для этого достаточно нажать одну из следующих кнопок: Create Free Point Light, Create Target Point Light, Create Free Spot Light, Create Target Spot Light.

**ПРИМЕЧАНИЕ**

В сцене источник света обозначается маленьким желтым символом.

Источники света Free Point Light и Free Spot Light, как и обычные объекты, можно выделять и подвергать различным манипуляциям, используя предназначенный для вращения трекбол.

**ПРИМЕЧАНИЕ**

Выбранный источник света приобретает красный цвет.

Действие источников Target Point Light и Target Spot Light всегда направлено на точку вращения объекта, который был выделен до их вставки в сцену.

6. Нажмите кнопку Apply, чтобы выполненные установки стали действительными, а текстовый объект в окне программы принял новый облик.
7. Чтобы получить доступ к элементам, предназначенным для управления параметрами объемности букв, перейдите на панель Property и щелкните на заголовке Bevel.
8. В списке Style выберите стиль объемности (прежде чем сделать окончательный выбор, поэкспериментируйте с различными вариантами).
9. В поле Depth введите значение величины скоса символов. Кнопки со стрелками позволяют увеличивать и уменьшать это значение.
10. Выберите один из переключателей в группе Face, определяющих вид скоса.
11. Определите позицию ползунка на регуляторе Smoothness. От нее зависит, как Swift 3D будет рисовать контуры букв. Чем ближе находится ползунок к метке Course, тем более угловатым получается текст. Перемещение ползунка к метке Smooth приводит к сглаживанию контуров.

#### ПРИМЕЧАНИЕ

Повышение уровня сглаживания сопровождается увеличением размера файла.

12. Посредством регулятора Mesh Quality подберите подходящий размер ячеек сетки трехмерных объектов.
13. Выберите на панели Property группу параметров Sizing.
14. Введите значения в поля Height (Высота), Width (Ширина) и Depth (Глубина), чтобы придать требуемый размер трехмерному тексту.
15. Укажите в поле Inter Character интервал между символами.
16. В случае необходимости задайте междустрочный интервал в поле Inter Line.

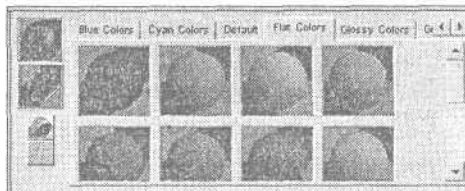
### Работа с заготовленными материалами

Ограничения, с которыми связано создание пользовательских материалов в Swift 3D, отчасти компенсируются возможностью использования заготовок из библиотеки.

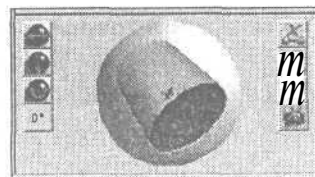
В отличие от других программ трехмерного моделирования, предоставляющих доступ к всевозможным инструментам для создания и смешивания материалов, специфика Flash (особенно при работе с растровыми изображениями) ограничивает типы материалов, которые вы можете присваивать объектам в Swift 3D.

Рассмотрим, как применить готовый материал.

1. Убедитесь, что выбран нужный объект.
2. Нажмите кнопку Show Materials, расположенную в нижнем правом углу окна программы, чтобы загрузить палитру Material Palette.
3. Щелкните на закладке, соответствующей нужному набору образцов.
4. Перетащите материал из палитры на выбранный объект.



## Использование трекбола



Добавьте в сцену какой-либо объект. Далее мы поработаем над его ориентацией. Для этой цели предназначен трекбол (манипулятор в виде шара), расположенный в нижнем левом углу окна программы (в области Rotate Selected).

Трекбол неактивен до тех пор, пока вы не выберете объект в демонстрационном окне. Выделенный объект появляется внутри трекбола. В процессе вращения с помощью мыши вместе с трекболом вращается и объект.

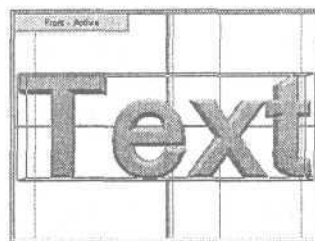
Вы можете ограничить область вращения объекта посредством кнопок Lock Axis, находящихся слева от трекбола. Возможны такие варианты:

- ▶ Lock Horizontal — вращение объекта осуществляется относительно его горизонтальной оси;
- ▶ Lock Vertical — вращение объекта осуществляется относительно его вертикальной оси;
- ▶ Lock Spin — вращение объекта осуществляется только по часовой стрелке или против нее.

В дополнение к указанным предусмотрена возможность задать шаг вращения трекбола (в градусах). Она реализуется посредством кнопки Rotation Increment, в результате активизации которой появляется меню для выбора подходящего значения.

## Работа с трехмерным текстом

Текст является мощным средством передачи информации, поэтому мы рассмотрим каждый шаг процедуры его создания и редактирования. Но прежде отметим, что в Swift 3D текст рассматривается как объект, подобный базовым геометрическим фигурам, с которыми вы уже познакомились.



1. Создайте новый документ и щелкните на кнопке Create Text. Программа создаст текстовый объект в соответствии с установками, принятыми по умолчанию.

### ПРИМЕЧАНИЕ

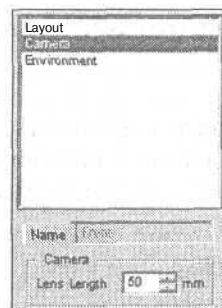
Как уже неоднократно отмечалось, программа предоставляет доступ к свойствам только выделенного объекта. Поэтому удостоверьтесь, что объект, с которым планируете работать, выделен. Для выделения объекта достаточно выполнить щелчок на нем.

2. Выберите группу параметров Text на панели Property.
3. Выберите шрифт в списке Font.
4. Введите нужный текст в поле Text.
5. Задайте подходящий способ выравнивания, щелкнув на соответствующей кнопке.

### Группа параметров Camera

Элементы управления, принадлежащие к категории Camera, служат для настройки параметров выбранной камеры.

- ▶ В поле Name отображается имя текущей камеры. Здесь вы можете изменить его.
- ▶ В поле Lens Length указывается фокусное расстояние объектива (в миллиметрах). При вводе больших значений в это поле объекты отображаются крупным планом, а в случае указания маленьких значений они кажутся удаленными от наблюдателя.



### ПРИМЕЧАНИЕ

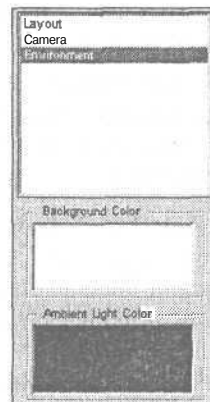
Как и в настоящей камере, если фокусное расстояние объектива очень мало, изображение искажается.

### Группа параметров Environment

Группа Environment включает два очень важных параметра: Background Color и Ambient Light Color.

В области Background Color определяется цвет фона изображения. Изменить цветовое решение фона несложно. Выполните двойной щелчок на образце текущего цвета. В результате откроется окно Color Picker, где и осуществляется выбор цвета для фона. Укажите цвет и подтвердите установку щелчком на кнопке Apply.

Эффект отражения света от всех объектов в заданной области принято называть *окружающим светом*. Чтобы изменить цвет окружающего света, выполните двойной щелчок в области Ambient Light Color, вследствие чего появится окно Color Picker, выберите новый цвет и нажмите кнопку Apply.



### Добавление базовых геометрических объектов

Трехмерное моделирование в Swift 3D часто связано с добавлением базовых геометрических фигур и управлением ими. Производить настройку параметров фигуры (размера, положения и т. д.) можно сразу после ее добавления в сцену.

Вставить базовый объект очень легко: достаточно щелкнуть на кнопке с его изображением. Фигура появится на пересечении осей X, Y и Z.



## Панель Property программы Swift 3D

Панель Property расположена вдоль левой границы окна программы. На ней сосредоточены основные инструменты Swift 3D. С их помощью вы сможете получить доступ ко всем свойствам сцены, над которой работаете. Свойства каждого вновь добавленного объекта сразу фиксируются на панели Property. Для получения доступа к ним надлежит щелкнуть на соответствующем пункте в поле списка.

### Группа параметров Layout

Основное предназначение элементов управления, относящихся к категории Layout, — управление визуальными характеристиками сцены.

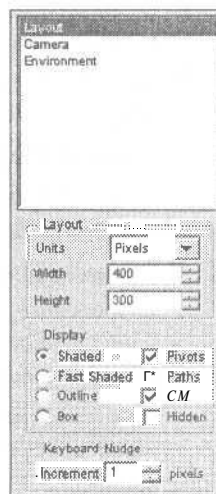
Щелкните на пункте Layout в поле списка на панели Property. В результате вам будет предоставлен доступ к трем группам элементов управления: Layout, Display и Keyboard Nudge. Давайте познакомимся с ними.

- ▶ Layout. Элементы управления, входящие в состав этой группы, позволяют управлять размерами проекта. Процесс регулирования размеров прост: прежде всего в списке Units выбирается единица измерения, после чего в поля Height и Width вводятся значения высоты и ширины.
- ▶ Display. Элементы управления, расположенные в этой области, предназначены для управления отображением объектов в окне программы. Остановимся на каждом элементе. Четыре переключателя позволяют задать, в каком виде выводить объекты.

- > Shaded — объекты отображаются в том виде, в котором они будут представлены после экспорта в формат SWF: с воспроизведением их структуры и имитацией подсветки.
- > Fast Shaded — объекты отображаются так же, как и в предыдущем случае, но без градиентных заливок.
- ▷ Outline — выводятся каркасы объектов.
- > Box — каждый объект отображается в виде трехмерного блока.

Далее следует группа флажков, результаты активизации которых описаны ниже.

- > Pivots — отображается точка центра вращения, которая является регистрационной точкой трехмерного объекта.
- > Path — пути движения в сцене обозначаются пурпурными линиями.
- > Grid — на экран выводится координатная сетка.
- > Hidden — объекты при воспроизведении сцены не отображаются.
- ▶ Keyboard Nudge. В этой области определяется дистанция (в пикселах), на которую сместится выбранный объект, когда пользователь нажмет одну из клавиш со стрелками.



недостает этих возможностей! Важно также то, что программа Swift 3D способна функционировать на платформе как Macintosh, так и Windows. Хотя для Swift 3D характерна совместимость с несколькими операционными средами, сферой трехмерного моделирования традиционно являются Windows и Unix. До недавнего времени пользователи Macintosh оставались в стороне от мира трехмерного моделирования, в связи с чем в настоящее время межплатформенное приложение Swift 3D представляет для них особый интерес.

Было бы заблуждением полагать, что программа Swift 3D может все. Ее сравнение с такими средствами, как Maya, 3ds max, LightWave 3D и Cinema 4D, является неуместным. Возможности Swift 3D по созданию сложных трехмерных моделей весьма ограничены. Если вы заинтересованы в использовании многоугольных моделей и желаете поэкспериментировать с ними, Swift 3D вам не подходит. Это не означает, что данная программа бесполезна. Она легка в изучении, в отличие от многих средств трехмерного моделирования, освоение которых требует нескольких лет. Те, кто уже работал с более мощными программами, могут использовать Swift 3D в качестве промежуточного фильтра, позволяющего задать необходимые для Flash параметры макета, импортированного из программы 3ds max, перед экспортом в формат SWF.

Процессу создания трехмерных изображений в программе Swift 3D посвящен следующий раздел. Следует отметить, что, хотя Swift 3D не принадлежит к разряду мощнейших программ трехмерного моделирования, она обладает широким кругом возможностей, что не позволяет исчерпывающе описать ее в рамках настоящей книги. Для получения полной информации о программе Swift 3D следует обратиться к ее сопроводительной документации.



Демонстрационную версию Swift 3D вы найдете на прилагаемом компакт-диске.

## Создание проекта Swift 3D

Создать новый проект Swift 3D можно несколькими способами.

- ▶ Активизируйте команду File ▶ New.
- ▶ Нажмите кнопку New в левом верхнем углу окна программы.
- ▶ Для открытия существующего документа нажмите кнопку Open в левом верхнем углу окна программы. В диалоговом окне Open File найдите и выделите требуемый файл.
- ▶ Чтобы импортировать файл Encapsulated PostScript (EP) или Adobe Illustrator (AI), активизируйте команду File ▶ Import. В окне Import File найдите и выделите требуемый файл.

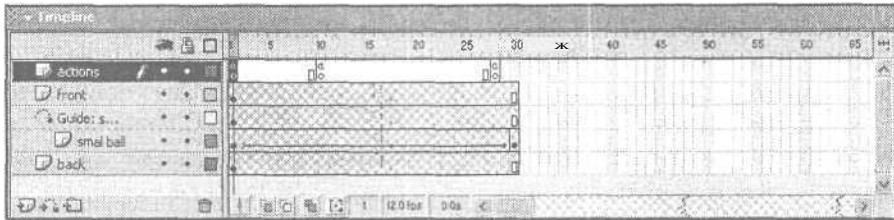
### ПРИМЕЧАНИЕ —

Любое изображение Encapsulated PostScript или Adobe Illustrator при импорте в формат Swift 3D преобразовывается в трехмерный объект, в то время как его оригинал остается двумерным. В процессе импортирования любой предварительно заданный цвет будет утрачен.

- ▶ Если требуется создать новый документ на основе файла 3ds max (.3ds), вызовите команду File ▶ New from 3DS.



31. Повторите процедуру, описанную в пунктах 24 и 25, или скопируйте соответствующий сценарий. Таким образом вы укажете Flash сделать видимой сферу front и скрыть символ back. Далее показано, как в итоге должна выглядеть временная шкала.



В созданной анимации маленькая сфера вращается вокруг большой сферы. Оба объекта являются псевдотрехмерными. Протестируйте фильм с помощью команды Control ▶ Test Movie. Выполнив упражнение из этого раздела, вы получили базовые знания, необходимые для перехода к созданию более сложных эффектов. Попробуйте добавить еще несколько орбит со сферами, чтобы смоделировать атом, или создайте анимацию, в которой движется несколько сфер.

## Экспорт трехмерных изображений в формат SWF

Когда Flash и интегрированные средства по разработке трехмерной графики завоевали некоторую популярность, несколько проницательных компаний, специализирующихся на разработке программного обеспечения трехмерного моделирования, ухватились за новую идею и внедрили в свою продукцию возможность экспорта в формат SWF. Это стало вехой в истории развития Flash. Теперь для воспроизведения трехмерного пространства лишь в крайних случаях приходится прибегать к хитроумным трюкам. Как правило, алгоритм таков: вы создаете трехмерную модель или анимацию (которые экспортируются в формат SWF) и импортируете ее непосредственно в приложение Flash.

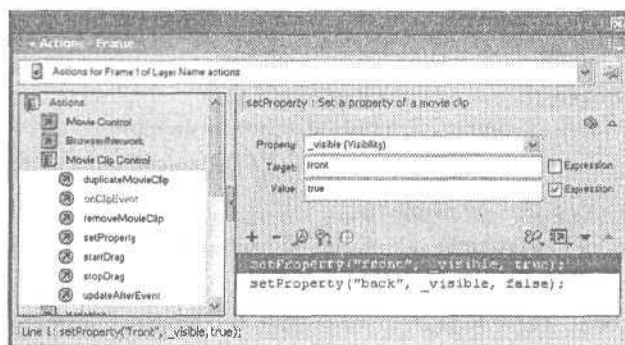
В следующем разделе вам будут представлены программы трехмерного моделирования, позволяющие получать на выходе файлы в формате SWF: Swift 3D, Amorphium Pro, Poser Pro Pack, Amapr 3D и Vecta 3D.

### Swift 3D

Программа Swift 3D от компании Electric Rain ([www.erain.com](http://www.erain.com)) в числе первых поддерживала экспорт трехмерных моделей в формат SWF. Хотя это приложение нельзя сравнивать со многими мощными программами трехмерного моделирования, оно обладает обширным набором полезных функций, предназначенных, в частности, для прямого импорта из форматов 3DS (3D Studio Max), EPS и AI, а также моделирования сложного освещения. Кроме того, Swift 3D предоставляет в распоряжение разработчиков шаблоны анимации, которые позволяют создавать и оптимизировать трехмерные модели для Flash. Многим программам

Actions будет выведена в режиме Expert Mode, нажмите комбинацию клавиш Ctrl+Shift+N для перехода в режим Normal Mode.

24. На панели ActionScript выберите категорию Actions ▶ Movie Clip Control. Выполните двойной щелчок на действии `setProperty()`, чтобы добавить его в окно ActionScript.
25. В списке Property укажите свойство `_visible`. В поле Target введите имя клипа `front`, а в поле Value — значение `true`. Это послужит указанием Flash показать символ `front`. Удостоверьтесь в том, что рядом с полем Value установлен флажок Expression.
26. Повторите шаг 25, но теперь в поле Target введите значение `back`, а в поле Value — `false`, что заставит Flash скрыть символ `back`. Далее показано, каким должно быть результирующее изображение в окне Actions.



Если не указать в поле Target имя, введенное в поле Name панели свойств (шаг 22), макрокоманда не будет работать.

27. Создайте сценарий, который скрывает сферу `front` и выводит вместо нее символ `back`, вследствие чего складывается впечатление, как будто орбитальный шар проходит перед большой сферой. Сначала выберите кадр 10 слоя actions (или кадр, в котором расстояние между сферами на первой половине траектории становится максимальным).
28. Активизируйте команду Insert ▶ Keyframe.
29. Повторите шаги 24— 26, поменяв местами значения `true` и `false`.

### СОВЕТ

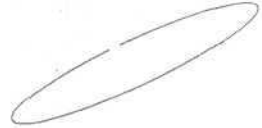
Можно вырезать и вставить текст сценария, созданный на этапах 24—25, и поменять в нем местами значения `true` и `false`.

30. Следующий сценарий должен скрывать символ `back` и выводить клип `front`. Будет создана видимость, что маленький шар проходит за большой сферой и заканчивает свое движение по орбите. Выделите кадр 28 (или кадр, в котором расстояние между сферами на второй половине траектории становится максимальным), и активизируйте команду Insert ▶ Keyframe.

14. Теперь займемся траекторией движения маленького шара. Отмените отображение слоев `back` и `front`, щелкнув на соответствующих им кнопках `Show/Hide` в столбце с изображением глаза.

После завершения настройки траектории движения не забудьте возобновить отображение слоев на экране, повторив щелчки на указанных кнопках.

15. С помощью инструмента `Zoom` максимально увеличьте траекторию движения (сжатый овал позади большой сферы). Выделите небольшой фрагмент траектории посредством инструмента `Lasso` и удалите его. После удаления восстановите исходный масштаб.



В результате траектория движения не будет больше замкнутой, в ней появится разрыв.

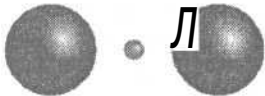
16. Следующий шаг — привязка маленькой сферы к траектории движения. Щелкните на первом ключевом кадре слоя `small ball` и перетащите символ `small ball` из библиотеки в область действия. Открыв окно `Library` (`Window` ▶ `Library`), перетащите из библиотеки в область действия символ `small ball` и расположите его на одном из концов траектории движения.
17. Выделите кадр 30 слоя `small ball` и активизируйте команду `Insert` ▶ `Keyframe`.
18. Выберите последний кадр слоя. Перетащите из библиотеки в область действия символ `small ball` и расположите его на другом конце траектории движения.
19. Чтобы задать движение по созданной траектории, выделите первый ключевой кадр слоя `small ball`, перейдите на панель `Properties` (`Window` ▶ `Properties`) и в списке `Tween` выберите пункт `Motion`.
20. Протестируйте анимацию, нажав клавишу `Return/Enter` или активизировав команду `Control` ▶ `Test Movie`. Обратите внимание на то, что на определенном отрезке пути движущийся маленький шар скрывается за символом `front`. В дальнейшем вы будете использовать команды языка `ActionScript` для динамического изменения видимости больших сфер, чтобы создать иллюзию, как будто маленький шар проходит спереди и позади одного и того же объекта.
21. Переключите видимость двух больших сфер.
22. Перейдите на слой `front` и выделите символ `front`. В поле `Name` панели `Properties` (`Window` ▶ `Properties`) введите значение `front`. Повторите процедуру для символа `back` слоя `back`, но на этот раз в поле `Name` укажите `back`. Таким образом вы назначите уникальный идентификатор экземпляру; это важное условие корректной работы команд языка `ActionScript`.

#### ПРИМЕЧАНИЕ

Поскольку наши символы расположены один над другим, удобной будет такая процедура выделения: для выбора символа `front` следует скрыть слой `back`, а чтобы активизировать символ `back`, надлежит отменить отображение слоя `front`.

23. Приступим к работе с языком `ActionScript`. Выберите первый фрейм слоя `actions` и загрузите панель `Actions`, вызвав команду `Window` ▶ `Actions`. Если панель

Итак, давайте перейдем от теории к практике.

1. Создайте новый документ, выбрав в меню File команду New. Откройте окно Document Properties (Modify > Document) и задайте для ширины и высоты области действия значения 600 и 400 соответственно.
2. Используя описанные ранее приемы, нарисуйте три сферы: одну маленькую и две большие. У больших сфер должны совпадать размер и цвет. 
3. Преобразуйте каждую сферу в клип. Имя маленькой сферы, которая будет двигаться по орбите, может быть произвольным. Большим сферам присвойте имена front и back.

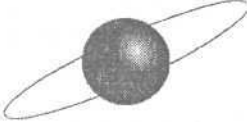
#### ПРИМЕЧАНИЕ

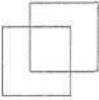
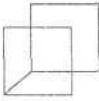

Почему в данном случае следует использовать клипы, а не графические символы? Потому что действие команд языка ActionScript, с которым мы планируем работать, распространяется только на клипы.

4. Если сферы уже преобразованы в символы, удалите исходные объекты.
5. Добавьте три слоя (всего получится четыре). Назовите их так (сверху вниз): actions, front, small ball и back.
6. Выделите первый ключевой кадр слоя front и перетащите символ front из библиотеки в область действия.
7. Загрузите панель Info командой Window ▶ Info. В поле X введите значение 253.3, а в поле Y — значение 153.3.

#### СОВЕТ

Координаты X и Y можно изменить и другим путем — посредством панели свойств (Window ▶ Properties).

8. Выделите кадр 30 на слое front и активизируйте команду Insert ▶ Frame (или нажмите клавишу F5).
9. Выделите кадр 30 на слое back и вызовите команду Insert ▶ Frame.
10. Выделите первый ключевой кадр слоя back и перетащите символ back из библиотеки в область действия.
11. Выделите символ back и загрузите панель Info (Window ▶ Info). В поле X введите значение 253.3, а в поле Y — значение 153.3. Это позволит добиться точного совмещения двух одинаковых клипов (см. шаги 6 и 7), которые будут выступать в качестве одной и той же сферы.
12. Перейдите на слой small ball и добавьте траекторию движения, нажав кнопку Add Motion Guide, которая расположена слева внизу относительно временной шкалы, или активизировав команду Insert ▶ Motion Guide.
13. Выберите первый фрейм слоя Guide: small ball (он появится чуть выше слоя small ball) и нарисуйте вытянутый овал с помощью инструмента Ellipse. Эта фигура будет играть роль орбиты маленького шара. Убедитесь, что у объекта отсутствует заливка. Расположите овал-орбиту над большой сферой. 

3. Убедитесь в том, что квадрат не закрашен. При наличии заливки удалите ее, предварительно выделив инструментом **Arrow**.
  4. Выделите объект с помощью инструмента **Arrow** и активизируйте команду **Modify ▸ Group**.
  5. Вы создали переднюю грань куба. Теперь нужно создать его заднюю грань. Выделите квадрат, скопируйте его в буфер (**Edit ▸ Copy**) и вставьте в область действия (**Edit ▸ Paste**).
  6. Позиционируйте новый квадрат таким образом, чтобы он находился чуть выше и правее первого квадрата. Убедитесь в том, что нижние левые углы квадратов можно соединить диагональной линией. 
  7. Следующий шаг — создание боковых граней куба. Для этого соедините линиями соответствующие углы двух квадратов. В процессе выполнения указанной операции используйте инструмент **Line**. Чтобы облегчить задачу, включите режим привязки к объектам, установив в меню **View** флажок **Snap to Objects**. Как следствие, проводимые линии будут притягиваться к углам квадратов.
  8. Активизируйте инструмент **Line**. Установите указатель в левом нижнем углу передней грани куба. Удерживая нажатой левую кнопку мыши, сместите указатель к левому нижнему углу задней грани куба. Линия будет присоединена к вершине угла. На данном этапе изображение должно выглядеть так, как показано на следующем рисунке справа. 
  9. Повторите описанную процедуру для остальных углов. 
- В результате вы получите каркас куба.

## Анимирование псевдотрехмерных объектов

Псевдотрехмерные объекты можно анимировать. Самое главное при этом — создать иллюзию глубины. Существует несколько приемов, позволяющих добиться указанного эффекта. Основной прием заключается в создании анимации, в которой подобные объекты двигаются перед другим объектом и за ним, за счет чего достигается нужный эффект.



В этом разделе вам представится возможность попрактиковаться в анимировании псевдотрехмерных объектов. Мы создадим простую анимацию, в которой маленькая сфера вращается вокруг большой сферы. В процессе решения задачи будут использоваться средства языка **ActionScript**, в частности действие **setProperty()**.



Завершенная версия этого проекта (в форматах **FLA** и **SWF**) находится в каталоге **Chapter 26** на прилагаемом компакт-диске.



### ПРИМЕЧАНИЕ

Если у вас нет опыта работы с языком **ActionScript**, внимательно изучите четвертую и пятую части этой книги.

11. Убедитесь, что заливка сферы не выделена.
12. Активизируйте инструмент Fill Transform и щелкните на заливке сферы. В результате появится рамка выделения с маркерами. (Если вы не помните, как: управлять заливкой градиента, обратитесь к главе 4.)
13. Для изменения позиции точки отражения света установите указатель на центральном маркере. Обратите внимание на то, что вид указателя изменился. Перетащите маркер вправо и вверх. Итоговое изображение представлено справа. 
14. Установите указатель на *квадратном* маркере рамки выделения радиального градиента. Вид указателя снова изменится. Разместите его немного ближе к центру, вследствие чего градиент внешне несколько приблизится к линейному. 

### СОВЕТ

Если в процессе редактирования градиент «притягивается» к границам объекта, активизируйте команду View ▶ Snap to Objects.

15. Поверните градиент таким образом, чтобы казалось, что источник света находится над сферой справа. С этой целью выделите заливку с помощью инструмента Transform Fill и установите указатель на верхнем маркере. Сместите его так, чтобы область отражения оказалась повернутой на угол 45 градусов. 
16. Переместите центральный маркер на столько, чтобы границы градиента охватывали сферу. 

Поздравляем! Используя простой круг и радиальный градиент, вы создали изображение трехмерного объекта — сферы.

Сохраните сферу, она пригодится в процессе изучения раздела «Анимирование псевдотрехмерных объектов».

### ПРИМЕЧАНИЕ

Варьирование параметров градиента является удобным способом видоизменения объектов.

## Каркас куба

Упражнение, предлагаемое в настоящем разделе, демонстрирует, как можно создать каркас объекта, в частности куба.

### ПРИМЕЧАНИЕ

Термин *каркас* взят из традиционного моделирования трехмерных объектов. По существу это режим, в котором объекты представлены только линиями, вследствие чего создается впечатление, что они сделаны из проволоки.

1. Откройте новый документ, вызвав команду File ▶ New.
2. Нарисуйте квадрат с помощью инструмента Rectangle. Для получения правильной фигуры в процессе черчения удерживайте нажатой клавишу Shift.

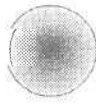
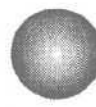
Не следует недооценивать описываемые приемы — овладев ими, вы сделаете первый шаг в мир трехмерной графики. Возможность попрактиковаться мы предоставим вам немедленно. Следующий раздел содержит практическое упражнение, цель которого — получить трехмерную анимацию.

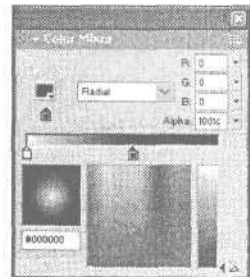
## Имитация трехмерного пространства с помощью инструментов рисования

Вспомните рисунки из школьного учебника по геометрии. Они казались трехмерными, не так ли? В этом разделе вы узнаете, как добиться такого эффекта посредством инструментов рисования Flash.

### Сфера

Создадим один из наиболее часто применяемых трехмерных объектов — сферу.

1. Откройте новый документ, вызвав команду File ▶ New.
2. Активизируйте инструмент Ellipse. Удерживая нажатой клавишу Shift, нарисуйте в центре области действия круг диаметром 3 сантиметра.
3. Убедитесь, что круг имеет сплошную заливку. Для этого отобразите палитру Fill Color, которая находится в секции Colors панели инструментов.
4. С помощью инструмента Arrow выделите заливку круга.
5. Вызовите команду Color Mixer меню Window, вследствие чего на экране появится панель Color Mixer.
6. В списке Fill Style выберите пункт Radial. На данном этапе сфера должна выглядеть так, как показано на рисунке справа. 
7. Как видите, в градиентной заливке по умолчанию используются два цвета: белый (внешняя часть) и черный (внутренняя часть). Перед нами стоит задача создать эффект отражения света от поверхности трехмерной сферы. Для этого цвета нужно изменить таким образом, чтобы внутренняя область заливки была светлее внешней. Прежде всего выделите заливку с помощью инструмента Arrow.
8. Перейдите на панель Color Mixer. Щелкните на правом (белом) бегунке полосы градиента. Затем откройте палитру цветов и выберите темный цвет (в рассматриваемом примере — черный).
9. Щелкните на левом (черном) бегунке полосы градиента и выберите в палитре светлый цвет (в нашем примере — белый). На рисунке справа показано, каким должно быть изображение. 
10. Теперь градиент надлежит отредактировать так, чтобы эффект отражения света от поверхности получился максимально реалистичным. На панели Color Mixer сместите правый бегунок к центру градиента. Таким образом вы измените размер светлой области, призванной имитировать область отражения света.



## Глава 26

# Создание трехмерных объектов в среде Flash

До этого момента мы оперировали только двухмерными изображениями. Трехмерное пространство имитировалось за счет применения законов перспективы (см. практикум 2).

Объекты вокруг нас имеют три измерения: длину, ширину и высоту. В настоящее время для телевидения, фильмов и игр с помощью многочисленных специальных средств создаются трехмерные модели и анимация. Соответствующая технология является узко специализированной, в связи с чем до недавнего времени миры Flash и третьего измерения не соприкасались. Поскольку приложение Flash стало выдающимся средством разработки двухмерных изображений, расширение его возможностей до уровня программы создания трехмерной графики стало неизбежным.

Приложение Flash является средством подготовки двухмерных изображений, а его инструменты рисования не приспособлены к созданию трехмерных объектов. Однако, к счастью, во многих современных программах трехмерного моделирования поддерживается экспорт объектов в среду Flash.

Настоящая глава охватывает следующие темы:

- ▶ имитация трехмерного пространства в программе Flash;
- ▶ экспорт трехмерных изображений, созданных с помощью внешних приложений, в формат SWF;
- ▶ работа с программами, не поддерживающими экспорт в формат SWF.

## Имитация трехмерного пространства в программе Flash

Как отмечалось, Flash не является средством создания трехмерной графики, но в этом приложении с помощью специальных приемов можно изобразить на плоскости трехмерное пространство. Вспомните практикум 2. В процессе рисования клумб мы достигали эффекта трехмерного пространства, трансформируя прямоугольник с помощью инструмента Transform. Вследствие этого двухмерные объекты выглядели так, как будто имеют глубину.

### ПРИМЕЧАНИЕ

Наряду с перспективными изображениями вы можете создавать полноценные трехмерные объекты благодаря дополнениям ActionScript (которые не описываются в данной книге). Примеры эти доступны для просмотра в разделе FLA Web-страницы [www.ult-rashock.com](http://www.ult-rashock.com). Представляют интерес и разработки на Web-странице компании Wireframe Studio ([www.wireframe.co.za](http://www.wireframe.co.za)).



## Разработки

В дополнение мы хотим рассказать о не описанном ранее приложении, которое называется Beatnik, Beatnik не является аудиоприложением. Мы упоминаем его здесь, потому что, подобно Pro Tools и Sound Forge, Beatnik позволяет создавать звуки для Web-страниц, приложений Director и Shockwave, а также, конечно, для Flash. Beatnik не является средой для редактирования и микширования звуков. Оно состоит из четырех главных элементов: подключаемого модуля проигрывателя Beatnik Player, воспроизводимых этим проигрывателем MIDI-файлов, созданных другими программами, и двух элементов кода: объекта Beatnik Music Object и вашей программы (на языке JavaScript, Lingo или ActionScript), с помощью которых вы управляете воспроизведением. Можно сказать, что Beatnik является аудиодвижком для создания музыки и эффектов, не требующих высокой производительности. Используя формат Beatnik RMF (Rich Music Format), можно сжимать аудиоинформацию и передавать ее через узкий канал. Для получения подробных сведений посетите Web-страницу [www.beatnik.com](http://www.beatnik.com).

Пример созданного для Flash аудиомикшера, в котором используется Beatnik, вы найдете по адресу [www.nick.com/music/beatmixer/index.jhtml](http://www.nick.com/music/beatmixer/index.jhtml) (рис. 25.11).

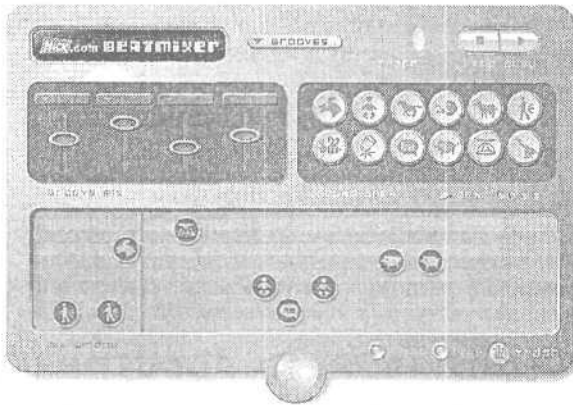
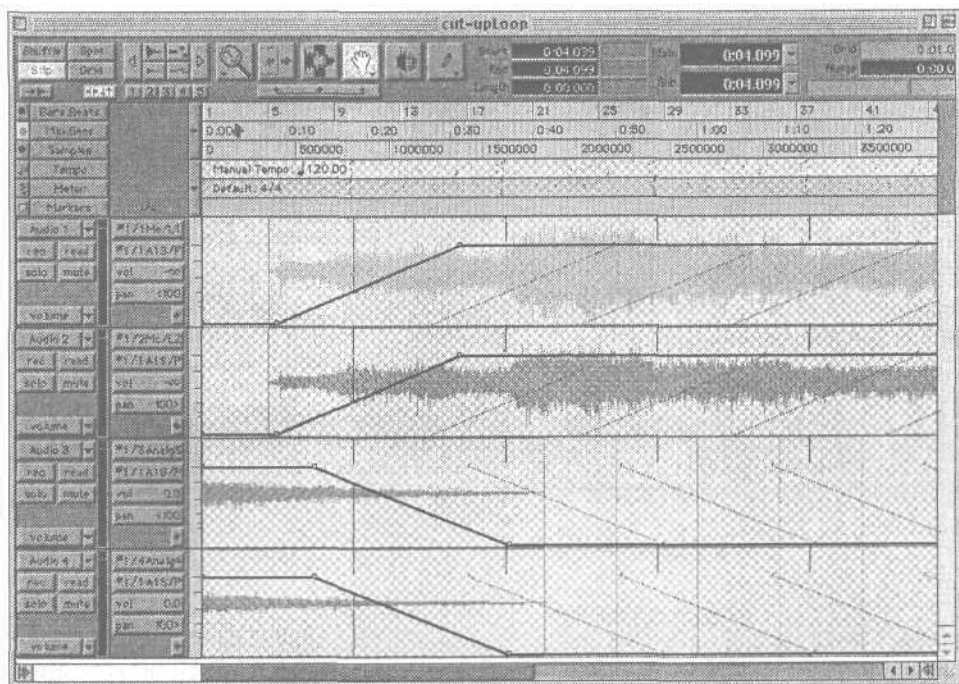


Рис. 25.11. Звуковой микшер Nick.com Beatmixer

## Резюме


В этой главе подведены итоги, касающиеся создания звукового сопровождения Flash-фильмов. Здесь было представлено большое количество разнообразных музыкальных редакторов, с помощью которых вы можете создать звуки, и рассказано о способах работы с этими приложениями. Вы изучили несколько эффектов, которые могут пригодиться при работе со звуком, и ознакомились с некоторыми приемами работы с повторениями. Поэкспериментируйте, и вы поймете, что эти приемы довольно полезны для создания и улучшения качества музыки во Flash-фильмах.



**Рис. 25.10.** Эффект плавного затухания обеспечивает плавный переход от конца к началу звука

5. В заключение выполните микширование редактируемого звука в два канала (стереозвук). Теперь звук готов к импорту во Flash-фильм.

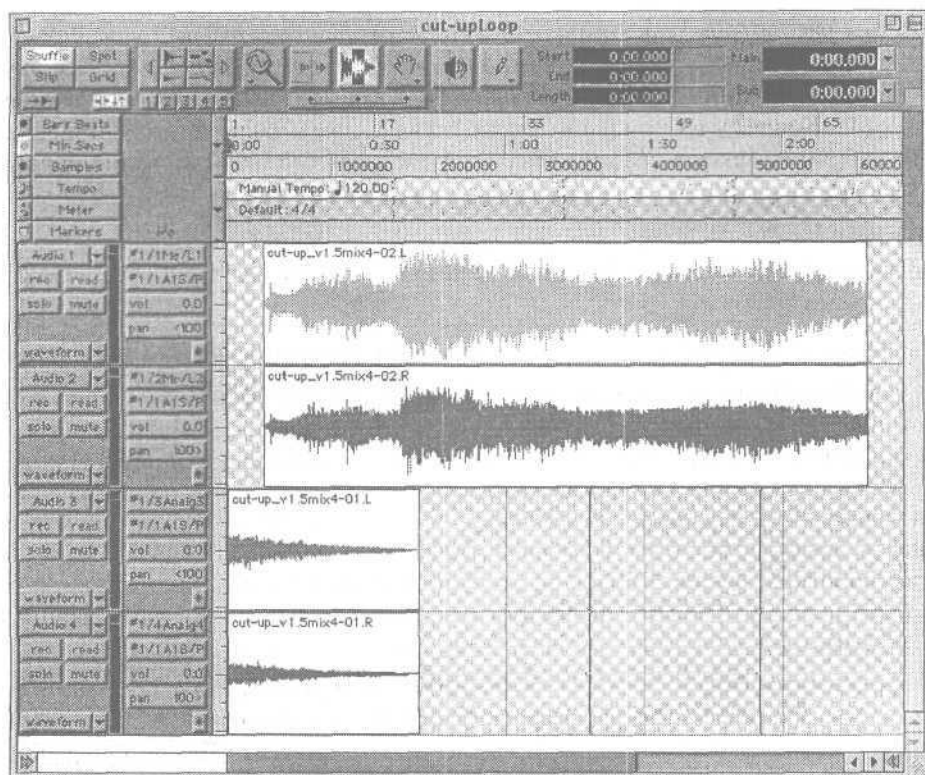
Эмбиент-повторения служат прекрасным фоном, а благодаря неритмичности их можно повторять каким угодно причудливым образом. Не имея резко выделяющихся всплесков, они создают альтернативу ритмическим повторениям.

 На прилагаемом к книге компакт-диске находится файл, в котором используются как ритмические, так и эмбиент-повторения. Открыв в браузере файл `loopDesign.html`, находящийся в каталоге Chapter 25, вы сможете услышать оба вида повторений.

Эмбиент-повторение сохранено в звуковом SWF-файле, который загружается на слой `_level1` в начале фильма. Повторение воспроизводится с синхронизацией типа Event. При его разработке использован только что описанный прием создания плавных (и непрерывных) фоновых повторений.

В этом примере имеются также несколько ритмических повторений, которые хорошо сочетаются со звуковым фоном и создают различное настроение при просмотре разных частей фильма.

Идея заключается в том, что эмбиент-звук проходит через весь фильм, в то время как ритмические повторения вносят разнообразие в его отдельные сцены.



**Рис. 25.9.** Исходный звуковой файл после вырезания нескольких последних секунд звука и вставки их в два новых канала

4. Теперь вы готовы к созданию плавного перехода от звука новых каналов (или от окончания старого звукового файла) к началу звука. Это можно сделать двумя способами.
  - > Задать эффект плавного нарастания звука для каналов с оригинальным звуком и эффект плавного затухания для новых каналов.
  - ▷ Создать перекрестный переход (если редактор поддерживает эту возможность) между четырьмя каналами так, чтобы уровень громкости плавно повышался в начале звука и плавно понижался в конце вырезанного фрагмента.

#### СОВЕТ

Иногда удобно сдвинуть исходный звук вправо, освобождая место для более плавного перехода совмещаемых звуков.

Поскольку новое начало совпадает с концом (а их мы и должны были совместить), в созданном повторении не будет дефектов. И так как вы применили плавный переход между старым началом и старым концом звука, на протяжении повтора не будет резких всплесков или разрывов (рис. 25.10).

Ниже мы рассмотрим, как создать эмбиент-повторения, которые будут служить в качестве звука заднего плана Flash-фильма. *Эмбиент-повторение* — это звук, обычно не имеющий четко выраженного ритма и повторяющийся непрерывно. Питер МакКоннелл из компании Lucas Arts называет такие повторения звуковыми «обоями», позволяющими создать аудиосреду без использования большого количества звуков.

Эмбиент-повторения полезны при создании Web-страниц и интерактивных интерфейсов. Они являются общим элементом дизайна наряду со стилем, шрифтами, цветом и графикой.

Для создания плавного повторения эмбиент-звука выполните следующие действия.

1. Откройте файл с нужным звуком и прослушайте его. Определите, сильно ли отличается звучание в начале и конце звука. Если это так, возможно, потребуется отредактировать звук, изменив порядок следования его частей. Для достижения лучших результатов следует по возможности использовать звуки, в которых начало и конец стыкуются без перепадов уровня. Продолжительность звука должна быть не меньше тридцати секунд. При использовании более коротких звуков в большей степени проявляется их повторяемость, поэтому предпочтительней использовать звуковой файл настолько длинный, насколько это позволяют размеры фильма.
2. Используя звуковой редактор, выделите звуковую волну на последних нескольких секундах. (Выделенный фрагмент будет служить переходом для начала следующего звука. Попробуйте выделить как можно больший фрагмент так, чтобы было достаточно места для перехода. От четырех до десяти секунд будет достаточно.) Если звук стереофонический, необходимо производить выделение в обоих каналах. Удостоверьтесь в том, что выделенный фрагмент начинается в точке, где звуковая волна пересекает нулевую амплитудную линию. Если выделение начнется в другой точке волны, то возникнет скачок амплитуды при повторении звука.
3. Выберите команду Edit > Cut (клавиши Cmd/Ctrl+X), чтобы вырезать выбранный фрагмент. Затем создайте два новых канала (или одну стереопару) в звуковом файле. Выделите новые каналы. Удостоверьтесь, что курсор находится в самом начале звукового файла, и затем выберите команду Edit ▶ Paste (клавиши Cmd/Ctrl+V) для вставки в новые каналы информации, которую вы вырезали из конца файла. Теперь форма волны звука должна выглядеть примерно так, как показано на рис. 25.9.

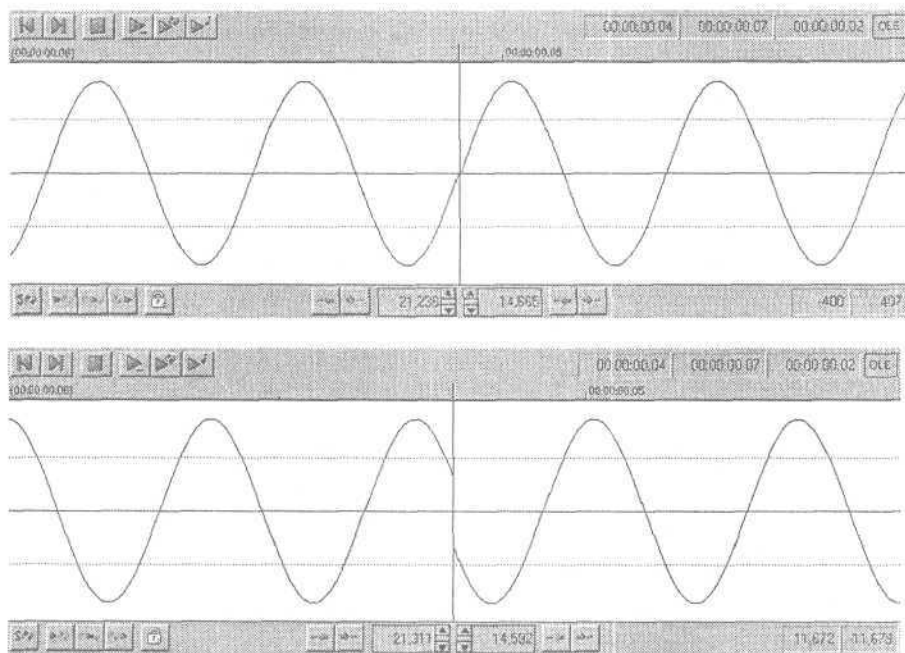
## СОВЕТ

При использовании редакторов, которые не поддерживают работу с многоканальным звуком (например Sound Forge или Peak), также можно применять этот прием, но с меньшей гибкостью. Вместо создания двух новых каналов вставьте фрагмент, вырезанный из конца, в начало файла. Задайте параметры вставленного фрагмента так, чтобы его громкость изменялась от максимальной до нуля, а исходного звука — от нуля до максимальной.

## Создание плавных повторений звука

В главе 19 были описаны повторяющиеся музыкальные фрагменты, а в практике 6 вы научились вставлять эти фрагменты во временную шкалу Flash-фильма с сохранением ритма при переходе к очередному повтору. Хотя повторения привносят элемент монотонности в звуковое сопровождение, тем не менее, используя их, можно создать довольно интересные композиции. Следует помнить, что повторения совсем не обязательно должны восприниматься как примитивные или навязчивые звуки, такими их может сделать неверное применение.

Каким бы образом вы не использовали повторения, главное, чтобы звук был чистым, без хлопков, щелчков и других помех. Наилучший способ это проверить – убедиться, что звуковой файл содержит единую область звука без скачков и разрывов. Если в форме волны имеются какие-либо скачки или пустые места, они будут резать слух при каждом повторе (рис. 25.8).



**Рис. 25.8.** На верхнем изображении показано повторение без каких-либо помех, а на нижнем имеется разрыв волны, который будет воспроизводиться при каждом повторе

Плавность звучания при повторе обеспечивается отсутствием разрывов звуковой волны. Поскольку найти естественный звук, который бы удовлетворял этим требованиям, практически невозможно, звуковой файл следует отредактировать. Существует множество приложений обработки звука, предоставляющих инструменты создания повторений. Такие инструменты могут быть очень полезны для редактирования ритмических повторений, например музыки, но их довольно трудно применять к неритмичным «рваным» звукам.

- ▶ Затухание. Этот параметр управляет *длительностью* реверберации.
- ▶ Диффузия. С помощью данного параметра задается *интенсивность* реверберации.

## Изменение высоты звука

*Изменение высоты звука* (pitch shift) — это эффект, который позволяет создавать довольно необычную и гротескную музыку. Данный эффект трансформирует звук в более высокий или более низкий. Используя его, можно, например, сделать голоса людей похожими на голоса героев мультфильмов. Изменение высоты звука — прекрасный способ создания интересных и неожиданных звуковых эффектов.

Во многих случаях для *изменения* высоты звука изменяется его *длительность*: при *сокращении* времени звучания звук становится выше, а при *увеличении* — ниже. В современных программах для создания этого эффекта вы можете сохранять исходную *длительность* звука, сделав соответствующие установки.

Чтобы *изменить* высоту звука, выполните следующие действия.

1. Выделите фрагмент звука или звук целиком, для которого вы хотите применить этот эффект.
2. Активируйте команду Pitch Shift из меню Effects. Откроется *окно* Pitch Shift.
3. Введите значение, на которое следует изменить высоту звука, и *нажмите* кнопку ОК.

Немного поэкспериментировав, вы будете удивлены тем, какие интересные звуки можно создавать с помощью этого эффекта. Изменение высоты звука широко используется во Flash-мультфильмах.

## Изменение времени звучания

Эффект *изменения времени звучания* (time shift) позволяет изменить *длительность* звука без изменения его высоты. Для применения этого эффекта выполните следующие действия.

1. Выделите фрагмент файла, для которого хотите *изменить* время звучания.
2. Активируйте команду Time Shift (Временной сдвиг) из меню Effects аудиоприложения. Она может также называться Tempo (Темп) или Stretch (Растянуть).
3. В открывшемся окне либо передвиньте ползунок для *установки значения*, либо введите значение.
4. Используйте функцию предварительного прослушивания звука и, если звучание вас удовлетворяет, нажмите кнопку ОК.

*Изменение времени звучания* можно использовать в различных ситуациях, например, если вам необходимо продлить звучание какого-либо звука на нужное время (скажем, звука взрыва).

и впадин звука, сохраняя их отношение, а компрессия влияет на динамический диапазон звука и сближает значения амплитуд. Если вы имеете звук с сильно меняющимся уровнем громкости, компрессор может уменьшить его динамический диапазон, понизив амплитуду громких частей до уровня тихих. Затем после уравнивания амплитуд компрессор увеличит уровень громкости звука во всем файле.

Для компрессии звука выполните следующие действия.

1. Выделите весь звуковой файл или фрагмент, для которого нужно применить компрессию.
2. Активизируйте команду Compressor (Компрессор) или Dynamics (Динамический диапазон). Обычно она находится в меню Process или Effects. Откроется окно Dynamics, в котором для большинства редакторов необходимо задать пять параметров: Threshold (Порог), Ratio (Отношение), Attack (Нарастание), Release (Спад) и Output (Выход).
  - ▷ Threshold задает величину порога, при котором начинается компрессия.
  - ▷ Ratio указывает коэффициент компрессии. Обычно используется коэффициент в пределах от 2:1 до 6:1. То есть каждые 2 дБ (6 дБ) исходного динамического диапазона сжимаются в 1 дБ.
  - ▷ Attack и Release задают скорость действия компрессора.
  - ▷ Output позволяет увеличивать уровень громкости компрессированного звука.
3. Используйте функцию предварительного прослушивания звука для проверки результатов компрессии. Если результаты вас устраивают, нажмите кнопку ОК.

Сжатие может пригодиться во Flash-проектах. Используйте сжатия для диалога и пересказа так, чтобы звук был ровным и не содержал «дефектов речи», препятствующих четкости восприятия смысла.

## Реверберация

*Реверберация* или *искусственное эхо* используется для создания звука, который звучит как бы в некотором акустическом пространстве. Этот эффект помогает создать звуки, источник которых расположен в ванной, холле или кафедральном соборе. Обычно искусственное эхо применяется для имитации его естественного прототипа.

Работая со звуками для Flash, вы можете использовать эхо для придания большей естественности голосу (разговору или песне) либо для придания ощущения реальности изображенному в анимации интерьеру, если он «в жизни» обладает акустическими свойствами. Существует много видов искусственного эхо. Ниже мы приводим описание некоторых его параметров, чтобы затем рассмотреть инструменты для его создания.

- ▶ **Размеры помещения.** Позволяет имитировать естественную акустику различных помещений.
- ▶ **Фильтры низких и высоких частот.** Предназначены, чтобы подавлять высокие или низкие частоты звука, для которого применяется реверберация.

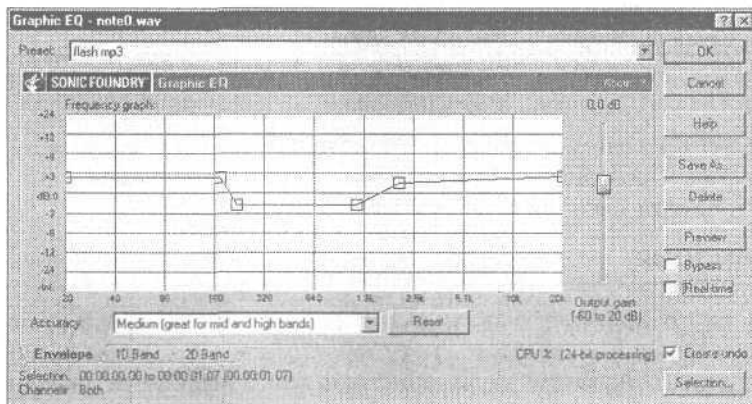
### Баланс частот в звуковых MP3-файлах Flash

При экспорте звука с внутренним MP3-сжатием среды Flash ухудшается его качество, причем проявляется это по-разному. Для некоторых звуков изменения, обусловленные сжатием, являются минимальными, а для других катастрофически ухудшают качество. В основном, с уменьшением частоты и глубины выборки: качество звука понижается, поскольку при сжатии он изменяется больше.

Не следует оставлять в фильме эти изменения, не пытайтесь настроить звук. Такие настройки можно сделать, используя эквалайзер. Следует изучить, какие частоты преобразуются при сжатии, и затем изменить их уровень громкости. Понижая: уровень громкости частот, в области которых сжатие увеличивает амплитуду, и: повышая уровень громкости подавляемых сжатием частот, вы можете добиться сохранения большинства исходных характеристик звуков, сжатых при публикации фильма.

Эти изменения нужно делать непосредственно перед импортом звука в среду Flash. Таким образом вы заранее скомпенсируете эффекты искажения звука при сжатии. В идеальном случае после публикации Flash-фильма все звуки возвращаются к оригиналу. Понятно, что звук никогда не будет точно соответствовать исходному, но без такой настройки это различие будет более заметным.

Используйте рисунок, приведенный ниже, как руководство для баланса частот в звуковых файлах, которые будут сжаты при импорте.



Заметьте, что следует увеличить уровень громкости низких частот, уменьшить уровень громкости в области средних (ближе к низким) частот и слегка повысить уровень громкости высоких частот. Эта схема является весьма приближенной, поскольку одни и те же параметры эквалайзера действуют различным образом на разные звуки. Используйте приведенный на рисунке пример как некую точку отсчета. Немного поэкспериментировав, вы сможете сделать звучание сжатых файлов близким к оригиналу.

### Компрессия

Компрессия, как и нормализация, влияет на уровень громкости звука, но это не просто изменение уровня громкости. Нормализация влияет на амплитуды пиков



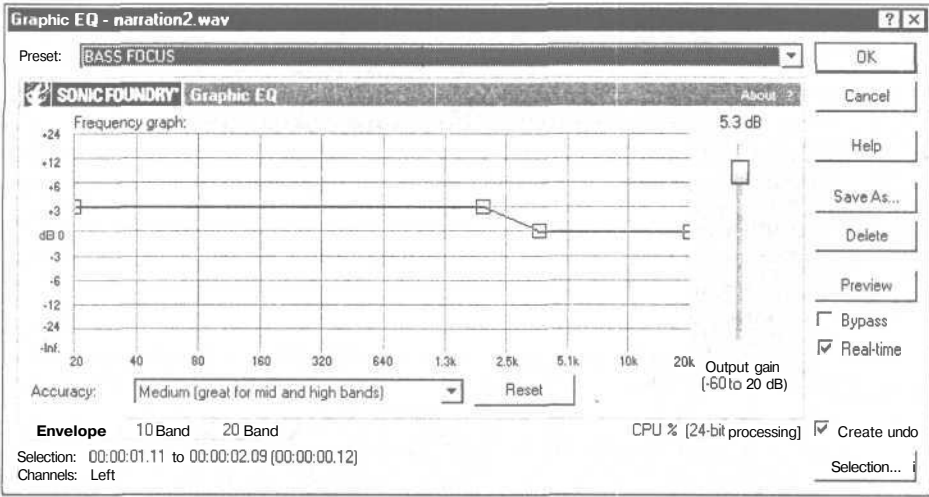


Рис. 25.7. Графический эквалайзер программы Sound Forge

- Используя ползунки эквалайзера, задайте значения уровня громкости для различных частот звука.

Таблица 25.1. Таблица частот, которую создал Берни Мак

| Частота звука (в герцах) | Описание звука             |
|--------------------------|----------------------------|
| 60-80                    | Bass (Бас)                 |
| 200-250                  | Muddy (Низкий)             |
| 350                      | Pinched, nasal (Носовой)   |
| 450                      | Warm (Теплый)              |
| 5000                     | S's (Шипящий)              |
| 7000                     | Attack (Атака)             |
| 12000                    | High end, bright (Высокий) |

Список частот приведен в табл. 25.1. Инженер звукозаписи и преподаватель из Чикаго Берни Мак разработал эту таблицу для того, чтобы помочь студентам сравнивать звуковые характеристики или описание со значениями частот. Например, вы слышите звук и определяете его как muddy. Muddy (Низкий) — хорошее слово для описания, но оно не говорит вам, как сделать звук чище. Используя таблицу, вы можете увидеть, что частота для звука Muddy обычно находится в пределах 200-250 герц. Уменьшая уровень громкости этих частот, вы можете сделать звук чище. Чтобы сделать звук более агрессивным, нужно изменять уровень громкости для частот звука Attack: 7000 герц. Согласитесь, эта таблица очень полезна для настройки звука, поскольку руководствуясь ею, можно сделать каждый звук более выразительным.

- Некоторые эквалайзеры позволяют выполнить предварительное прослушивание звука с новыми установками. Нажмите кнопку Preview для прослушивания звука. После завершения редактирования щелкните на кнопке ОК.

заранее предположить, что звуки будут воспроизводиться при максимально возможном уровне громкости.

Нормализация используется для приведения уровня громкости звука к определенной норме. При выполнении этой процедуры в звуковом файле отыскивается место с максимальным уровнем громкости, затем этот уровень громкости изменяется до определенного значения (увеличивается или уменьшается), а все остальные уровни громкости изменяются пропорционально.

Для нормализации звукового файла выполните следующие действия.

1. Выделите фрагмент звукового файла, который вы хотите нормализовать. Как правило, нормализуют весь файл.
2. Активизируйте команду **Normalize** (Нормализовать). Обычно она находится в меню **Process** (Обработка) или **Effects** (Эффекты). В результате откроется окно **Normalize**.
3. В поле этого окна введите значение уровня громкости, обычно 96–98 %. В этом случае уровень громкости в точке самого громкого звучания станет равным 96–98 % максимально возможной громкости, а уровни громкости в других точках будут изменены пропорционально.
4. Нажмите кнопку **ОК**. Вы увидите, как изменилась амплитуда звукового файла. Пример изменения показан на рис. 19.9 в главе 19.

Перед импортом в среду **Flash** нужно нормализовать все звуки, чтобы гарантировать их неискаженное воспроизведение при максимальном уровне громкости. Позже, в среде **Flash**, вы всегда можете понизить их уровень громкости, отредактировав огибающую звуковой волны в окне **Edit Envelope** (кнопка **Edit** на панели **Properties**) или с помощью методов объекта **Sound**.

## Эквалайзер

*Эквалайзер* — это инструмент, предназначенный для изменения баланса частот звука. При понижении частоты вы получите глухой звук, а при увеличении — более звонкий.

Обычно эквалайзер используется для того, чтобы сделать звук чище. Как и при использовании большинства эффектов, вашим главным индикатором является слух. Работайте с эквалайзером до тех пор, пока не услышите, что звук стал такой, как вам нужно. Необходимо постоянно следить за уровнем громкости, чтобы не произошло искажение звука, поскольку вы можете не заметить, как эквалайзер увеличит громкость выше нормы.

Чтобы применить эквалайзер к звуковому файлу, выполните следующие действия.

1. Выделите фрагмент звука, для которого вы хотите установить баланс частот. Обычно это делается для всего звукового файла.
2. Активизируйте команду **Equalizer** (Эквалайзер), которая находится в меню **Process** или **Effects**. Откроется окно **Equalizer** (рис. 25.7).

что Cubase VST совместимо с платформами как Windows, так и Macintosh. Для более детальной информации посетите Web-страницу [www.steinberg.net](http://www.steinberg.net).

- ▶ **Digital Performer.** Это приложение, созданное фирмой Mark of the Unicorn (MOTU), работает только на платформе Macintosh. Оно также предназначено для работы с MIDI-последовательностями и любой другой музыкой. Для более детальной информации посетите Web-страницу [www.motu.com](http://www.motu.com).

Все продукты, рассмотренные в этой главе, схожи между собой и в равной степени пригодны для создания цифрового звука. Основываясь на том, что требуется для вашего Flash-фильма, вы экспериментальным путем выберете для себя наиболее подходящее приложение.

## Приемы обработки цифрового звука

В предыдущем разделе мы рассмотрели большое количество инструментов создания, редактирования, микширования, а также подготовки к производству музыки и звуков для Flash-фильмов и анимации. Впрочем, выбор инструмента для работы это только начало, ведь нужно еще уметь эффективно и профессионально его использовать. А чтобы научиться этому, следует понять, как средства звуковых редакторов обрабатывают файлы.

В этом разделе описаны различные приемы обработки цифрового звука, которые помогут вам подготовить звуковое сопровождение Flash-фильма и других видов цифровых средств массовой информации. Если вы не знакомы с принципами цифровой обработки звука, обратитесь к главе 19. Поскольку существует очень много различных приложений обработки звука, мы приводим в этом разделе только общие инструкции и перечни действий. Каждый прием показан таким образом, чтобы его можно было использовать в любом приложении.

## Аудиоэффекты и обработка

В мире цифрового звука есть два вида редактирования: деструктивное и недеструктивное. Названия говорят сами за себя. *Недеструктивное* редактирование не изменяет оригинал. Обычно создается и изменяется новый файл, а содержимое оригинального файла не изменяется. При *деструктивном* редактировании звук изменяется, если вы не отмените операцию редактирования. В зависимости от выбранного редактора вы будете выполнять либо деструктивное редактирование, либо недеструктивное, а может быть и то и другое. Перед работой удостоверьтесь, что неверные изменения оригинального файла не повлекут неисправимых последствий. Если все-таки вы сделали изменения там, где они не нужны, используйте для отмены команду Undo либо клавиши `Cmd/Ctrl+Z`.

## Нормализация

В определенном смысле *нормализация* является наиболее важной частью обработки цифрового звука. Причина этого заключается в том, что разработчик не знает, какой уровень громкости будет установлен при воспроизведении звука, и не может управлять этим процессом. Единственное, что можно сделать, это

С помощью редактора ACID PRO можно изменять темп звучания и уровень громкости (путем редактирования огибающей звуковой волны), создавать эффекты панорамирования, затухания и плавного перехода. Как и другие приложения компании Sonic Foundry, редактор ACID PRO поддерживает 18 подключаемых модулей DirectX, разработанных этой фирмой и предназначенных для создания дополнительных эффектов и обработки звука. В комплект поставки ACID PRO входит компакт-диск, содержащий большую библиотеку музыкальных фраз различных стилей. Кроме того, библиотеки повторений можно получить от компании Sonic Foundry и других производителей. ACID PRO работает в среде Windows 98 SE, ME, 2000 и XP.



Демонстрационная версия программы ACID Xpress 3.0 находится на прилагаемом компакт-диске. Эта пробная версия редактора поможет вам без оплаты за обучение освоить приемы создания музыки с повторениями.

## Редакторы MIDI-музыки

Стандарт MIDI или Musical Instrument Digital Interface (Цифровой интерфейс музыкальных инструментов) широко используется в компьютерной обработке звука уже многие годы. Этому стандарту можно посвятить отдельную книгу. MIDI-музыка состоит из двух главных компонентов: последовательности и звукового банка, или модуля. *Последовательность* — это звуковой файл, сохраненный в MIDI-совместимом формате, который служит в качестве своеобразного цифрового пианино. При воспроизведении этого файла интерфейс MIDI связывается со *звуковым модулем*. Звуковой модуль содержит образцы звуков. Он получает MIDI-информацию от последовательности и воспроизводит в реальном масштабе времени нужные звуки и эффекты. Для создания MIDI-музыки вам нужно два компонента: звуковой модуль с образцами звуков и программа создания MIDI-файлов.

MIDI-музыку можно использовать во Flash-фильмах. Очень важно понимать, что MIDI-последовательность сама по себе не музыка, а список инструкций, которые пересылаются в звуковой модуль. Когда звуковое устройство принимает инструкции, оно воспроизводит соответствующие ноты. В результате, чтобы сформировать MIDI-музыку, нужно записать в цифровой аудиофайл звуки, получаемые от воспроизводящего устройства.

Ниже перечислены приложения, предназначенные для создания MIDI-последовательностей.

- ▶ **Cakewalk Home Studio 2002.** Это приложение совместимо только с операционной системой Windows. Оно предназначено для формирования MIDI-последовательностей и поддерживает некоторые другие описанные в этой главе возможности, в частности, работу с многоканальным цифровым звуком. Для получения более подробной информации об этом редакторе и других подобных программах, таких как Cakewalk Pro Audio или Sonar, посетите Web-страницу [www.cakewalk.com](http://www.cakewalk.com).
- ▶ **Cubase VST.** Разработанное компанией Steinberg Media Technologies AG, это приложение, как и Cakewalk, может «делать все». Различаются они только тем,

Поскольку этот редактор разработан достаточно давно, он имеет минимальные требования к системе: ОС Mac System 7.1 или выше; процессор 68030 или быстрее.

## Редакторы повторений

Редакторы повторений удобно применять, если нужно создать музыкальное сопровождение к фильму, не углубляясь в тонкости композиции. При использовании повторений необходимо скомпоновать и отредактировать не большое произведение, а только один фрагмент, в дальнейшем его можно копировать, разделять на слои, микшировать, чтобы создать музыкальное произведение большего размера. Редакторы повторений являясь хорошим инструментом для создания оригинальной музыки к Flash-фильмам. Более подробно специфические приемы работы с повторениями без нарушения ритма описаны в разделе «Создание плавных повторений звука» этой главы и в практикуме 6.

### ACID PRO

Программа ACID PRO, разработанная компанией Sonic Foundry, предназначена для создания повторений и музыки с повторениями. Используя эту программу, вы выбираете нужные повторения и выстраиваете их на временной шкале редактора ACID PRO, а затем нажимаете кнопку Play и прослушиваете результат. Поскольку большинство повторений являются законченными звуками, вам остается лишь выстроить их в нужной последовательности. ACID PRO позволяет легко создавать музыкальные произведения (рис. 25.6).

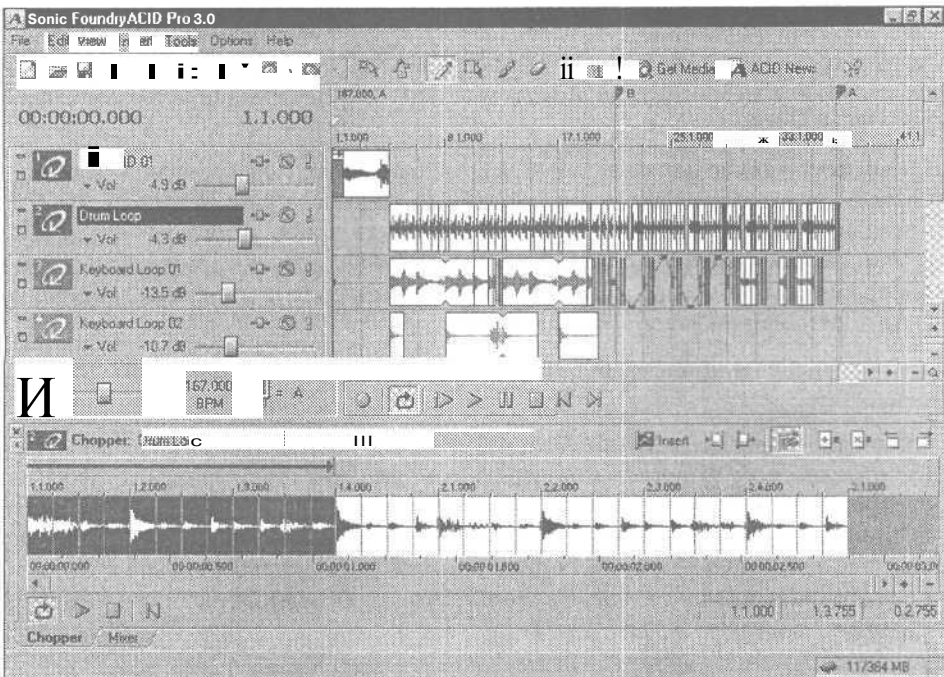


Рис. 25.6. Среда редактора ACID PRO

уже не совершенствуется, он по-прежнему широко используется во многих сферах цифровой медиаиндустрии, и на Web-узле компании Macromedia имеется страница его поддержки ([www.macromedia.com/software/sound](http://www.macromedia.com/software/sound)). Как редактор формы волны он имеет большое количество типичных для подобных приложений возможностей: копирование, вырезание и вставка фрагментов, создание эффектов затухания и нарастания звука, нормализация, изменение темпа звучания и многое другое. SoundEdit 16 предоставляет возможность создания и вставки точек повтора в аудиофайл или выборку. Проще говоря, SoundEdit 16 (рис. 25.5) является простым и удобным аудиоредактором с широкими возможностями.

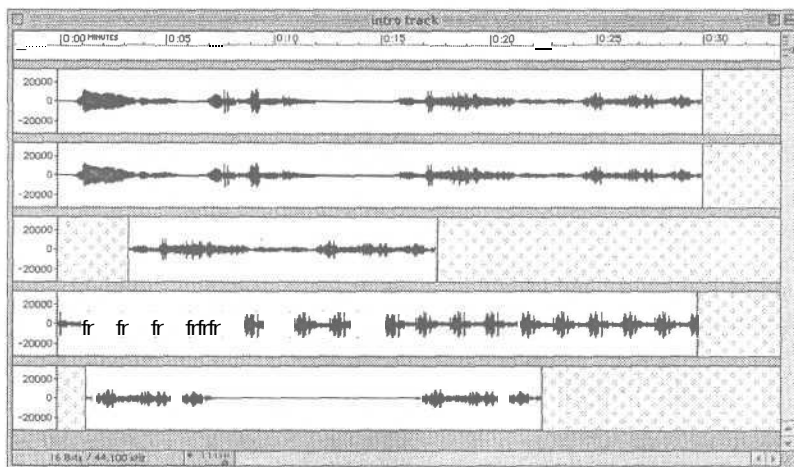
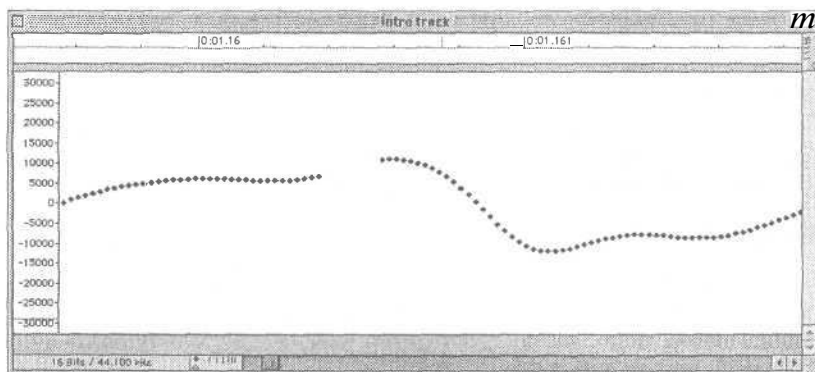


Рис. 25.5. Звуковой редактор Sound Edit 16

Главным отличием SoundEdit 16 от остальных редакторов формы волны является то, что он поддерживает несколько звуковых каналов. Это свойство делает его незаменимым для создания уникальных многослойных звуков или редактирования и синтеза многокомпонентных звукозаписей. Хотя у него нет такого количества функций как, например, у Pro Tools или некоторых других рассмотренных редакторов, он может сделать больше, чем Sound Forge и Peak 3 вместе взятые. Кроме всего прочего, в этом редакторе можно редактировать отдельные выборки.



позволяя вырезать и вставлять фрагменты, нормализовать звук, создавать эффект затухания и кросс-затухания, изменять высоту и длительность звуков, а также выполнять преобразование монозвука в стерео. Помимо детального отображения формы волны, этот редактор предоставляет полезную возможность видеть форму волны всего звукового файла (рис. 25.4).

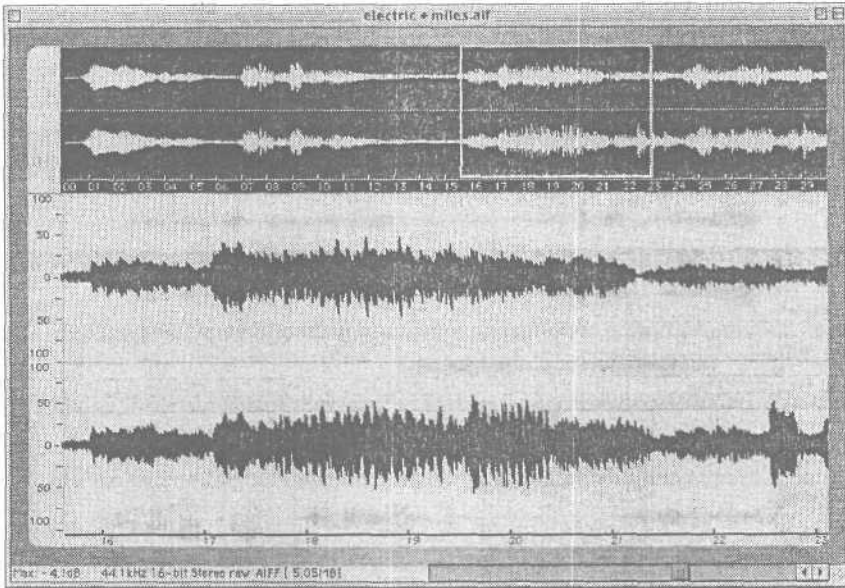


Рис. 25.4. Интерфейс редактора Peak 3

В дополнение ко всему этому Peak 3 поддерживает подключаемые модули Steinberg VST, которые можно использовать для реализации звукового эквалайзера, сжатия, создания эффекта реверберации и других видов обработки звука. Компания BIAS Inc. разработала также собственный десятиполосный эквалайзер SuperFreq.

Кроме того, редактор Peak 3 предоставляет возможность пакетной обработки звуковых файлов, что может оказать неоценимую помощь, когда потребуется, например, изменить частоту выборки для огромного количества файлов, преобразовать каждый из них в другой формат и т. п.

Peak 3 работает на компьютерах PowerPC или G3/G4 Macintosh и адаптирован для операционной системы Mac OS X. Чтобы получить дополнительную информацию о системных требованиях и совместимости или загрузить пробную версию программы Peak, посетите Web-страницу компании BIAS ([www.bias-inc.com](http://www.bias-inc.com)).

## Sound Edit 16

SoundEdit 16 является классическим звуковым редактором. Большинство пользователей компьютеров Macintosh, создающих продукты мультимедиа или редактирующих видео, используют его в своей работе. Это уникальное аудиоприложение создано компанией Macromedia. Хотя в настоящее время этот редактор

## Sound Forge

Программа Sound Forge быстро стала одной из ведущих в цифровой медиаиндустрии. Этот звуковой редактор, созданный только для операционной системы Windows, обеспечивает запись, вырезание, вставку и обрезание звука, создание эффекта затухания, редактирование повторений и многое другое. Кроме того, в программе Sound Forge имеется три вида эквалайзеров: графический, параметрический и параграфический. Редактор можно использовать для формирования файлов MP3 с целью дальнейшего импорта их во Flash. Вдобавок Sound Forge предоставляет ряд звуковых подключаемых модулей DirectX: Amplitude Modulation, Chorus Delay, Distortion, Flange, Gapper, Noise Gate, Pitch Bend, Reverb, Vibrato, Time Compression, Wave Hammer и др. На рис. 25.3 показано окно редактора Sound Forge.

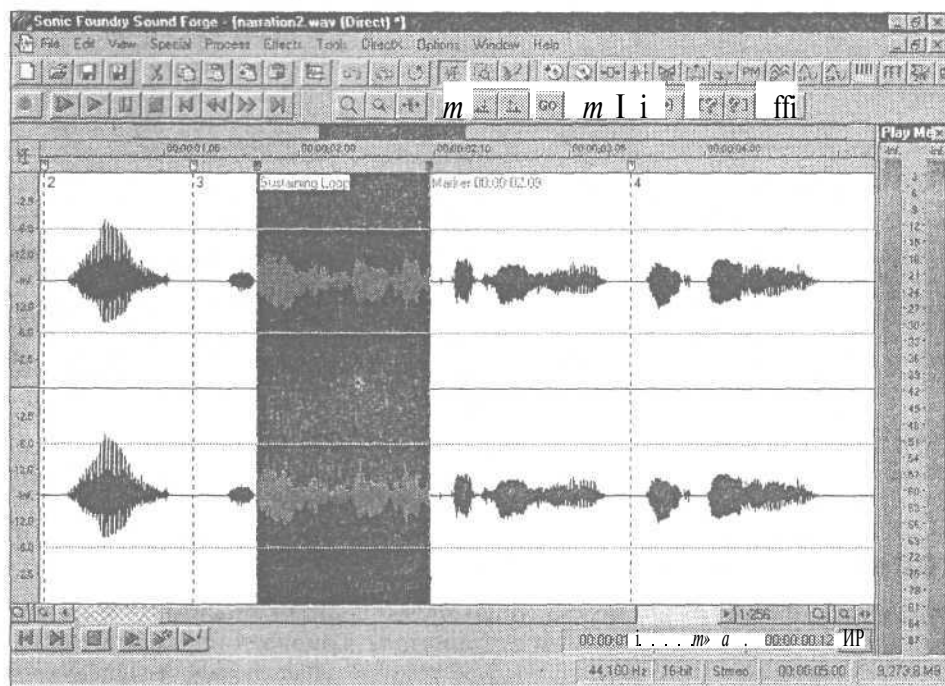



Рис. 25.3. Окно редактора Sound Forge

 Sound Forge был создан компанией Sonic Foundry ([www.sonlcfoundry.com](http://www.sonlcfoundry.com)) и работает в среде операционных систем Windows 98 SE, Me, 2000 и XP. Демонстрационная версия редактора находится на компакт-диске, прилагаемом к этой книге.

## Peak 3

Редактор Peak 3 разработан компанией BIAS (Berkley Integrated Audio Software). Это один из немногих профессиональных редакторов, созданных для платформы Macintosh. Как звуковой редактор он имеет большие возможности,



восьми каналов. Обе версии работают в операционных системах Windows 98 SE, ME, 2000 и XP.

Полную информацию о редакторе Vegas Audio и требованиях к системе вы можете получить, посетив Web-узел компании Sonic Foundry. Оттуда же можно загрузить пробную версию программы.

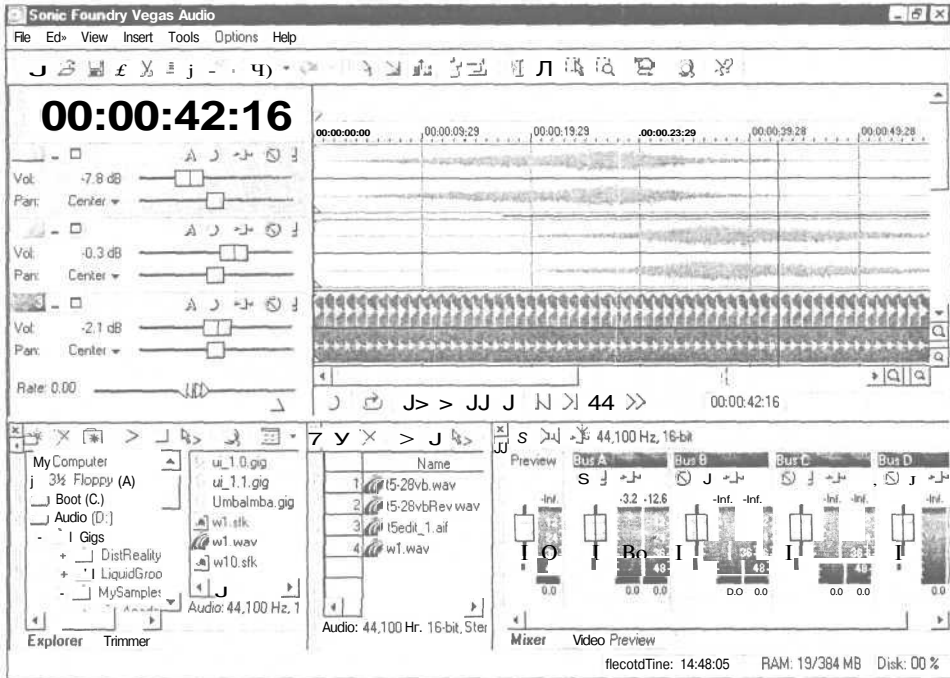


Рис. 25.2. Окно редактора Vegas Audio

## Редакторы формы волны

В редакторах формы волны используется другой подход к редактированию звука. Большинство этих редакторов обладает многими возможностями редакторов обработки многоканального звука, а главное отличие их заключается в следующем: редакторы формы волны в основном позволяют работать одновременно только с одним или двумя каналами (моно- и стереофайлами). На первый взгляд это ограничение может показаться вам слишком серьезным, чтобы использовать такие редакторы, но здесь есть свои преимущества. Например, редакторы формы волны имеют меньшее количество опций, и поэтому позволяют работать быстрее. Возможно, эти редакторы по своим возможностям и уступают редакторам обработки многоканального звука, но зато предоставляют простую и удобную среду для редактирования и подготовки звуковых файлов. При помощи редактора формы волны можно выполнять от 85 % до 90 % всего объема работы со звуком, необходимой для создания полноценного Flash-фильма.

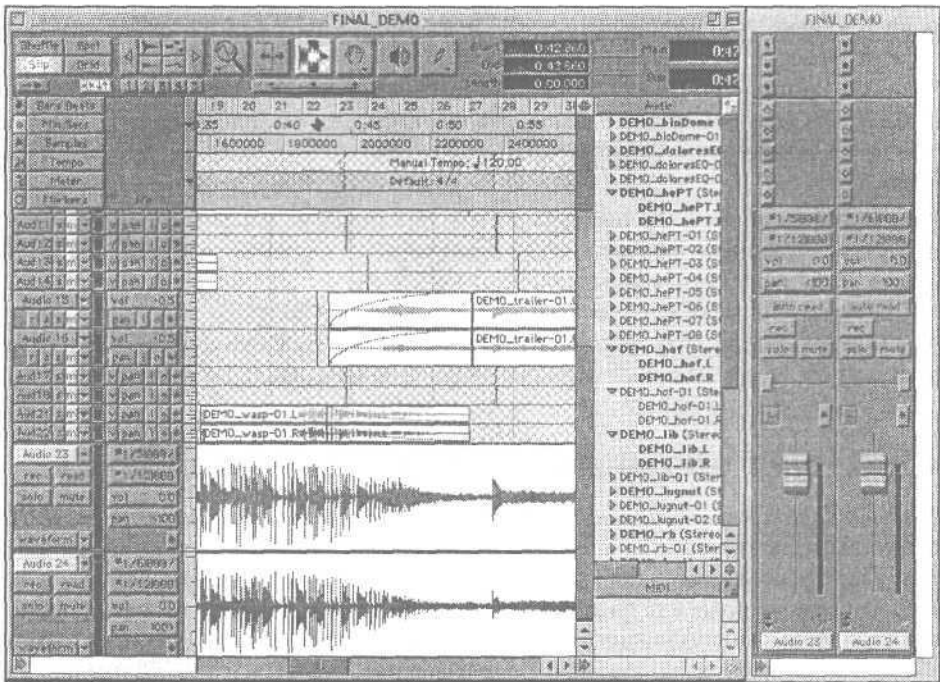


Рис. 25.1. Окно звукового редактора Pro Tools LE

В настоящее время средства Pro Tools работают на платформах Macintosh, Windows 2000, 98 и ME. Для получения последней информации о Pro Tools посетите Web-страницу компании Digidesign ([www.digidesign.com](http://www.digidesign.com)). Чтобы получить пробную версию программы, активизируйте ссылку Pro Tools FREE, и хотя вы загрузите неполную версию, она позволит почувствовать всю мощь средств Pro Tools. Бесплатная пробная версия работает на платформах Macintosh и Windows.

## Vegas Audio

Программа Vegas Audio является редактором для платформы Windows. Это редактор многоканального звука с инструментами микширования и редактирования, созданный компанией Sonic Foundry ([www.sonicfoundry.com](http://www.sonicfoundry.com)). В отличие от Pro Tools TDM он не требует специализированного оборудования, а использует процессор компьютера. Приложение Vegas Audio имеет простой интерфейс, поддерживающий функции drag-and-drop, который позволяет комбинировать различные звуковые элементы для создания многогранного звукового сопровождения фильма. На рис. 25.2 показан интерфейс редактора Vegas Audio.

Программа поддерживает неограниченное число звуковых каналов и до 18 аудиоэффектов Sonic Foundry DirectX (XFX 1,2 и 3). Также обеспечивается синхронизация с QuickTime 4, которая может быть использована при публикации Flash-фильмов в формате QuickTime. В настоящее время компания Sonic Foundry предоставляет две версии этого редактора: Vegas Audio и Vegas Audio LE. Неполная версия Vegas Audio LE имеет большинство возможностей полной версии, но с некоторыми ограничениями, например, возможно использование не более

предоставляют в распоряжение пользователя пробные версии своих приложений. Согласитесь, что бесплатный тест-драйв, дающий возможность полностью проверить программу, является прекрасной возможностью для выбора продукта. В этой главе мы приводим Web-адреса, по которым можно получить дополнительную информацию о каждом приложении, его производителе, "совместимости и цене.

Данный раздел состоит из четырех частей: приложения обработки многоканального звука, редакторы формы волны, редакторы повторений и редакторы MIDI-музыки. Если вы уже знакомы с этими терминами, то без труда найдете нужную информацию. Если же все эти термины вы слышите впервые, продолжайте читать главу подробно. Мы постарались привести все нужные определения и рассказать о работе приложений каждого вида, чтобы читатель смог выбрать приложения, необходимые для реализации его проектов.

## Редакторы обработки многоканального звука

Основные функции приложений обработки многоканального звука заключаются в следующем: запись звука, вырезание фрагментов, редактирование и микширование звука. Большинство из этих приложений используется для реализации сложных проектов, которые требуют объединения многих различных звуковых элементов в одну музыкальную композицию. Если вы планируете создавать собственную музыку или компоновать звуковое сопровождение, используя повествование, диалог, музыку и звуковые эффекты, то эти редакторы предназначены для вас.

### Pro Tools

Средства обработки звука Pro Tools, разработанные компанией Digidesign, в настоящее время являются промышленным стандартом средств для профессиональной звукозаписи, редактирования и микширования цифрового звука. Эти приложения позволяют обрабатывать многоканальный, объемный и MIDI-звук. Кроме обработки звука Pro Tools поддерживает использование видео.

Средства Pro Tools содержат системы двух типов. Системы первого типа называются системами TDM (Time Division Multiplexing -- мультиплексирование с разделением времени). Посредством системы Pro Tools TDM звуковая информация передается через специализированное оборудование, установленное на компьютере. Она управляет цифровой обработкой сигналов и позволяет компьютеру одновременно работать с несколькими аудиопотоками в реальном времени. Системы TDM предоставляют большую мощь и гибкость, но стоят довольно дорого и предназначены для профессионалов.

Системы второго типа называются RTAS (Real-Time Audio Suite — набор программ обработки звука в реальном масштабе времени). Главным их отличием от систем TDM является то, что вместо специальных устройств обработки звука используется процессор компьютера. Поэтому производительность системы в большой степени зависит от скорости и эффективности выполнения программы компьютером. Система Pro Tools LE типа RTAS в основном предназначена для реализации небольших проектов и имеет все возможности редактирования звука, присущие системам TDM. На рис. 25.1 показано окно редактора Pro Tools LE.

## Глава 25

# Использование звуковых редакторов совместно с Flash

Из всех приложений, которые можно использовать совместно с Flash, звуковые редакторы менее всего похожи на Flash. В них нет ни кадров с анимацией, ни видеоклипов, ни макроязыка. Они предназначены для изменения звуковых файлов при помощи богатых наборов инструментов. Чтобы редактировать звук, требуется особый вид мышления, способность переходить от графической временной шкалы к звуковой. При этом наиболее важным элементом пользовательского интерфейса являются уши!

Из этой книги вы пока совсем немного узнали о цифровом звуке: как подготовить звуковой файл для использования в среде Flash и как организовать управление им в этой среде. Теперь пришло время вспомнить все, что вы знаете о звуке, и подробнее рассмотреть этот дополнительный ресурс, который можно редактировать вне среды Flash, чтобы научиться придавать вашим фильмам прекрасное звуковое оформление. Имея хорошие инструменты и освоив приемы их использования, вы откроете для себя целый мир новых творческих возможностей.

В этой главе рассмотрены следующие темы:

- ▶ основы работы со звуковым редактором;
- ▶ обзор приложений, обрабатывающих MIDI-звук;
- ▶ аудиоэффекты и приемы компьютерной обработки звука;
- ▶ создание плавных переходов при повторах звука;
- ▶ средства Flash для организации интерактивных повторений звука в фильмах.

## Звуковые редакторы

Сейчас стало доступно большое количество звуковых редакторов: от профессиональных систем обработки многоканального звука до простых любительских приложений. В зависимости от возможностей бюджета вы можете выбрать подходящее приложение для работы по созданию звукового сопровождения фильма.

Перечень рассмотренных в этом разделе приложений далеко не полон. Существует много других программных инструментов, при помощи которых можно создать прекрасное звуковое сопровождение для вашего фильма. Здесь мы рассмотрим приложения, которые наиболее часто используются для обработки звука. Выбор одного из них (или более), которое окажется для вас самым подходящим, может занять некоторое время. Уже в процессе выбора вы приобретете некоторый опыт обработки звука: выясните, исходя из стиля работы, имеющегося оборудования, сложности разрабатываемых проектов и т. д., наиболее полезные для вас возможности звукового редактора. В большинстве случаев производители программного обеспечения помогут вам сделать выбор. Многие компании

## Разработки

Для Web-анимаций важнейшим является качество изображения. Эта книга содержит множество интересных примеров Web-анимаций, созданных с помощью Flash. Одни из наиболее качественных и интересных — анимационные фильмы компании Renegade Cartoons ([www.renegadecartoons.com](http://www.renegadecartoons.com)), являющейся подразделением Renegade Animation ([www.renegadeanimation.com](http://www.renegadeanimation.com)).

Renegade Cartoons — небольшая компания, созданная Дареллом Ван Циттерсом и Эшли Поустлвейт, выпускает стильные фильмы, большинство из которых посвящено приключениям любвеобильного героя по имени Элмо Ардварак. На рис. 24.1 показана Web-страница узла компании.



Рис. 24.1. Renegade Cartoons производит стильную Flash-анимацию

## Резюме

В этой главе мы рассмотрели, как можно объединить возможности приложений Flash и Macromedia FreeHand. Вначале был приведен обзор программы FreeHand, затем вы научились осуществлять в ней многокадровую анимацию при помощи создания слоев и размещения на них изображения вручную, а также посредством создания эффекта перехода одного объекта в другой. Вы узнали, как можно просматривать анимацию в режиме Anti-Alias и задавать команды в документе, созданном в среде FreeHand. Наконец, вы научились экспортировать анимацию в формат Flash и изучили параметры экспорта.

- ▶ Autoplay (Автоматическое воспроизведение). Установка этого флажка задает автоматическое воспроизведение анимации при ее открытии в проигрывателе Flash.
- ▶ Full screen playback (Полноэкранное воспроизведение). Установка этого флажка задает полноэкранное воспроизведение анимации (до тех пор, пока пользователь не нажмет клавишу Esc). Следует отметить, что использование полноэкранного режима возможно только при отдельном просмотре фильма в проигрывателе Flash. Если фильм внедрен в какую-либо Web-страницу, этот режим не поддерживается.
- ▶ Compatibility (Совместимость). Список предназначен для указания версии Flash, которая будет поддерживать экспортируемую анимацию. Следует помнить, что в различных версиях Flash существуют разные уровни поддержки (или ее отсутствие) языка ActionScript.

---

#### ПРИМЕЧАНИЕ -

Так как программа FreeHand версии 10 была выпущена до Flash MX, наивысшей версией программы Flash, которую вы увидите в списке Compatibility, является версия 5. Но это не должно вас беспокоить, поскольку Flash MX может легко импортировать SWF-файлы Flash 5.

---

- ▶ Protect from import (Защита от импорта). Установка этого флажка запрещает импортировать содержимое SWF-файла.
- ▶ High quality printing (Высокое качество печати). Позволяет пользователю печатать кадры анимации с высоким разрешением. Если данный флажок не выбран, печать возможна только с экранным разрешением 72 точки на дюйм.

После установки всех нужных параметров экспорта нажмите кнопку ОК. Теперь необходимо экспортировать анимацию. Этот процесс рассматривается в следующем разделе.

## Экспорт файлов FreeHand во Flash

После задания параметров экспорта в окне Movie Properties вы можете осуществить экспорт фильма. Для этого выполните следующие действия.

1. Нажмите кнопку Test Movie, расположенную на панели Controller, или активизируйте команду Control ▶ Test Movie, после чего откроется окно Movie Player.
2. Нажмите кнопку Export Movie на панели Controller или активизируйте команду Control ▶ Export Movie (Управление ▶ Экспорт фильма).
3. В открывшемся окне Export Movie введите имя SWF-файла, выберите папку, в которую хотите его поместить, и нажмите кнопку Save (Сохранить).

---

#### СОВЕТ

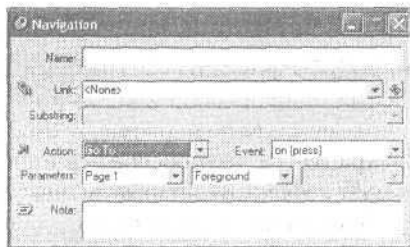
Существует и другой способ экспортировать фильм — активизировать команду File ▶ Export (Файл ▶ Экспорт). В открывшемся окне Export Movie выберите тип файла Macromedia Flash (SWF), введите имя SWF-файла и задайте папку, в которой вы хотите его сохранить, а затем нажмите кнопку Save.

---

Приведем описание элементов окна Movie Settings.

- ▶ Path Compression (Сжатие направляющей движения). Определяет степень сжатия направляющей движения в процессе экспорта. При выборе в этом списке элемента None первоначальное количество точек направляющей движения сохраняется, что обеспечивает исходное качество анимации, но в этом случае SWF-файл будет иметь максимальный размер. Выбор же элемента Maximum (наименьшее количество точек направляющей) позволяет получить SWF-файл минимального размера и анимацию самого низкого качества.
- ▶ Trace dashes strokes (Обработка пунктирных линий). Установка этого флажка гарантирует, что во время экспорта пунктирные линии векторных изображений будут преобразованы в отдельные объекты. При этом увеличивается время экспорта и размер файла.
- ▶ Image Compression (Сжатие изображений). Определяет степень сжатия (следовательно, и качество) при сохранении растровых изображений в формате JPG. Если в этом списке выбран элемент None, сохраняется исходное качество изображений, но файл имеет максимальный размер. Выбор же элемента Maximum позволяет получить файл минимального размера и самое низкое качество изображений.
- ▶ Text (Текст). Этот список позволяет задать параметры экспорта текста. В случае выбора значения Maintain blocks (Сохранять блоки) текст экспортируется в виде текстового блока, доступного для редактирования в программе Flash. В результате выбора значения Convert to paths (Преобразовать в кривые) символы преобразуются в кривые, что исключает возможность последующего редактирования текста. Если же выбрать значение None (Нет), то текст вовсе не будет экспортирован.
- ▶ Pages (Страницы). Этот переключатель предназначен для указания диапазона экспортируемых страниц многостраничного документа FreeHand.
- ▶ Animation (Анимация). Данный список позволяет указать способ экспорта созданной анимации. При выборе элемента
  - > Layers and Pages (Слои и страницы) вначале в виде последовательности кадров экспортируется содержимое слоев первой страницы документа, затем — второй и т. д.;
  - > Layers (Слои) содержимое всех слоев документа экспортируется в виде последовательности кадров;
  - > Pages (Страницы) содержимое каждой страницы экспортируется как один кадр в отдельный SWF-файл; если на странице имеется несколько слоев, они предварительно будут преобразованы в один;
  - > None (Нет) содержимое каждой страницы экспортируется в отдельный SWF-файл.
- ▶ Frame rate (Частота смены кадров). Поле предназначено для ввода значения частоты смены кадров анимации в единицах fps (frames per seconds — кадров в секунду). Если вы не помните, что такое частота смены кадров, обратитесь еще раз к главе 9.

- Откройте панель Navigation, выбрав пункт меню Window > Panels > Navigation (Окно > Панели > Навигация).
- В списке Actions (Действия) на этой панели выберите команду, которую хотите привязать к объекту. (Более подробная информация об этом находится в главе 12.)
- В списке Event (Событие) выберите событие, в результате которого должна выполняться команда. (Более подробная информация об этом находится в главе 12.)
- При выборе команды Go To, Print, Load/Unload Movie или Tell Target установите параметры команды в группе полей со списками Parameters. Первое поле служит для выбора страницы, второе — для выбора слоя.
- По окончании выбора закройте панель Navigation.



## Задание параметров Flash-фильма во FreeHand

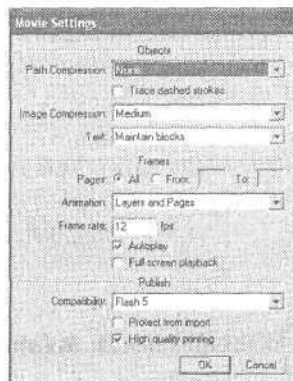
После формирования анимации в программе FreeHand необходимо экспортировать ее в SWF-формат. При экспорте FreeHand предоставляет удобные средства задания параметров SWF-файла.

### ПРИМЕЧАНИЕ

Для передачи анимации во Flash программа FreeHand может экспортировать ее только в SWF-формат, но не в формат FLA. В результате в программе Flash нельзя просто открыть изображение, созданное во FreeHand. Его следует импортировать посредством Flash в формат FLA.

Чтобы задать параметры экспорта анимации в SWF-файл, нужно открыть окно Movie Settings одним из указанных ниже способов:

- активизировать команду File > Export и в открывшемся окне Export Document выбрать элемент Macromedia Flash (SWF) из списка Save as Type, а затем нажать кнопку Setup;
- нажать кнопку Movie Settings, которая находится на панели Controller (к ней можно получить доступ, выбрав команду меню Window > Toolbars > Controller).



### ПРИМЕЧАНИЕ

Команда Export доступна только после нажатия кнопки Test Movie на панели Controller или активизации команды Control > Test Movie.



3. При воспроизведении фильма используйте для управления панель Controller. Если данная панель не отображается над рабочей областью среды FreeHand, выберите элемент меню Window ▶ Toolbars ▶ Controller (Окно ▶ Панели инструментов ▶ Контроллер).

Кроме инструментов, предназначенных для управления воспроизведением фильма, на данной панели располагаются три следующие кнопки.

- ▶ Test Movie — предварительный просмотр фильма (эквивалент команды Control ▶ Test Movie);
- ▶ Movie Settings — задание параметров фильма;
- ▶ Export Movie — экспорт фильма.

#### ПРИМЕЧАНИЕ

Программа FreeHand обеспечивает просмотр анимации при помощи проигрывателя Flash. Поэтому кнопки управления воспроизведением анимации становятся доступными только, когда проигрыватель запущен выбором команды Control ▶ Test Movie или нажатием кнопки Test Movie панели Controller.

## Использование режима Anti-Alias

Если вы создадите статичную анимацию во Flash, а затем такую же во FreeHand и сравните их, то заметите различия. Это объясняется тем, что хотя обе программы работают с векторной графикой, принципы построения этой графики в сравниваемых программах различаются. Ведь FreeHand предназначалась для подготовки изображений с целью вывода на печать, а Flash создана для работы с изображениями на экране монитора. Вы должны быть готовы к тому, что при импорте во Flash анимации, созданной во FreeHand, результат может не соответствовать вашим ожиданиям.

Давайте еще раз вернемся к анимации, воспроизводимой с помощью проигрывателя Flash. В обычном режиме редактировать ее нельзя, но можно воспользоваться режимом Anti-Alias, предоставляемым программой FreeHand. Он предназначен для работы с макетами, которые будут импортироваться в среду Flash.

Доступ к режиму Anti-Alias можно получить двумя способами. Во-первых, выбором пункта Flash Anti-Alias меню View, во-вторых, выбором элемента Flash Anti-Alias из списка Drawing Mode на панели состояния, расположенной под рабочей областью среды FreeHand.

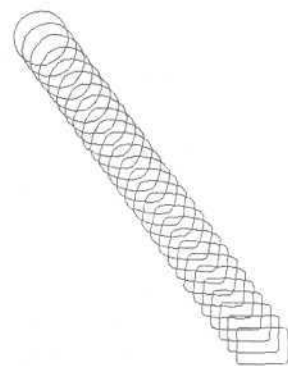
## Привязка команд к графическому объекту

Одним из достоинств программы FreeHand является возможность вставлять в проекты команды Flash. Еще до передачи в среду Flash можно вставить в локальную анимацию некоторые команды языка ActionScript.

Для этого необходимо выполнить следующие действия.

1. Выделите объект, к которому вы хотите привязать команды.

5. Выделите эти два объекта инструментом Pointer. Для этого охватите их рамкой выбора или, удерживая нажатой клавишу Shift, щелкните мышью на каждом из них.
6. Теперь активизируйте команду Modify ▶ Combine ▶ Blend (Модификация ▶ Комбинирование ▶ Переход). После этого будет автоматически создана серия промежуточных фигур для реализации перехода из первого объекта во второй.
7. Оставив эту серию фигур выделенной, откройте инспектор объектов (если его еще нет на экране), выберите команду меню Window ▶ Inspectors ▶ Object (Окно ▶ Инспекторы ▶ Объект).



В окне инспектора объектов введите в поле Steps значение для количества промежуточных объектов (по умолчанию 25). Таким образом будет задано количество промежуточных фигур, образующих переход. Учтите, чем больше шагов, тем более сглаженной будет анимация, но тем большим размер файла.

8. После задания количества промежуточных объектов активизируйте команду Xtras ▶ Animate ▶ Release to Layers (Xtrace > Анимировать ▶ В виде слоев), в результате чего откроется окно Release to Layers. Удостоверьтесь, что в списке Animate (Анимировать) выбран элемент Sequence (Последовательность), и нажмите кнопку ОК.

### СОВЕТ

Если вы хотите, чтобы переход осуществлялся в обратном направлении, установите в окне Release to Layers флажок Reverse Direction (Обратное направление).

Вы смоделировали анимацию перехода от одного объекта к другому, теперь осталось осуществить предварительный просмотр, после чего созданную анимацию можно будет экспортировать.

## Предварительный просмотр Flash-анимации во FreeHand

Завершив процесс создания покадровой анимации во FreeHand, вы, конечно, захотите увидеть, как она будет выглядеть после экспорта в SWF-файл. (О том, как производится экспорт в SWF-файл, будет рассказано в разделе «Экспорт файлов FreeHand во Flash» данной главы.)

Чтобы осуществить предварительный просмотр, выполните следующие действия.

1. Откройте документ с анимацией, предварительный просмотр которой вы хотите осуществить.
2. Активизируйте команду Control ▶ Test Movie (Управление ▶ Проверить фильм), вследствие чего анимация будет воспроизведена проигрывателем Flash внутри FreeHand.

## СОВЕТ

Для отображения линейки выберите команду меню View > Page Rulers ▶ Show (Вид ▶ Линейки ▶ Показать). Если вы хотите отредактировать единицы измерения на линейке, активизируйте команду View ▶ Page Rulers ▶ Edit (Вид ▶ Линейки ▶ Править), после чего откроется окно Edit Units (Правка единиц измерения), в котором можно выбрать единицы измерения.

6. Чтобы начать процесс анимации, преобразуйте созданный круг в символ. Для этого вызовите команду Modify ▶ Symbol ▶ Convert to Symbol (Модификация ▶ Символ > Преобразовать в символ) или нажмите клавишу F8, после чего символ будет автоматически вставлен в библиотеку FreeHand. Доступ к этой библиотеке можно получить, выбрав команду Library (Библиотека) меню Window (Окно).
7. Создайте новый слой, выполнив действия из пункта 3.
8. Открыв библиотеку, щелкните на символе круга, перетащите его и поместите на странице немного ниже первого.
9. Вот вы и создали два первых кадра, теперь просто повторяйте действия с 3 шага по 8 (то есть создавайте новые слои и перетаскивайте на них символ круга) до тех пор, пока символы не заполнят всю страницу по длине.



Поздравляем, вы создали вашу первую Flash-анимацию во FreeHand. Просмотреть ее вы сможете, изучив раздел «Предварительный просмотр Flash-анимации во FreeHand». Если результат вам не очень понравится, не огорчайтесь — ведь это только первый опыт.

## ПРИМЕЧАНИЕ —

Вместо использования отдельных слоев для создания кадров анимации можно применять отдельные страницы. Но таким образом нельзя создать описанную в следующем разделе анимацию с изменением формы.

# Моделирование анимации перехода

В предыдущем разделе описано создание многокадровой анимации с использованием слоев и символов. Теперь рассмотрим возможность изменения формы объекта в анимационном фильме. Хотя программа FreeHand в отличие от Flash не обладает возможностями для создания анимации типа *shape-tweening*, но позволяет имитировать изменение формы объектов путем задания перехода от одного объекта к другому. Сделать это можно следующим образом.

1. Создайте новый документ, выбрав команду File ▶ New или нажав комбинацию клавиш Cmd/Ctrl+N.
2. Создайте новый слой. Для этого откройте панель Layers, выбрав команду меню Window ▶ Panels ▶ Layers. В меню этой панели выберите команду New.
3. Выделите на панели Layers созданный слой и, используя инструмент Ellipse, нарисуйте большой круг в левом верхнем углу страницы.
4. В этом же слое при помощи инструмента Rectangle нарисуйте в нижнем правом углу маленький квадрат.

Пожалуй, наибольшим преимуществом является возможность совместного использования программ FreeHand и Flash. Обе они работают с векторной графикой, и в последних версиях FreeHand можно осуществлять экспорт созданных в этой программе проектов (даже таких, как многокадровая анимация) в среду Flash. Это делает программу FreeHand очень полезным средством для пользователей Flash.



Демонстрационная копия Macromedia FreeHand 10 находится на компакт-диске, прилагаемом к этой книге.

## Создание многокадровой анимации

В программе FreeHand можно создавать анимационные фильмы, а затем перенести их во Flash. Причем совместное использование программ допускает создание многокадровой анимации сразу во FreeHand и ее перенос для дальнейшего использования во Flash. То есть вы можете не создавать отдельные кадры, которые будут затем переданы один за другим, а перенести фильм целиком.

### ПРИМЕЧАНИЕ —

Поскольку FreeHand не обладает возможностями создания разветвленной динамически управляемой анимации, в некоторых случаях следует экспортировать созданную в этой программе простую покадровую анимацию во Flash для дальнейшей работы.

Теперь перейдем к практическому освоению покадровой анимации программы FreeHand и выполним несложное задание — создадим движущуюся сферу. Это даст вам необходимые навыки для работы с более сложными проектами. Выполните следующие действия.

1. Запустите программу FreeHand и создайте новый документ, вызвав команду File ▶ New (Файл ▶ Новый) или нажав комбинацию клавиш Cmd/Ctrl+N.
2. Задайте масштаб, при котором страница будет отображаться полностью. Для этого активизируйте команду View ▶ Fit All (Вид ▶ Показать все). По умолчанию во FreeHand размер нового документа равен 8,5x11 дюймов.
3. Перед созданием объекта необходимо создать новый *слой*. Выберите команду Window ▶ Panels ▶ Layers (Окно ▶ Панели ▶ Слои), после чего откроется панель Layers. Разверните меню данной панели, нажав в верхнем правом углу на стрелку, указывающую вправо, и щелкните на команде New.

### ПРИМЕЧАНИЕ —

При импорте анимации во Flash содержимое слоев анимации FreeHand преобразуется в отдельные кадры.

4. Создайте объект, который вы будете анимировать. Для этого в панели инструментов выделите инструмент Oval (если панель отсутствует на экране, выберите команду Tools меню Window).
5. Установите указатель мыши в верхней части страницы. Нажмите кнопку мыши и перемещайте указатель таким образом, чтобы создать круг диаметром 3 сантиметра.



## Глава 24

# Совместная работа Flash и FreeHand

В главе 4 вы познакомились с инструментами Flash для рисования и научились создавать с их помощью графические объекты. Но при разработке сложных приложений может возникнуть ситуация, когда одних инструментов Flash окажется недостаточно. В этом случае следует воспользоваться возможностями еще одной мощной программы фирмы Macromedia — FreeHand, являющейся одним из лидеров среди программ создания векторной графики.

Изучив данную главу, вы получите представление о программе FreeHand и научитесь использовать ее совместно с программой Flash. В главе рассмотрены возможности программы FreeHand, позволяющие:

- ▶ создавать многокадровую Flash-анимацию;
- ▶ моделировать анимацию перехода;
- ▶ предварительно просматривать Flash-анимацию;
- ▶ использовать режим Anti-Alias;
- ▶ создавать привязку команд к графическому объекту;
- ▶ устанавливать параметры Flash-фильма;
- ▶ экспортировать файлы FreeHand во Flash.

## Общие сведения о программе FreeHand

До появления цифрового мультимедиа одним из основных информационных средств была печать. Конечно, и сейчас никто не отрицает ее огромных возможностей, но многие программные средства, которые ранее использовались только для создания печатного слова, теперь нашли свое применение в цифровых дизайнерских проектах. Мы имеем в виду последние версии таких программ, как QuarkXPress, Adobe Illustrator, Adobe PageMaker, которые позволяют экспортировать публикации в формате HTML.

Не осталась в стороне и компания Macromedia, создав программу FreeHand, работающую с векторной графикой. Хотя эта программа в первую очередь предназначена для создания изображений в печатных публикациях, она может также в полной мере использовать возможности других средств цифрового дизайна. Существенно увеличивает ее возможности то, что специалисты компании Macromedia, разработавшие такие мощные мультимедийные инструменты, как Dreamweaver, Dreamweaver UltraDev, Fireworks и Flash, позаботились о создании серии программ, которые позволяют совместно использовать инструменты мультимедиа и FreeHand.

## Разработки

Возможности, предоставляемые Flash и Director, очень велики. Чтобы увидеть приложение, созданное с помощью Flash и Director, посетите Web-узел [www.lego.com/build](http://www.lego.com/build). Здесь (рис. 23.3) представлена одна из наиболее забавных игр — Junkbot, разработанная компанией New York City's gameLab. В этой игре используется комбинация элементов Flash и среды Director (Shockwave) — игра создана в среде Director, а ее предисловие анимировано во Flash.



Рис. 23.3. Junkbot — онлайн-игра, созданная при помощи программ Director и Flash

## Резюме

В этой главе описаны возможности совместного использования двух мощных мультимедийных приложений: Macromedia Flash и Macromedia Director. Мы рассказали об использовании SWF-файлов в среде Director и импортировании Flash-фильмов в эту среду. Также были рассмотрены параметры, определяющие поведение Flash-фильмов в среде Director, и некоторые другие настройки.

Одним из наибольших преимуществ комбинирования является расширение возможностей сценариев. Так же как Flash, Director имеет свой язык написания сценариев — Lingo. Мы рассказали, как при помощи этого языка можно управлять элементами Director-фильма, и ознакомили вас с возможностью использования языков ActionScript и Lingo для передачи информации между фильмами.

Этот блок сценария управляет указанным экземпляром клипа. Вспомните, в главе 16 мы говорили, что во Flash клип может содержать вложенные экземпляры других клипов, каждый из которых является отдельной частью главного клипа, поэтому для управления каждым экземпляром клипа требуется отдельный блок кода в сценарии. То же справедливо и для Director-фильма: используя сценарии Lingo, можно применять масштабирование к компонентам ядра, основанное на векторной графике, и управлять их воспроизведением.

С помощью Lingo спрайтами Flash управляют так же, как другими элементами Director-фильма, — масштабируют, поворачивают и т. д. Язык Lingo предоставляет набор свойств спрайта, значения которых можно устанавливать и возвращать, для настройки характеристик Flash-спрайта в Director-фильме. Рассмотрим синтаксис команды для установки свойства спрайта:

```
sprite(номер_спрайта).свойство=значение
```

В Lingo сначала следует указать номер спрайта, которым вы хотите управлять, затем — свойство, значение которого следует изменить, а далее — новое значение этого свойства. Рассмотрим следующий пример:

```
on mouseUp me
 spin=sprite(me.spriteNum).rotation
 sprite(me.spriteNum).rotation=spin+90
end
```

Здесь спрайт, к которому привязан сценарий, поворачивается на 90 градусов при каждом щелчке мыши на области спрайта. Текущее значение угла поворота спрайта сохраняется в переменной *spin*. Далее к этому значению прибавляется еще 90 градусов.

В табл. 23.3 приведены некоторые свойства Lingo, используемые для управления спрайтами Flash.

**Таблица 23.3.** Свойства Lingo для управления спрайтами Flash

| Свойство | Описание                                                                                                             |
|----------|----------------------------------------------------------------------------------------------------------------------|
| rotation | Задаёт значение угла поворота спрайта в градусах                                                                     |
| scale    | Задаёт значение масштабирования спрайта относительно исходного размера                                               |
| locH     | Управляет горизонтальным положением спрайта на области действия, начиная с левого верхнего угла в пикселах           |
| locV     | Управляет вертикальным положением спрайта на области действия, начиная с левого верхнего угла в пикселах             |
| locZ     | Управляет порядком слоев спрайтов на панели Score                                                                    |
| flipV    | Определяет, было ли изменено горизонтальное положение спрайта на области действия; если да, возвращает значение true |
| flipH    | Определяет, было ли изменено вертикальное положение спрайта на области действия; если да, возвращает значение true   |
| skew     | Устанавливает значение угла наклона для спрайта, выраженное в градусах                                               |
| blend    | Устанавливает значение прозрачности спрайта: 100 — полностью непрозрачный, 0 — прозрачный                            |

Полный список с описанием всех свойств спрайта находится в документации, поставляемой с Director и Lingo.

Для управления воспроизведением фильма главной временной шкалы и экземпляра клипа во Flash-фильме используются действия `tellTarget()` и `endTellTarget()`. И хотя применение данных действий во Flash согласно новой документации, требуется свести к минимуму, в среде Director они используются для указания на экземпляры клипа. Вот их синтаксис в Lingo:

```
on mouseDown
 sprite(номер_спрайта).tellTarget("путь_к_клипу")
 --некоторые инструкции для указания пути к требуемому клипу
 sprite(номер_спрайта).endTellTarget()
end
```

В шаблоне сценария команды `tellTarget()` и `endTellTarget()` используются как начало и конец «книжки», а в середину вкладываются команды управления клипом или фильмом главной временной шкалы. В команде `tellTarget()` необходимо указать путь к клипу, для чего применяется синтаксис с использованием символа `/`. Рассмотрим следующий пример:

```
on mouseDown
 sprite(1).tellTarget("/menu")
 sprite(1).play()
 sprite(1).endTellTarget()
end
```

С помощью данного кода воспроизводится клип `menu`. Тот же сценарий в ActionScript выглядит следующим образом:

```
on(press){
 _root.menu.play();
}
```

Ниже приведены *несколько* команд Lingo, используемых для управления экземплярами клипа во Flash-фильмах.

- ▶ `play()`. При вызове команды начинается воспроизведение клипа. В ActionScript команда соответствует действию `play()`.
- ▶ `stop()`. При вызове команды останавливается воспроизведение клипа. В ActionScript команда соответствует действию `stop()`.
- ▶ `gotoFrame()`. При вызове команды головка воспроизведения в клипе перемещается к заданному кадру. В ActionScript команда соответствует действию `gotoAndStop()`.

Если вы хотите, чтобы при выполнении команды `gotoFrame()` осуществлялся переход головки воспроизведения к нужному кадру и восстанавливалось воспроизведение с этого кадра (как при выполнении действия `gotoAndPlay()`), можете использовать сценарий, описанный ниже. С помощью данного сценария головка воспроизведения экземпляра клипа `menu` перемещается к кадру 10 и возобновляется воспроизведение клипа с десятого кадра.

```
on mouseDown
 sprite(1).tellTarget("/menu")
 sprite(1).gotoFrame(10)
 sprite(1).play()
 sprite(1).endTellTarget()
end
```



Этот сценарий можно использовать для проверки значения прозрачности (`#alpha` в Lingo, `_alpha` в ActionScript) экземпляра клипа `topleft` Flash-фильма, находящегося в спрайте с номером 1. Если свойство `#alpha` имеет значение 100, оно устанавливается в 50, если отлично от этого значения, то устанавливается в 100. При использовании функции `getFlashProperty()` вам не требуется устанавливать значение свойства, просто примените пустую строку ("") для этого аргумента.

В табл. 23.2 указаны все свойства Lingo, используемые для манипулирования свойствами Flash-фильма в среде Director, и приведено описание каждого из них.

**Таблица 23.2.** Свойства Lingo для манипулирования свойствами Flash-фильмов

| Имя                                 | Тип* | Описание                                                                                                       |
|-------------------------------------|------|----------------------------------------------------------------------------------------------------------------|
| <code>#posX</code>                  | RW   | Положение экземпляра клипа по оси X в пикселах                                                                 |
| <code>#posY</code>                  | RW   | Положение экземпляра клипа по оси Y в пикселах                                                                 |
| <code>tfscaleX</code>               | RW   | Значение масштабирования экземпляра клипа по оси X в пикселах                                                  |
| <code>#scaleY</code>                | RW   | Значение масштабирования экземпляра клипа по оси Y в пикселах                                                  |
| <code>#visible</code>               | RW   | Определяет видимость экземпляра клипа                                                                          |
| <code>#rotate</code>                | RW   | Значение угла поворота для экземпляра клипа в градусах                                                         |
| <code>#alpha</code>                 | RW   | Процентное значение прозрачности экземпляра клипа                                                              |
| <code>#name</code>                  | RW   | Имя экземпляра клипа                                                                                           |
| <code>#width</code>                 | R    | Ширина области отображения экземпляра клипа в пикселах                                                         |
| <code>tfheight</code>               | R    | Высота области отображения экземпляра в пикселах                                                               |
| <code>#target</code>                | R    | Полный путь к экземпляру клипа                                                                                 |
| <code>#url</code>                   | R    | Полный путь к экземпляру клипа в формате HTTP                                                                  |
| <code>#dropTarget</code>            | R    | Полный путь для клипа, на который был помещен другой перемещаемый клип                                         |
| <code>#totalFrames</code>           | R    | Общее количество кадров в клипе                                                                                |
| <code>#currentFrame</code>          | R    | Положение головки воспроизведения на временной шкале клипа                                                     |
| <code>#lastFrameLoaded</code>       | R    | Последний кадр, загруженный для спрайта в память (смотрите свойство <code>_framesLoaded</code> в ActionScript) |
| <code>#focusRect</code>             | RWG  | Управляет видимостью прямоугольного фокуса кнопки во Flash-фильме                                              |
| <code>#spriteSoundBufferTime</code> | RWG  | Определяет время воспроизведения звука в секундах до начала воспроизведения                                    |

\* R — свойства, которые возвращают значения; W — свойства, которые устанавливают значения; G — глобальное свойство для Flash-фильма.

#### ПРИМЕЧАНИЕ

В табл. 23.2 приведены два глобальных свойства, которые применяются ко всему фильму. При работе с ними в качестве аргумента нужно вводить пустую строку, например, `sprite(1).getFlashProperty("", #focusRect, "")`.

воспроизведения фильма к указанному кадру. В данном примере головка воспроизведения будет перемещена к кадру с меткой `start`.

Это напоминает предыдущий пример, где применялась конструкция `on getURL me`. Но существуют и определенные отличия: когда используется протокол событий, Flash обращается к созданной вами функции в Director-фильме. Преимуществом такого подхода является то, что функцию не нужно связывать с каждым Flash-спрайтом, поскольку она находится в сценарии клипа Director-фильма и всегда доступна Flash-фильмам.

Действие ActionScript `getURL()` предоставляет другой способ передачи информации из Flash в Director — с помощью протокола `lingo`. Благодаря использованию этого протокола информацию в Director-фильм можно передавать напрямую из Flash-фильма. Например:

```
on(release){
 getURL("lingo:go to frame 10")
}
```

Если связать этот сценарий с Flash-кнопкой, он будет передавать сообщение в Director, указывая головке воспроизведения перейти к кадру номер 10. Поскольку протокол `lingo` использует команды Lingo, не требуется дополнительных инструкций в Director.

## Манипулирование спрайтами Flash при помощи Lingo

Когда спрайты, управляемые командами Lingo, содержатся в Director-фильме, Flash-фильмы теряют свою специфичность, но зато появляется возможность управлять спрайтами. Для осуществления навигации в фильме можно использовать команды `tellTarget()`, `endTellTarget()` и основные команды, применяемые к временной шкале: `play()`, `stop()` и `gotoFrame()`. Для установки и получения значений свойств элементов Flash-фильма в среде Director используйте функции `setFlashProperty()` и `getFlashProperty()`. Первая из них возвращает текущее значение свойства элементу Flash-фильма, а вторая задает новое значение свойства. Например, если Flash-фильм содержит клип, вы можете использовать эти функции, чтобы контролировать значения свойств клипа и при необходимости изменять их. Вот синтаксис этих функций:

```
sprite(номер_спрайта).setFlashProperty("имя_клипа", #свойство, новое_значение)
sprite(номер_спрайта).getFlashProperty("имя_клипа", #свойство, значение)
```

Сначала нужно указать номер спрайта Flash-фильма, затем в каждой функции указать имя нужного клипа, свойство, значение которого вы хотите задать или получить, и его новое значение. Рассмотрим следующий пример:

```
on mouseUp
 if sprite(1).getFlashProperty("topleft",#alpha,"")=100
 then
 sprite(1).setFlashProperty("topleft",#alpha,50)
 else
 sprite(1).setFlashProperty("topleft",#alpha,100)
 end if
end
```

В этом примере `directorMarker` является меткой кадра на панели Score, к которому вы хотите переместить головку воспроизведения после нажатия созданной во Flash кнопки. Flash передает в Director информацию как строку.

Наконец, необходимо добавить несколько команд Lingo, которые обеспечат прием строки `directorMarker` из Flash в Director. В среде Director создайте сценарий и свяжите его с Flash-спрайтом, содержащим кнопку. Сценарий, написанный на Lingo, будет выглядеть так:

```
on getURL me, flashStringInfo
 go to frame flashStringInfo
end
```

В этом коде спрайт получает строковые данные и сохраняет их в значении переменной `flashStringInfo`. Эту переменную можно позже использовать в сценарии. Значение переменной `flashStringInfo` содержит метку кадра (строковое значение) Director-фильма. Если в вашем Flash-фильме несколько кнопок с различными аргументами, которые необходимо передать в Director-фильм, то этот сценарий должен содержать соответствующее количество переменных, в которых будет сохраняться каждое строковое значение аргумента.

Действие `getURL()` также позволяет определять некоторое событие в среде Director. Например, передвижение по интерактивному фильму — это событие, повторяющееся по ходу фильма. Вы можете написать собственный обработчик события для прямой навигации таким образом, что каждый раз при нажатии Flash-кнопки Director-фильм будет перемещаться на заданную позицию. Для этого вам нужно связать действие `getURL()` с кнопкой, но теперь необходимо задать событие и аргумент.

```
on(release){
 getURL("event:flashButton \"start\"")
}
```

В этом примере протокол `event` указывает на событие с именем `flashButton`. Аргументом для этого события является `start`.

#### ПРИМЕЧАНИЕ

В этом примере аргумент `start` должен содержать разделители (символ обратной косой черты), чтобы Director мог правильно интерпретировать строку с меткой кадра.

Для работоспособности данного сценария необходимо, чтобы событие, вызванное во Flash, произошло в среде Director. В противном случае событие не будет идентифицировано и выполнено в Director-фильме. Функция для Director создается посредством написания с помощью Lingo и сохранения в сценарии клипа. В этом примере функция должна выглядеть следующим образом:

```
on flashButton me. swfString
 go to frame swfString
end
```

В этом фрагменте кода событие `event`, определенное во Flash, является обработчиком `flashButton`. Аргумент `start` передается как строка и сохраняется в значении переменной `swfString`, а затем используется в сценарии для перемещения головки

```

 end if
end

```

Здесь в условии используется свойство `frame` и при каждом повторении кадра производится проверка того, является ли номер воспроизводимого кадра (`sprite(1).frame`) меньшим, чем общее число кадров во Flash-фильме (`sprite(1).member.frameCount`), который содержится в спрайт-канале с номером 1. Если это так, воспроизведение фильма продолжается, если нет, головка воспроизведения переходит к кадру с меткой `done`.

## Работа со сценариями ActionScript

Использование свойства `actionsEnabled` позволяет активизировать или отключать действия ActionScript.

Рассмотрим следующий пример:

```

on mouseDown
 if sprite(1).actionsEnabled=1 then
 sprite(1).actionsEnabled=0
 else
 sprite(1).actionsEnabled=1
 end if
end

```

Этот сценарий можно связать с кнопкой и применять ее как «выключатель» действий ActionScript Flash-фильма, помещенного в спрайт-канал с номером 1. Использование структуры условного оператора `if...end` позволяет проверить состояние действий ActionScript для первого спрайта: если они включены, то будут выключены, и наоборот. Это свойство может быть полезно, когда в фильме используется действие `gotoAndPlay()`. С помощью выражения `actionsEnabled=1` эти сценарии (и все другие тоже) во Flash-фильме игнорируются, и он воспроизводится непрерывно.

## Передача информации из Flash в Director

Иногда требуется передать информацию из Flash-фильма в Director-фильм непосредственно. Например, если вы создали во Flash навигационную панель с анимированными кнопками и хотите использовать ее, чтобы управлять воспроизведением Director-фильма. В таком случае необходимо для каждой кнопки создать сценарий, где в качестве аргументов соответствующих команд указать, к какому кадру Director-фильма следует перейти головке воспроизведения, и передать эти аргументы в Director-фильм. Выполнить это можно при помощи действия `getURL()` языка ActionScript. Данное действие передает строку информации из Flash-фильма, где это действие вызывается, в Director-фильм, который служит хостом для Flash-фильма.

Чтобы создать во Flash-фильме кнопку, после щелчка на которой информация будет передаваться в Director-фильм, выберите кнопку на области действия и откройте панель Actions. Введите следующий код:

```

on(release){
 getURL("directorMarker")
}

```

Данный сценарий, связанный к какой-либо кнопкой, остановит воспроизведение спрайта `sprite(1)` после щелчка мыши на указанной кнопке.

## Команда `rewind`

Для перемещения головки воспроизведения Flash-фильма к первому кадру применяется команда `rewind()`.

```
on mouseUp
 sprite(1).rewind()
end
```

Этот сценарий может быть связан с элементом управления воспроизведением фильма (кнопкой перемотки) или использован как часть сценария перемотки.

Изначально в Lingo нет команды ускорения воспроизведения Flash-фильмов, но ее можно создать. Рассмотрите следующий сценарий:

```
on exitFrame
 if gFwd=true then
 swfFrame=sprite(1).frame
 sprite(1).goToFrame(swfFrame+2)
 end if
end
```

Здесь номер текущего кадра Flash-фильма сохраняется в значении переменной `swfFrame`, затем с помощью команды `goToFrame()` производится переход на два кадра вперед. Эти действия повторяются для каждого текущего кадра, если переменная `gFwd` имеет значение `true`. Вы можете связать данный сценарий с кнопкой, после щелчка на которой фильм будет воспроизводиться в ускоренном режиме (в данном случае через два кадра).

## Свойства `frameCount` и `frame`

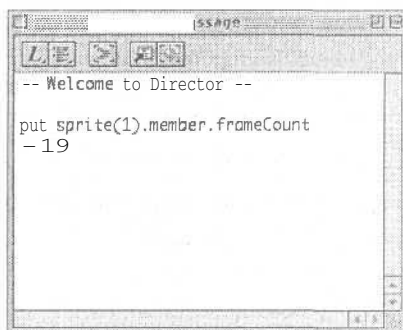
Свойство `FrameCount` возвращает общее количество кадров в фильме. Например, с помощью инструкции

```
put sprite(1).member.frameCount
```

в окне `Message` будет выведено общее количество кадров Flash-фильма, которые содержатся в спрайт-канале с номером 1. Если Flash-фильм имеет в спрайт-канале 19 кадров, в окне `Message` будет выведена приведенная на рисунке справа информация.

Свойство `frame` возвращает номер текущего кадра в процессе воспроизведения фильма. В сочетании со свойством `FrameCount` его можно использовать, например, для перехода к новому кадру после окончания воспроизведения Flash-фильма. Например:

```
on exitFrame
 if sprite(1).frame<sprite(1).member.frameCount then
 go to the frame
 else
 go to "done"
```



## Комбинирование языков ActionScript и Lingo

Теперь, когда вам знакомы технические аспекты работы со спрайтами и компонентами ядра, мы можем перейти к изучению основ языка Lingo. Комбинируя Flash и Director, вы не только объединяете две мощные мультимедийные программы, но и позволяете им взаимодействовать друг с другом. Используя такое взаимодействие, можно создавать приложения с огромными интерактивными возможностями.

Lingo является достаточно гибким языком программирования, при помощи которого можно свободно управлять элементами Flash, входящими в Director-фильм. В этом разделе представлены основные команды Lingo для работы с Flash-фильмами. За более детальной информацией обращайтесь к файлам справки Lingo (Help ▶ Lingo Dictionary).

### Управление воспроизведением при помощи Lingo

Довольно часто Flash-компоненты ядра используются как вложенные элементы главного фильма. В среде Director с помощью Lingo можно управлять вложенными Flash-фильмами точно так же, как во Flash с помощью ActionScript можно управлять клипами, являющимися независимой анимацией. Кроме того, владея основами Lingo, вы сможете создавать собственные сценарии для различных элементов управления. Ниже приводятся некоторые основные команды этого языка, которые позволят вам управлять воспроизведением Flash-фильма.

#### Команда play

Для того чтобы начать воспроизведение Flash-фильма, воспользуйтесь командой play O. Например:

```
on mouseUp
 sprite(1).play()
end
```

Этот сценарий, связанный с какой-либо кнопкой, будет воспроизводить спрайт sprite(1) после щелчка на указанной кнопке.

#### СОВЕТ

---

Если вы не хотите, чтобы Flash-фильм начал воспроизводиться сразу на области действия, активизируйте флажок Paused панели Property Inspector либо воспользуйтесь свойством pausedAtStart.

---

#### Команда stop

Команда stop O используется для остановки воспроизведения Flash-компонента ядра в его текущей позиции. Например:

```
on mouseUp
 sprite(1).stop()
end
```

- ▶ Местоположение. Если для Flash-фильма активизирован флажок Direct to Stage, то этот фильм всегда помещается поверх всех остальных спрайтов. Используйте этот флажок для наиболее важных фильмов, которые должны воспроизводиться с наивысшим приоритетом. Для всех остальных фильмов этот флажок лучше отключить.
- ▶ Эффекты Ink. Эффекты Ink являются установками, которые применяются к спрайтам для изменения их внешнего вида на области действия. Значения Blend и эффекты Ink можно установить в области Sprite Toolbar панели Score (рис. 23.2). Доступ к эффектам Ink вы получите, нажав клавишу Cmd/Ctrl и щелкнув на спрайте. После этого появится раскрывающееся меню, где можно выбрать один из указанных элементов:
  - Copy. При выборе этой команды Flash-фильм будет последовательно, пиксел за пикселом, отображаться так, как он был создан. Это команда по умолчанию.
  - Background Transparent. С помощью этой команды можно сделать фоновый цвет спрайта невидимым (прозрачным). При этом вы сможете увидеть содержимое, находящееся ниже элементов спрайта.
  - Transparent. Данная команда позволяет применять размывку цветов. Этот эффект лучше использовать с темным и светлым оттенками цвета.
  - Blend. Назначение этого поля аналогично назначению свойства `_alpha` в языке ActionScript. В поле указывается прозрачность для каждого спрайта.

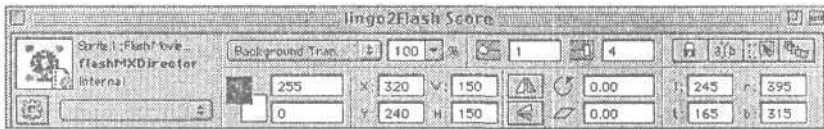


Рис. 23.2. Область Sprite Toolbar панели Score

Для большинства Flash-фильмов рекомендуем применять команду Copy, поскольку при этом вы сэкономите ресурсы памяти.

- ▶ Звук. И во Flash-фильме и в Director-фильме использование звуковых потоков является очень сложной процедурой, поскольку оба приложения разделяют звуковые каналы. Если вы работаете в Windows, то можете воспользоваться глобальным свойством `soundMixMedia`. Когда значение этого свойства равно `true`, Director смешивает свои звуковые каналы, содержащиеся на панели Score, со звуковыми каналами Flash-компонента ядра. При задании значения `false` звуки Flash и Director проигрываются отдельно.

## СОВЕТ

Смешивание звука Flash-фильма со звуком Director-фильма может привести к нежелательным результатам. Поэтому рекомендуем вам производить обязательную проверку, чтобы быть уверенными в возможности управлять всеми звуковыми элементами с помощью среды Director и языка Lingo.

## Использование Flash-фильмов в качестве спрайтов

Для того чтобы Flash-компонент ядра стал элементом Director-фильма, нужно поместить этот компонент на панель Stage (путем перетаскивания или с помощью Lingo). Пока это не сделано, компонент ядра не принимает никакого участия в фильме. После помещения компонента ядра на область действия он становится спрайтом и автоматически получает номер, соответствующий следующему свободному спрайт-каналу панели Score. Например, если на области действия есть шесть спрайтов, и вы перетащили еще один компонент ядра, то он становится спрайтом, имеющим номер 7 (`sprite(7)`). Этот номер очень важен, поскольку он идентифицирует Flash-фильм, и при обращении к данному фильму достаточно указать его номер. Например, используя инструкцию `sprite(7).visible=0`, вы делаете спрайт, имеющий номер 7, невидимым. Кроме того, спрайт можно создать непосредственно, перетащив компонент ядра на панель Score, что позволит поместить спрайт в нужный вам спрайт-канал. После этого точка регистрации Flash-фильма будет автоматически центрирована в области действия.

Со спрайтом можно производить следующие действия.

- ▶ **Воспроизведение.** Количество кадров Flash-фильма не должно превышать количество кадров Director-фильма, лишние кадры просто не будут воспроизводиться. Например, если Flash-фильм содержит 100 кадров, а Director-фильм 50, то воспроизводиться будут только 50 кадров Flash-фильма. Для полного воспроизведения Flash-фильм нужно привязать к кадру Director-фильма с помощью следующего кода:

```
on exitFrame
 do to the frame
end
```

При воспроизведении в Director-фильме кадра, с которым связан данный сценарий, Flash-фильм будет воспроизводиться полностью. Если в дополнение к этому активизировать в окне Flash Asset Properties флажок Loop, можно получить циклическое воспроизведение Flash-фильма в Director-фильме. Подробнее об этом рассказано в разделе «Управление воспроизведением при помощи языка Lingo».

- ▶ **Векторный формат.** Векторную графику удобно использовать, когда во Flash-фильме необходимо изменить размеры графического изображения или текста без искажения и потери качества. В среде Director с помощью Lingo это можно делать динамически, основываясь на интерактивном взаимодействии зрителей и фильма.
- ▶ **Манипуляция спрайтами.** Вы можете размещать Flash-фильм на области действия как спрайт. Более подробно об этом рассказывается в следующем разделе.



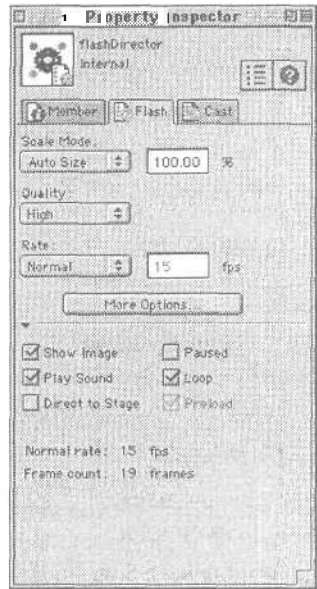
## Rate

Используя этот список, можно изменить скорость воспроизведения (частоту смены кадров) Flash-фильма. Во Flash вы не можете выполнить аналогичные изменения, поэтому есть смысл комбинировать Flash и Director. Список Rate содержит три элемента:

- ▶ установка элемента Normal сохраняет частоту смены кадров исходного SWF-файла;
- ▶ с помощью элемента Lock-Step кадр за кадром устанавливается одинаковая частота смены кадров Director-фильма и Flash-фильма. Если выбрать этот элемент и установить для Director-фильма частоту смены кадров, равную 30 fps, то и Flash-фильм будет иметь такую же частоту;
- ▶ при выборе элемента Fixed активизируется поле fps (количество кадров в секунду), позволяющее установить нужное значение частоты смены кадров для Flash-фильма. Но скорость воспроизведения Flash-фильма не может быть больше, чем частота смены кадров установленная в Director-фильме.

Помимо окна Flash Asset Properties в среде Director есть еще несколько средств изменения параметров Flash-компонента ядра. Одним из них является панель Property Inspector, очень похожая на панель Properties среды Flash. На этой панели можно получить доступ ко многим настройкам, которые определяют уникальные параметры для каждого элемента или компонента ядра вашего фильма: масштабирование, изменение цвета, имя спрайта и др.

Чтобы отобразить панель Property Inspector с установленными параметрами для Flash-компонента ядра, выберите компонент на панели Internal Cast и щелкните на кнопке Property Inspector (с изображением буквы i) панели инструментов. Эта кнопка присутствует и в некоторых других элементах интерфейса среды Director и служит для вызова панели Property Inspector. Данная панель используется для настройки воспроизведения, а также задания значений следующих параметров Flash-фильма:



- ▶ имени компонента ядра и пути к нему;
- ▶ режима масштабирования;
- ▶ качества;
- ▶ частоты смены кадров.

После щелчка на кнопке More Options откроется окно Flash Asset Properties, в котором можно установить дополнительные параметры, недоступные на панели Property Inspector.

- ▶ с помощью флажка Sound осуществляется управление звуком Flash-фильма. Если флажок активизирован, звук фильма включен; если нет, звук выключен;
- ▶ флажок Paused управляет воспроизведением фильма при первой загрузке на панели Score. При установленном флажке воспроизведение фильма остановится на первом кадре. Для продолжения воспроизведения необходимо указать соответствующую команду языка Lingo или воспользоваться соответствующим элементом управления фильмом. Когда флажок сброшен, фильм начинает воспроизводиться сразу после загрузки на область действия;
- ▶ флажок Loop определяет, будет ли воспроизведение фильма повторяющимся. Если флажок активизирован, фильм воспроизводится циклически, если сброшен, воспроизведение фильма прекратится после однократного воспроизведения;
- ▶ флажок Direct to Stage определяет приоритет, который Director назначает Flash-компоненту ядра. Если этот флажок установлен, Director использует все ресурсы, чтобы при воспроизведении Flash-фильм сглаживался. Если этот флажок не активизирован, то Flash-спрайт будет установлен поверх всех остальных элементов Director-фильма.

### Quality

Среда Director содержит несколько настроек, управляющих качеством воспроизведения Flash-компонентов ядра. Эти настройки аналогичны настройкам вкладки HTML в окне Publish Settings программы Flash. Раскрывающийся список Quality содержит следующие элементы: High, Auto-High, Low и Auto-Low. Более подробная информация содержится в главе 27.

### Scale

В данном поле вы можете установить значение масштабирования для Flash-компонента ядра. Это значение будет использоваться каждый раз при воспроизведении Flash-фильма в Director-фильме.

### Scale Mode

В этом раскрывающемся списке задается режим масштабирования для Flash-компонентов ядра:

- ▶ элемент Auto-Size установлен по умолчанию. С его помощью блокируется масштабирование области отображения Flash-фильма. Если изменить размер спрайта, изменится и размер области отображения Flash-фильма. По умолчанию при выбранном элементе Auto-Size в поле Scale установлено значение масштабирования 100 %. Если изменить это значение, изменится и масштаб области отображения Flash-фильма;
- ▶ элемент No Scale сохраняет значение масштабирования Flash-компонента ядра, которое указано в поле Scale. Независимо от размера спрайта этот элемент всегда устанавливает размер равным значению, заданному в поле Scale;
- ▶ назначение остальных элементов, Show All, No Border и Exact Fit, аналогично назначению элементов списка Scale вкладки HTML окна Publish Settings среды Flash. Эти настройки подробно рассмотрены в главе 27.

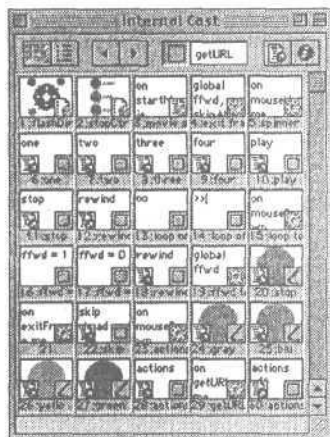
Кроме описанных кнопок и методов Director предоставляет и другие настройки импорта, которые в этом разделе мы не рассматриваем. В случае необходимости обращайтесь за **дополнительной** информацией к файлам справки среды Director (Help ► Director Help).

## Свойства Flash-фильма

В случае успешного импорта Flash-фильма в Director-проект этот фильм появится как компонент ядра на панели Internal Cast. Director автоматически включает модуль Flash Asset Xtra, который нужен для поддержки Flash-компонента ядра, поэтому вручную это делать не нужно.

Начиная с этого момента вы можете делать с фильмом всевозможные манипуляции, поскольку Flash-компоненты ядра являются чем-то уникальным в среде Director — они являются интерактивными мультимедийными фильмами в интерактивном мультимедийном фильме Director.

Существует несколько параметров, определяющих поведение компонентов ядра в среде Director. Эти параметры устанавливаются в окне Flash Asset Properties, для доступа к которому дважды щелкните на значке компонента ядра на панели Internal Cast. С помощью этого окна можно управлять многими параметрами Flash-фильма: качеством воспроизведения, параметрами масштабирования, звуковыми настройками и др. В нем представлены следующие настройки для управления компонентом ядра.



### Media

Если для связи с Flash-файлом вы используете кнопку Browse, в поле Import будет отображаться путь к файлу, где бы он не находился: на вашем жестком диске или в Интернете.

Flash-фильмы можно как импортировать, так и связывать. Если вы активизировали флажок Linked, файл Flash-фильма будет связан с его местоположением. При воспроизведении Director-фильм обратится к SWF-файлу по указанному пути и отобразит на области действия кадры связанного фильма. Если установить флажок Preload (Предварительная загрузка), то перед воспроизведением Director-фильма в память полностью загрузится связанный SWF-файл.

Рекомендуется не использовать связанные файлы, а импортировать их в ядро и сохранять как неотъемлемую часть фильма.

### Playback

Настройки управления воспроизведением Flash-фильмов в Director:

- флажок Image определяет, будут ли отображаться кадры фильма при его воспроизведении. Кадры Flash-фильма будут отображаться, только если этот флажок активизирован. При работе со звуковыми SWF-файлами или файлами, содержащими лишь сценарий, отключите этот флажок;

панель, называется *спрайтом*. Данная панель напоминает панель Timeline во Flash, содержащую слои и кадры, поскольку имеет горизонтальные ряды для спрайтов, называемых каналами, и вертикальные колонки для кадров, при этом ряды спрайт-каналов пронумерованы (рис. 23.1). Совершая манипуляции с элементами в среде Director, вы можете ссылаться на номера спрайтов, например `sprite(1)` или `sprite(15)`. Более подробная информация об этом находится в файлах справки среды Director (Help ▶ Director Help).

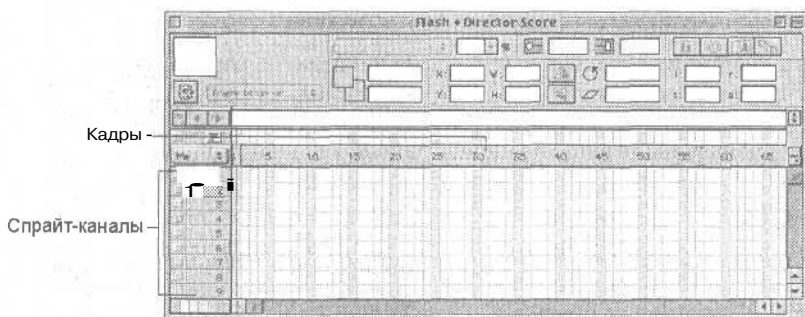
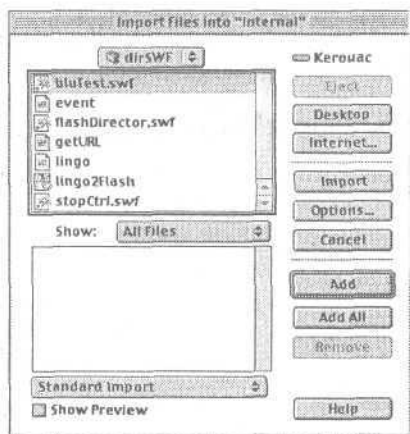


Рис. 23.1. Панель Score среды Director

Для того чтобы можно было начать работу с Flash-фильмом в среде Director, спрайт Flash-фильма должен находиться на панелях Score и Stage (аналог области действия). Для того чтобы Flash-фильм стал спрайтом, он должен быть компонентом ядра, а для того, чтобы он стал компонентом ядра, его нужно импортировать в Director-фильм. Такая процедура на первый взгляд может показаться: очень трудоемкой, но дальше мы продемонстрируем, что сделать это не сложно. Для импорта Flash-фильма в Director-фильм необходимо произвести следующие действия.

1. Выберите в среде Director команду File ▶ Import или воспользуйтесь комбинацией клавиш `Cmd/Ctrl+R`, после чего появится диалоговое окно Import Files; (Импортировать файлы). Оно похоже на аналогичное окно среды Flash.
2. В верхней области данного окна найдите нужные файлы. Для поиска SWF-файлов, которые нужно импортировать, вы можете воспользоваться раскрывающимся списком Show.
3. Когда нужный файл найден, выберите его и нажмите кнопку Add, после чего он будет перенесен в нижнюю область окна Import Files, где помещаются все выбранные файлы. Вы можете импортировать сразу несколько файлов, продолжая использовать для этого верхнюю область окна Import Files.
4. Закончив выбор файлов, нажмите кнопку Import, после чего все эти файлы будут импортированы в среду Director.



## ПРИМЕЧАНИЕ

Фильмы, созданные в среде Flash MX, всегда можно сохранить в формате более ранней версии, если у вас отсутствует необходимая версия Director. Об этом рассказывается в главе 27. Для более детальной информации, касающейся совместимости Flash и Director, обратитесь на Web-узел [www.macromedia.com/support/director](http://www.macromedia.com/support/director).

При помощи Lingo, языка среды Director, вы сможете манипулировать многими параметрами воспроизведения Flash-фильма, содержащегося в Director-фильме. Существует несколько удобств при использовании среды Director.

**Гибкость графики.** Director в основном работает с растровой графикой. А это значит, что если вы, например, измените масштаб изображения, его качество ухудшится. При работе же с векторной графикой ситуация иная — в этом случае нет потери качества. Как программа для создания анимации Flash, конечно, превосходит Director, но, тем не менее, в Director предусмотрены все необходимые инструменты для работы с векторной графикой, хотя их и меньше, чем во Flash.

**Расширенное управление анимационными элементами.** Во Flash клип является неотъемлемой частью фильма и воспроизводится автономно в своей временной шкале. Если, используя Flash, создать SWF-фильм, каждая часть которого является клипом, то затем в среде Director можно с помощью языка Lingo управлять каждым клипом отдельно, что создает определенные удобства. Подробнее об этом рассказано далее в разделе «Манипулирование спрайтами Flash при помощи Lingo».

**Наилучшие макроязыки.** Среда Flash содержит язык ActionScript, а среда Director — Lingo. Если вы знаете один из них или язык JavaScript, то сможете без проблем изучить другой. Но учтите, что ActionScript больше напоминает JavaScript, чем Lingo, хотя синтаксис всех трех языков похож. Язык ActionScript можно использовать для управления Flash-элементами в фильмах Director, и наоборот, с помощью Lingo управлять Director-элементами, которые входят во Flash-фильмы. Каждый из этих языков имеет свои преимущества, поэтому при разработке приложения вы должны выбрать тот, который лучше поможет решить поставленные задачи.

Достоинство среды Director состоит в том, что с ее помощью можно разрабатывать как простые приложения, например, галереи SWF-файлов, так и сложные, в которых пользователь сможет создавать собственных персонажей и взаимодействовать с персонажами других пользователей.

## Импорт Flash-фильмов

Для того чтобы внедрить Flash-фильм в Director-проект, сначала нужно поместить Flash-файл в среду Director. Этот процесс, так же как во Flash, называется импортом. После импортирования файла в среду Director он будет сохранен в ядре и готов к использованию.

Наверное, термин *ядро* вы встречали ранее. Любой элемент, который можно использовать в фильме, сохраняется в ядре и называется *компонентом ядра* (как в библиотеке Flash). Если это понятно, то перейдем к *области действия*.

Панель, с помощью которой можно управлять расположением компонентов ядра на области действия, называется Score. Компонент ядра, помещенный на эту

## Глава 23

# Совместная работа Flash и Director

Совместно с Flash может работать большое количество программ, одна из которых — Macromedia Director.

Если Flash, несомненно, является прекрасным средством быстрой разработки Интернет-приложений, то Macromedia Director великолепно подходит для публикации ваших фильмов на компакт-диски. Хотя эта программа довольно громоздкая и требует значительных ресурсов системы, она позволяет работать с двухмерными растровыми изображениями, двухмерной векторной графикой, цифровым видео, звуком, трехмерной графикой (появилась в версии 8.5) и наиболее важными SWF-файлами. В этой главе вы изучите совместную работу Flash и среды Director и научитесь создавать интерактивные фильмы, обладающие наилучшими характеристиками обоих сред разработки. Здесь будут рассмотрены следующие темы:

- ▶ вставка Flash-фильмов в Director;
- ▶ характеристики Flash в среде Director;
- ▶ функционирование Flash-фильмов в качестве спрайтов;
- ▶ комбинирование языков ActionScript и Lingo.

## Работа с Flash в среде Director

В среде Director Flash-фильмы можно просматривать с помощью модуля Flash Asset Xtra. Director 8.5 уже содержит этот модуль, который поддерживает функциональные возможности Flash 5. К моменту выхода этой книги Macromedia не представила новую версию модуля для Flash MX и проигрывателя Flash 6, поэтому все фильмы, которые вы будете запускать в среде Director, имеют формат Flash 5. Более ранние версии Director могут поддерживать Flash только с некоторыми ограничениями. В табл. 23.1 показаны соответствия между версиями Flash и Director.

**Таблица 23.1.** Совместимость Director и Flash

| Версия Flash         | Версия Director                     |
|----------------------|-------------------------------------|
| 6 (Flash MX) и ранее | Отсутствует, на момент выхода книги |
| 5 и ранее            | 8.5                                 |
| 4 и ранее            | 8.0, 7.02                           |
| 3 и ранее            | 7.0                                 |
| 2                    | 6.5, 6.02                           |

# ЧАСТЬ 7

---

## Использование Flash с другими программами

*Практически ни одну сложную интересную программу нельзя создать, используя только одно мультимедийное приложение. Хотя Flash является очень мощной средой разработки, бывают ситуации, когда эта технология, несмотря на гибкость и скорость векторной графики и анимации, а также интерактивные возможности, предоставляемые ActionScript, не в состоянии обеспечить всей функциональности, необходимой вашему приложению. В таких случаях совместно с Flash используют другие программы. О некоторых из них мы расскажем в данной части книги.*

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| <b>Глава 23.</b> Совместная работа Flash и Director.....                  | 464 |
| <b>Глава 24.</b> Совместная работа Flash и FreeHand.....                  | 481 |
| <b>Глава 25.</b> Использование звуковых редакторов совместно с Flash..... | 490 |
| <b>Глава 26.</b> Создание трехмерных объектов в среде Flash.....          | 510 |

задать для него низкий уровень качества. Для звуков, которые имеют тип синхронизации Stream, укажите следующие параметры: MP3, 8Kbps и Mono.

Чтобы уменьшить размер результирующего файла, нужно убрать повторения звукового файла Sync Track в звуковом сопровождении. При использовании звуков DoReMedia во Flash необходимо только одно повторение звукового файла Sync Track для обеспечения синхронизации. Так как повторения потоковых звуков увеличивают размер фильма, не следует применять их чаще, чем это требуется. Для сохранения синхронизации в части В вы должны связать еще один экземпляр звука Sync Track (SYNC\_100.aif) с ключевым кадром 481. Выделите кадр, активизируйте команду Insert ▶ Keyframe, а затем выберите файл SYNC\_100.aif в панели Properties. Установите тип синхронизации Stream.

#### СОВЕТ

В некоторых случаях звуковой файл Sync Track не позволяет синхронизировать частоту смены кадров фильма, если он начинается в первом кадре. Чтобы разрешить эту проблему, передвиньте начало воспроизведения звука Sync Track в кадр номер 10.

## Заключительные размышления о звуковом сопровождении

Вы только что завершили создание интерактивного звукового сопровождения фильма. Примите наши поздравления! Все, что мы рассмотрели в этом практикуме, поможет вам в дальнейшей работе со звуками DoReMedia, которые лучше всего использовать в интерактивных фильмах. Для закрепления изученного материала самостоятельно выполните следующие упражнения.

- ▶ Поэкспериментируйте с параметрами панорамирования для различных участков, слоев и звуковых эффектов. В демонстрационном файле whiskey100.swf есть несколько звуков гитар, которые переходят из одного канала на другой. Кроме того, стереоэффекта можно достичь, изменяя уровень громкости звука в разных каналах (используйте окно Edit Envelope).
- ▶ Попробуйте создать третью часть звукового сопровождения. Для этого у вас есть все необходимые элементы.
- ▶ Поэкспериментируйте со звуковыми эффектами и кнопками. Можно создать интерактивное звуковое сопровождение, в котором партии разных музыкальных инструментов «подключаются» при нажатии на различные кнопки.
- ▶ Для использования звуков DoReMedia нужна лицензия. Компания имеет большую звуковую библиотеку, содержащую музыку различных стилей. Посетите Web-страницу этой компании ([www.doremia.com](http://www.doremia.com)) для получения дополнительной информации о звуках и способах их применения.



Если вам понравилось работать с звуками *Whiskey*, можете поработать с музыкой в стиле «техно» *Passive Aggression*. На прилагаемом компакт-диске зайдите в каталог Hands On 6\DoReMedia. Здесь вы найдете два каталога, которые названы *Passive Aggression*. В одном из них находятся файлы AIF, а в другом — WAV. Создайте новое звуковое сопровождение, используя эти звуки и рассмотренную методику.



```

 _root.soundtrack.gotoAndPlay(481);
 gotoAndStop(2);
}
}

```

Часть А расположена в кадрах до 481 временной шкалы клипа звукового сопровождения. Если условие ее воспроизведения (первая строка) выполняется, клип со звуковым сопровождением начинает проигрываться с кадра номер 481 (вторая строка кода программы), то есть начинает воспроизводиться часть В звукового сопровождения. При этом: в основной временной шкале головка воспроизведения переводится в кадр номер 2 (третья строка). Если же при нажатии кнопки уже воспроизводится часть В, то воспроизведение повторно не активизируется.

- Теперь следует задать программу для воспроизведения части А звукового сопровождения фильма. Выделите кнопку, предназначенную для воспроизведения части А, перейдите к кадру номер 2 и откройте панель Actions. Введите следующие команды:

```

on(press){
 if(_root.soundtrack._currentframe>481){
 _root.soundtrack.gotoAndPlay(25);
 gotoAndStop(1);
 }
}

```

Эти команды аналогичны командам для кнопки воспроизведения части В. Если проигрывается часть В (кадры больше 481), тогда при нажатии кнопки воспроизводящая головка переходит к кадру номер 25, с которого начинается часть А, а на основной временной шкале — к кадру номер 1.

Выберите команду Control ▶ Test Movie, чтобы проверить работу кнопок.

- Для улучшения интерактивности интерфейса фильма зададим звуки, воспроизводимые при нажатии кнопок. Когда зритель нажимает кнопку, он не только проигрывает соответствующую часть звукового сопровождения, но и слышит звук, который помогает заполнить паузу.

Перейдите в режим редактирования символа для одной из ваших кнопок. Добавьте новый слой и вставьте ключевой кадр для события Down. Свяжите с этим кадром плавный звук гитарной струны и установите тип синхронизации Event. В примере whiskey100.swf в качестве звуков для кнопок используются файлы whiskey\_lead\_01.aif и whiskey\_lead\_02.aif. Эти звуки были выбраны, потому что они не подчинены ритму и медленно стихают, обеспечивая тем самым идеальный переход.

- После окончания работы с кнопками проверьте полученные результаты командой Control ▶ Test Movie. Создание вашей интерактивной звукозаписи завершено.

Перед публикацией фильма неплохо было бы проверить параметры звука в окне Publish Settings, которое можно открыть командой File ▶ Publish Settings. Для звуков типа Event, выбирайте параметры в зависимости от носителя, на котором публикуется фильм, скорости соединения компьютеров ваших зрителей с Интернетом и ряда других факторов. Вспомните, что вы снизили до нуля уровень громкости звукового файл Sync Track. Поскольку этого звука не будет слышно, вы можете

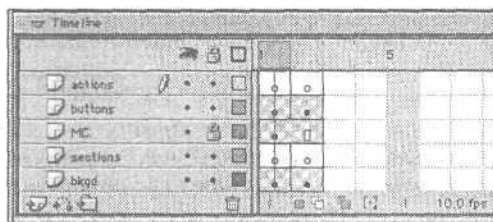
| Слой      | Номер кадра | Звук              |
|-----------|-------------|-------------------|
| A section | 481         | Whiskey_A.aif     |
| Drum      | 481         | Whiskey_drums.aif |

На этом мы завершаем создание звукового сопровождения фильма. Теперь у нас есть две независимые части музыки, для воспроизведения которых необходимо разработать соответствующий интерфейс.

## Разработка интерфейса

Для завершения этого практического задания следует добавить в библиотеку несколько элементов. Нам понадобится как минимум одна кнопка. Если вы открывали демонстрационный файл `whiskey100.swf`, то видели две кнопки, предназначенные для запуска различных частей звукового сопровождения. При желании вы можете использовать как одинаковые по виду кнопки, так и различные. Вы даже можете вместо кнопки поместить в фильм какой-нибудь графический элемент и применять два его состояния для выбора части.

1. Вставьте два новых слоя в главной временной шкале. Один из них назовите `actions`, а другой — `buttons`. Слой для звукового сопровождения назовите `MC`. Выделите его и на панели свойств в поле слева от кнопки `Swar` введите имя `soundtrack`.
2. Перетащите две кнопки в область действия и удостоверьтесь, что они находятся в слое `buttons`. Их позицию выберите по вашему усмотрению. Одна кнопка будет служить для воспроизведения части А, а другая — части В.
3. Вставьте новый ключевой кадр в слоях `buttons` и `actions`. Затем добавьте кадр (нажав клавишу `F5`) в слое `MC` и поместите в него клип со звуковым сопровождением. После этого основная временная шкала должна быть похожа на временную шкалу, показанную на рис. П6.5.



**Рис. П6.5.** В основной временной шкале имеются два кадра, каждый соответствует своей части звукового сопровождения

4. Выберите первый ключевой кадр в слое `actions` и откройте панель `Actions`. Введите команду `stop()`. Сделайте то же самое в ключевом кадре 2. Это позволит ограничить воспроизведение одним кадром.
5. Выделите кнопку, предназначенную для воспроизведения части В, и для первого кадра введите на панели `Actions` следующие команды:

```
on(press){
 if(_root.soundtrack._currentframe<481){
```

при повторении. Вставьте новые ключевые кадры в кадр 865 в двух слоях: `fills` и `loop riffs`. Свяжите звук `Whiskey_lead_03.aif` с ключевым кадром в слое `loop riffs`, а звук `Whiskey_crash.aif` — с ключевым кадром в слое `fills`. На этом создание части В звукового сопровождения завершается.

#### ПРИМЕЧАНИЕ

Вы, наверное, заметили, что команда возврата к началу мелодии `gotoAndPlay(481)` и звуки для обеспечения главного перехода находятся в разных кадрах. Мы сделали это специально. Поскольку команда повторения выполняется несколько позже начала воспроизведения переходных звуков, они короткое время будут звучать независимо друг от друга. Это придает переходу большую выразительность. Мы выбрали кадр номер 871, потому что решили, что это даст лучший результат. Но вы можете работать и с другим временным интервалом.

16. При воспроизведении часть В звукового сопровождения не слышна, потому что команда в кадре номер 457 возвращает головку воспроизведения в начало части А. Для проверки всего фильма нужно перейти к кадру номер 457, открыть панель `Actions` и ввести перед командой два символа косой черты (`//`), для того чтобы эта команда стала комментарием.

Выберите команду `Control ▶ Test Movie` и прослушайте обе части звукового сопровождения — А и В. Конечно, повторяться будет только часть В. После прослушивания части В необходимо убрать символы комментария (`//`) из оператора в кадре номер 457.

17. Создание звукового сопровождения идет по плану. Вы имеете две части песни для двух частей Flash-фильма. Теперь нужно создать интерфейс управления звуковым сопровождением. Заранее неизвестно, сколько времени зритель будет смотреть каждую часть фильма. И поэтому мы создали звуковое сопровождение, в котором каждая часть будет повторяться, что позволит слушать ее в течение разных интервалов времени.

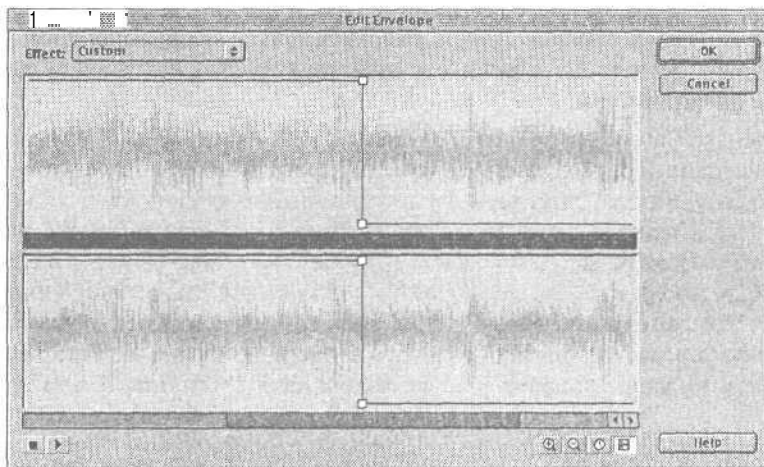
Вспомните, что мы везде использовали тип синхронизации `Event`. В этом случае звук воспроизводится до тех пор, пока не будет остановлен или пока не закончатся повторения. Таким образом, если головка воспроизведения будет прыгать между частями А и В звукового сопровождения, вы услышите диссонирующие звуки. Чтобы избежать этого и обеспечить свободный переход от одной части звукового сопровождения к другой, необходимо останавливать воспроизведение всех звуков одной части при переходе к другой. Для этого свяжите каждый звук, который нужно остановить, с новым ключевым кадром и выберите для него тип синхронизации `Stop`. В табл. Пб.1 перечислены имена слоев, звуки и номера кадров, в которых следует организовать остановку воспроизведения звука.

**Таблица Пб.1.** Звуки и кадры, в которых следует остановить их воспроизведение

| Слой       | Номер кадра | Звук              |
|------------|-------------|-------------------|
| Breakdown  | 25          | Whiskey_break.aif |
| B2 section | 25          | Whiskey_B_02.aif  |
| B section  | 25          | Whiskey_B.aif     |
| A2 section | 481         | Whiskey_A_02.aif  |

13. Продолжаем создание части В. Вставьте новый ключевой кадр в кадр 673 слоя В section. Свяжите с ним звук `Whiskey_B.aif` и задайте тип синхронизации Event, а также — количество повторений, равное 2. Затем для улучшения звучания свяжите звук `Whiskey_B_02.aif` с ключевым кадром 817 слоя В2 section. Задайте для этого звука тип синхронизации Event.
14. Обратите внимание, что звуки в слоях section В и section В2 перекрываются и воспроизводятся не очень удачно. В этом практическом задании файл `Whiskey_B_02.aif` занимает 2 интервала синхронизирующего звука `Sync Track`, оканчивая музыкальную фазу. Вся музыка части В имеет структуру в восемь тактов. Поэтому вы должны уменьшить уровень громкости звука файла `Whiskey_B.aif` до 0 на двух последних тактах (начиная с 817 кадра). В результате на протяжении шести тактов будет слышен звук `Whiskey_B.aif`, а во время двух следующих — звук `Whiskey_B_02.aif` — всего восемь.

Выделите ключевой кадр в позиции 673 слоя В section, где начинается файл `Whiskey_B.aif`. Щелкните на кнопке Edit на панели Properties для открытия окна Edit Envelope. Затем щелкните на кнопке Frames и прокрутите содержимое окна до кадра номер 817. Выберите элемент Custom из списка Effect и измените кривую громкости так, чтобы звук исчез полностью в кадре номер 817.



**Рис. П6.4.** Окно Edit Envelope с измененной кривой громкости

После окончания формирования уровня громкости в слое В section щелкните на кнопке ОК. Теперь, когда часть В звукового сопровождения создана, зададим повторы ее воспроизведения.

15. Для повторения части В применяется тот же прием, что и для повторения звуков части А. Вставьте новый ключевой кадр в кадр 871 слоя actions. Откройте панель Actions и введите следующую команду:

```
gotoAndPlay(481);
```

Эта команда перемещает головку воспроизведения к кадру, где начинается часть В. Теперь нужно задать два звука, которые обеспечат плавное звучание

9. Вы можете задать повторение звука, возвращая головку воспроизведения в кадр, где начиналось воспроизведение этой части звукового сопровождения. Но при этом Flash не сможет поддерживать нужный темп. В таком случае следует использовать файлы со звуковыми эффектами DoReMedia.

Перейдите к кадру номер 457 слоя `actions` и вставьте новый ключевой кадр. Откройте панель `Actions` и введите следующую команду:

```
GotoAndPlay(25);
```

Эта команда перемещает головку воспроизведения в кадр номер 25, где начинается часть А звукового сопровождения. Теперь необходимо сделать плавный переход, используя короткий звук. Вставьте еще один ключевой кадр в кадр 457, на этот раз в слое `loop riffs`. Свяжите с ним звук `Whiskey_lead_01.aif` и задайте тип синхронизации `Event`.

10. Активизируйте команду `Control ▶ Test Movie` для тестирования созданной части звукового сопровождения. После того как закончится воспроизведение части А, вы услышите звук гитары и вступление ударных — неплохой переход, не правда ли?
11. Приступим к другой половине звукозаписи или части В. Вставьте новый ключевой кадр в кадр 481 слоя `breakdown`. Свяжите с ним звук `Whiskey_break.aif`, задайте тип синхронизации `Event` и введите количество повторений, равное 2. Теперь добавим звук тарелок. Вставьте новый ключевой кадр в позицию 481 слоя `fills`. Свяжите с ним звук `Whiskey_crash.aif` и задайте тип синхронизации `Event`.
12. Звук `Whiskey_break.aif` — это дуэт бас-гитары и барабана. В нашем звуковом сопровождении он повторяется дважды. Для того чтобы подчеркнуть его звучание, добавим еще один звук. Вставьте новый ключевой кадр в кадр 577 слоя `loop riffs`. Свяжите с ним звук `Whiskey_lead_01.aif` и задайте тип синхронизации `Event`. В этом же слое свяжите звук `Whiskey_lead_03.aif` с ключевым кадром 625, а звук `Whiskey_lead_02.aif` — с ключевым кадром 649. Обратите внимание, что данные ключевые кадры согласованы с пиками звука `Sync Track`. Эти пики очень удобно использовать в качестве меток. Если хотите, можете добавить звучание тарелок в те же кадры слоя `fills` (рис. П6.3).

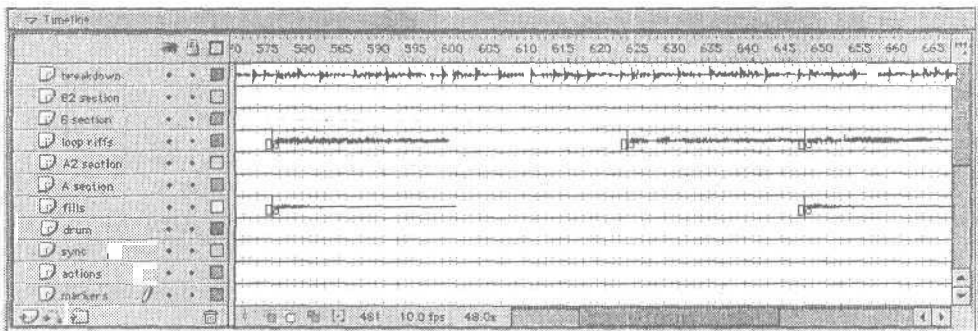


Рис. П6.3. Вторая часть звукового сопровождения фильма со звуками тарелок

7. Вставьте новый ключевой кадр в кадр 73 слоя A section. Свяжите с ним файл `Whiskey_A.aif`, задайте тип синхронизации Event и введите количество повторений, равное 2. Прослушав данный звук, вы поймете, что это начало песни. Можно добавить звучание ударных инструментов, которое обеспечит переход к звучанию гитары. Вставьте ключевой кадр в кадр 67 слоя `fills`. Свяжите с ним звук `Whiskey_fill_01.aif` и задайте тип синхронизации Event. Теперь воспроизведите звук. Вы услышите барабанное вступление, но когда начнут звучать основные ударные инструменты, станет заметным некоторое несоответствие между звуками, которые они воспроизводят. Такое несоответствие легко исправить. На графиках амплитуды звуков ударных видно, что в 68 кадре слоя `drum` есть промежуток, а в слое `fills` начинается его первый пик. Необходимо отрегулировать уровень громкости звука в слое `drum` так, чтобы звук в этот момент стихал, давая возможность ясно услышать воспроизведение звука в слое `fills`. Выделите ключевой кадр в позиции 25 слоя `drum`, там, где вы внедрили файл `Whiskey_drums.aif`. Щелкните на кнопке Edit на панели свойств, чтобы открыть окно Edit Envelope.

Щелкните на кнопке Frames, которая находится слева от кнопки Help в окне Edit Envelope, и вы увидите графики амплитуды звука относительно кадров. Выберите элемент Custom из списка Effect и добавьте на графике уровня громкости две точки, перетаскив их таким образом, чтобы убрать звук на 69 кадре (рис. П6.2). После чего щелкните на кнопке ОК.

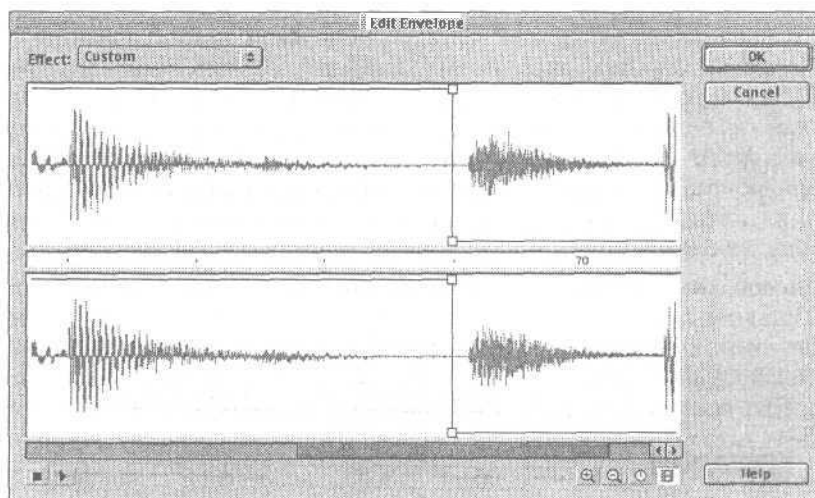


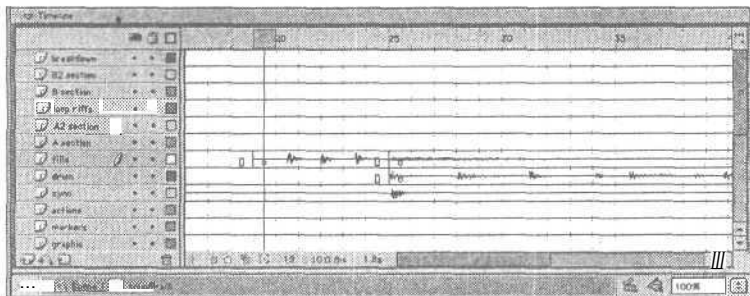
Рис. П6.2. Окно Edit Envelope с измененным уровнем громкости звука в слое `drum`

8. Продолжим создание части А звукового сопровождения фильма. Вставьте новый ключевой кадр в позицию 265 слоя A2 section. Свяжите с ним звук `Whiskey_A_02.aif`, задайте тип синхронизации Event и количество повторений, равное 2. Этот последний звук завершает часть А звукового сопровождения фильма. Она длится 45 секунд. Так как зритель может смотреть фильм дольше этого времени, данную часть нужно сделать повторяющейся.

Звуковой файл Sync Track, уровень громкости которого мы понизили до нуля, является ключевым элементом набора звуков DoReMedia. Это звуковой файл низкого качества, который оптимизирован для согласования темпа музыки. При его применении используется синхронизация типа Stream, чтобы согласовать воспроизведение звука с частотой кадров. Ранее вы задали эту частоту, равной 10 кадрам в секунду. Музыка, звучащая в фильме, имеет темп 100 тактов в минуту. На Web-узле DoReMedia представлен калькулятор частоты кадров. С его помощью вы сможете вычислить наиболее подходящую частоту кадров фильма, в котором используется музыка DoReMedia с определенным темпом.

Звуковой файл Sync Track имеет также другое назначение. График его амплитуды на временной шкале содержит большой пик. Использование этого пика в качестве визуальной ссылки облегчит синхронизацию темпа звучания музыки. Подобные пики могут служить своеобразной временной шкалой: задав количество повторений этого звука, равное 50, вы формируете как бы временную шкалу, имеющую 50 делений. Первый такт каждого звука музыкального сопровождения фильма должен начинаться синхронно с этими пиками.

5. Поместив на временную шкалу звуковой файл Sync Track, следует заняться остальными музыкальными компонентами звукового сопровождения. Выделите 1200-е кадры во всех слоях, кроме слоя sync, и нажмите клавишу F5. Этой операцией вы создадите более чем достаточное количество кадров, необходимых для звукового сопровождения фильма. Теперь обратите внимание на второй пик, находящийся в 25-ом кадре слоя sync. Он указывает на начало нового временного интервала. Вставьте ключевой кадр в позицию 25 слоя drum и свяжите с ним звук `Whiskey_drums.aif`. Задайте тип синхронизации Event. Для всех остальных звуков будем использовать этот тип синхронизации, так как если звук типа Event воспроизводится хотя бы один раз, он остается в памяти и его можно воспроизводить повторно без дополнительного увеличения размера фильма. Активизируйте команду Control ▶ Play для прослушивания нового звука. Чтобы остановить воспроизведение, активизируйте команду Control ▶ Stop.
6. Теперь добавим к музыкальному сопровождению звуки ударных инструментов. Вставьте ключевой кадр в позицию 25 слоя fills. Свяжите с ним звук `Whiskey_crash.aif` и задайте тип синхронизации Event. Затем вставьте еще один ключевой кадр в позицию 19 и свяжите с ним файл `Whiskey_fill_01.aif` типа Event. При воспроизведении вы услышите звук ударных инструментов.



3. Создайте 11 новых слоев во временной шкале (всего 12). Присвойте им, следуя снизу вверх, следующие имена: graphic (был создан на втором шаге), markers, actions, sync, drum, fills, A section, A2 section, loop riffs, B section, Ei2 section и breakdown (рис. П6.1).

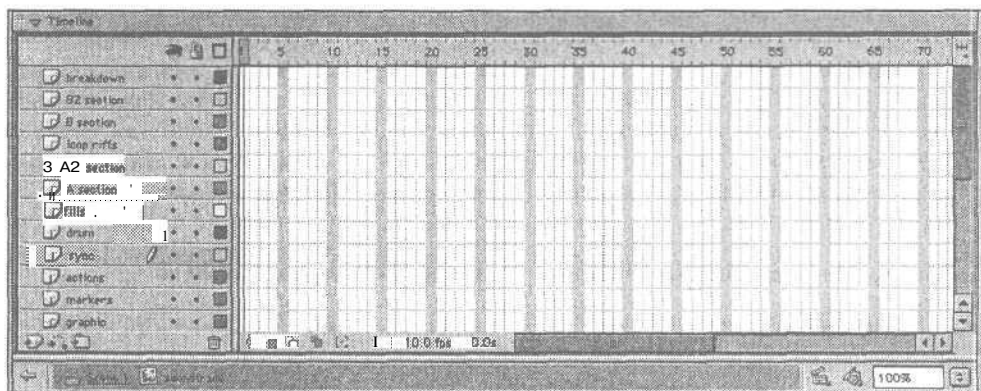
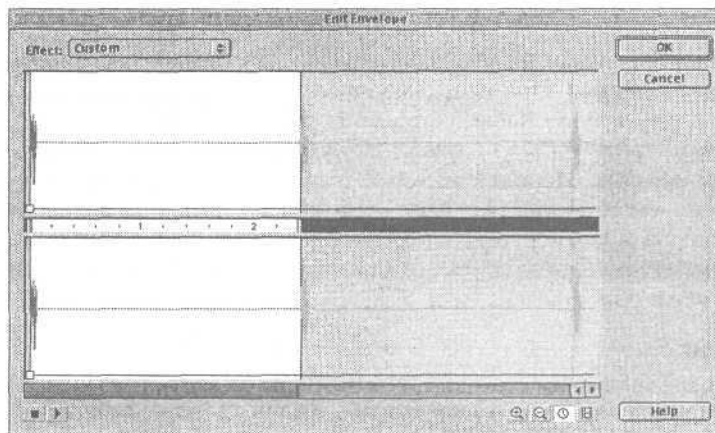


Рис. П6.1. Слои во временной шкале клипа soundtrack

4. Вставьте ключевой кадр в кадр 1 слоя sync. Привяжите звук SYNC\_100.aif к этому ключевому кадру, введите количество повторений 50 и задайте Stream-синхронизацию. Перейдите к кадру под номером 1200. (Если Flash отображает только 500 кадров во временной шкале, перейдите к кадру с максимальным номером, а затем нажмите клавишу F5. Flash осуществит прокрутку временной шкалы на следующие 500 кадров. Вновь выберите кадр с максимальным номером и опять нажмите клавишу F5. Продолжайте выполнять эти действия до тех пор, пока на шкале не появится 1200-й кадр.)

Далее откройте окно Edit Envelope, щелкнув на кнопке Edit на панели Properties. Из списка Effect выберите элемент Custom и перетяните метки кривых уровня громкости каждого канала вниз. Это делается для того, чтобы данный звук не был слышен при воспроизведении. Затем нажмите кнопку OK.






интегрированные аудиоэлементы фильма, хорошо сочетаясь при этом с графикой и средствами навигации.

Звуковые средства DoReMedia разделены на три категории: слои, фрагменты и звуковые эффекты. Основными элементами являются *слои*; например, ими могут быть отдельные музыкальные фразы (скажем, басовая партия или сопровождение сольной партии на ритм-гитаре). *Фрагменты* — это отдельные файлы, которые содержат фрагменты мелодии. Несколько слоев можно объединить для создания нового фрагмента или использовать их по отдельности. *Звуковые эффекты* — это звуковые файлы, которые можно запускать отдельно для улучшения звучания или для создания переходов. Далее рассказывается, как комбинировать данные элементы, создавая звуковое сопровождение фильма.

## Создание звукового сопровождения

 Прежде всего вы должны импортировать в фильм несколько файлов. Откройте каталог DoReMedia, которая находится в каталоге Hands On 6 на компакт-диске, и импортируйте оттуда в библиотеку фильма следующие файлы:

|                   |                     |
|-------------------|---------------------|
| SYNC_100.aif      | Whiskey_crash.aif   |
| Whiskey_A.aif     | Whiskey_drums.aif   |
| Whiskey_A_02.aif  | Whiskey_fill_01.aif |
| Whiskey_B.aif     | Whiskey_lead_01.aif |
| Whiskey_B_02.aif  | Whiskey_lead_02.aif |
| Whiskey_break.aif | Whiskey_lead_03.aif |

После этого вы можете приступить к работе над звуковым оформлением фильма с интерактивным взаимодействием. Использование аудиоэлементов в среде Flash подробно рассматривалось в главе 20.

1. Задайте частоту кадров фильма. Среди звуковых файлов DoReMedia есть специальные звуковые файлы типа Sync Track, которые служат для согласования темпов воспроизведения звуков. Об этом мы поговорим несколько позже, а пока выберите команду **Modify** ▶ **Document** (Модификация ▶ Документ). В поле **Frame Rate** (Частота кадров) открывшегося окна **Document Properties** (Параметры документа) введите частоту кадров вашего фильма, равную 10 кадрам в секунду, и подтвердите эту установку щелчком на кнопке **OK**.
2. Активизируйте команду **Insert** ▶ **New Symbol** (Вставка ▶ Новый символ) и создайте новый символ типа **clip** с именем **soundtrack**. Для этого следует ввести указанное имя в поле **Name**, установить переключатель **Movie Clip**, после чего щелкнуть на кнопке **OK**. В данный клип будет помещено все звуковое сопровождение фильма. Используя **ActionScript**, вы сможете организовать управление воспроизведением музыки с помощью интерактивных графических элементов. Задайте первому слою имя **graphic** и создайте простой графический элемент для визуального представления этого клипа. Перетащите экземпляр клипа в область действия.

### ПРИМЕЧАНИЕ

Теперь все изменения, касающиеся слоев, кадров и других элементов, будут произведены во временной шкале клипа. Поэтому убедитесь, что вы работаете с клипом в режиме редактирования символов.

Одним из преимуществ автономных файлов является возможность создания фильмов, способных работать на всех платформах. Независимо от операционной системы, используемой при создании Flash-фильмов, вы сможете создать конечные файлы, которые будут соответствовать возможностям пользовательского оборудования. Это делает Flash одним из наиболее переносимых среди современных мультимедийных приложений.

## Приложение Flash Player

Приложение Flash Player поставляется вместе с программой Flash. Его можно использовать для воспроизведения SWF-файлов без помощи Web-браузера. Окно Flash Player (рис. 29.1) содержит несколько команд меню, с помощью которых можно управлять воспроизведением и внешним видом фильма. Файл приложения Flash Player хранится в каталоге Players, который находится внутри каталога с файлами приложения Flash на жестком диске. На компьютерах Macintosh файл проигрывателя имеет имя SAFlashPlayer, а на компьютерах Windows он называется SAFlashPlayer.exe.

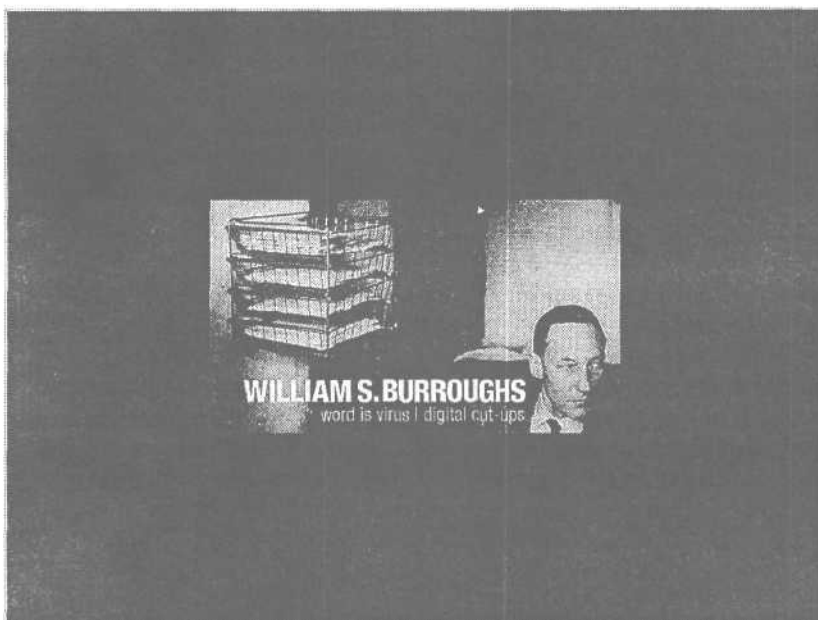


Рис. 29.1. Фильм, воспроизводимый с помощью приложения Flash Player

Фильмы в формате SWF можно распространять в виде небольших файлов, способных работать на различных платформах. Но для воспроизведения таких фильмов требуется наличие на компьютере пользователя установленного приложения Flash Player либо подключаемого модуля Flash. Поскольку большинство пользователей не имеют данного приложения, SWF-формат не является оптимальным для широкого распространения фильмов. Для полной уверенности, что созданный вами фильм смогут просмотреть на любом компьютере, используйте формат файла projector.

## Flash-фильм в формате projector

Автономные файлы формата Flash projector являются, возможно, самыми оптимальными для распространения за пределами Web. Файл в формате projector представляет собой исполняемый файл, способный воспроизвести Flash-фильм как в среде Windows, так и в среде Macintosh. Пользователи компьютеров с операционными системами Windows и Macintosh смогут увидеть ваш фильм без помощи дополнительного программного обеспечения или подключаемых модулей. Кроме того, поместив такой файл на компакт-диск, вы сможете воспроизвести фильм на любом компьютере, имеющем дисковод лазерных дисков. На рис. 29.2 показан пример воспроизведения файла формата projector.



**Рис. 29.2.** Фильм в формате projector может воспроизводиться в своем собственном окне либо в полноэкранном режиме

## Создание файлов в формате projector

Создание или, точнее, публикация фильмов в виде файлов projector — процесс достаточно простой. Кроме того, Flash позволяет одновременно создавать файлы данного типа как для Windows, так и для Macintosh. Одним из недостатков такого файла может считаться его размер, который будет на несколько сотен килобайтов больше аналогичного файла формата SWF. Это естественно, поскольку файл типа projector должен содержать информацию, необходимую для воспроизведения фильма с использованием ресурсов операционной системы.

Для создания файла формата projector выполните следующие действия.

1. Завершив создание фильма, выберите команду Save меню File.
2. Откройте диалоговое окно Publish Settings (Параметры публикации), выбрав команду Publish Settings меню File (рис. 29.3).
3. Установите флажок Windows projector или Macintosh projector в зависимости от того, для какой операционной системы предназначен фильм. Можно устанавливать одновременно оба флажка — в этом случае будет создано два файла. В полях Filename введите имена для создаваемых файлов. В поле имени файла Windows projector убедитесь в наличии расширения .exe.
4. Для завершения процесса публикации выполните одно из следующих действий.
  - ▷ Щелкните на кнопке Publish (Опубликовать). Ваш фильм будет тотчас опубликован. Эта кнопка используется в тех случаях, когда необходимо опубликовать несколько различных типов фильмов с одними параметрами.

- Щелкните на кнопке ОК для подтверждения всех параметров, установленных в диалоговом окне Publish Settings. Затем выберите команду Publish меню File или воспользуйтесь комбинацией клавиш Shift+F12.

В открывшемся окне Publish находится индикатор хода процесса публикации фильма. После завершения созданные файлы будут помещены в тот же каталог, где находится Flash-файл, с которым вы работаете в настоящий момент.

5. Завершите работу с приложением Flash, выбрав команду Exit меню File. Чтобы воспроизвести созданный файл типа projector, выполните двойной щелчок мышью на его значке. В таком виде файл готов для переноса на компакт-диск.

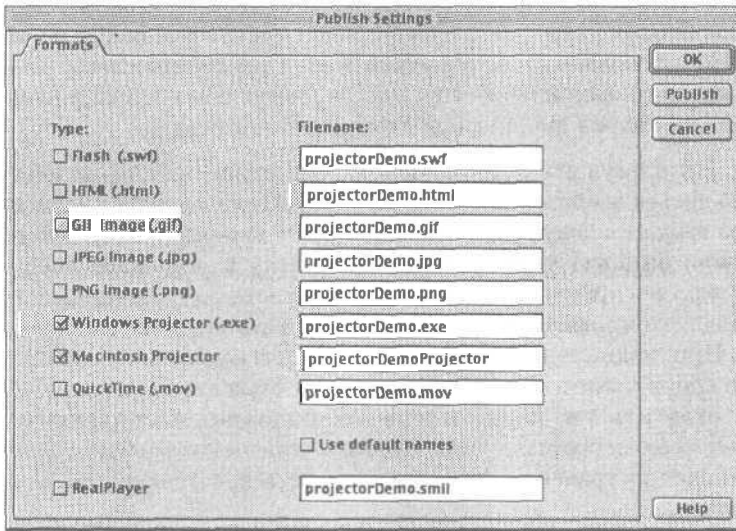


Рис. 29.3. Диалоговое окно Publish Settings позволяет установить параметры создания автономных файлов типа projector

## Интерфейс управления фильмом

Публикация автономного файла в формате projector является конечным этапом процесса, но существуют еще и дополнительные возможности, которые помогут вашему фильму приобрести вид независимого приложения. С помощью нескольких команд ActionScript можно превратить фильм Flash в независимый мультимедийный продукт с интерфейсом управления. Такая доработка не только сделает ваш продукт более профессиональным, но и даст пользователям возможность управлять режимом воспроизведения фильма.

### Действие fscommand

Действие fscommand является элементом языка ActionScript. С его помощью происходит обмен данными между программами Flash, Flash Player и приложением, которое исполняет роль хоста вашего фильма. В данном контексте под таким приложением подразумевается операционная система Windows или

Macintosh. Используя действие `fscommand()`, можно установить различные параметры представления фильма, а также расширить возможности управления воспроизведением фильма. Ниже перечислены аргументы действия `fscommand()`, предусмотренные в языке ActionScript.

- ▶ `full screen`. Данный аргумент принимает значение `true` или `false`. Если он вводится с аргументом `true`, фильм воспроизводится в полноэкранном режиме без строки меню в верхней части окна; если же с аргументом `false`, фильм воспроизводится в окне, размеры которого соответствуют параметрам `Dimensions`, установленным в диалоговом окне `Document Properties`.
- ▶ `allowscale`. Применяя данный аргумент со значением `true`, вы можете изменять масштаб отображения кадров фильма вместе с окном, где он воспроизводится. При использовании аргумента `allowscale` со значением `false` размеры окна, в котором воспроизводится фильм, могут быть изменены, но масштаб отображения кадров фильма при этом изменять нельзя.
- ▶ `showmenu`. Используя данный аргумент со значением `true`, вы разрешаете пользователю доступ к контекстному меню при воспроизведении фильма. Открыть это меню можно посредством щелчка правой кнопкой мыши (`Windows`) или щелчка при нажатой клавише `Ctrl` (`Macintosh`). С помощью команд данного меню можно воспроизводить или останавливать фильм, осуществлять перемотку, масштабировать окно, а также изменять качество воспроизводимого фильма. При использовании аргумента `showmenu` со значением `false` все команды меню кроме пункта `About Macromedia Flash Player` становятся недоступными. Следует отметить, что на некоторых операционных системах аргумент `showmenu` ведет себя непредсказуемо. Поэтому при использовании данного аргумента до распространения фильм необходимо протестировать на различных платформах.
- ▶ `trapallkeys`. Со значением `true` аргумент `trapallkeys` блокирует клавиатуру. Если вы хотите использовать данное действие для предотвращения некорректного ввода при работе с интерактивными фильмами, обязательно предусмотрите возможность завершения работы приложения. Это можно осуществить, добавив кнопку или другой элемент со строкой-оператором `fscommand("quit")`. По умолчанию используется значение `false`, при этом фильм может принимать данные, вводимые с клавиатуры.
- ▶ `exec`. Данный аргумент используется для запуска другого приложения из Flash-фильма. Новое приложение будет открыто поверх окна, где воспроизводится фильм. Значение для этого аргумента определяется как строка, содержащая полный путь к приложению, которое необходимо запустить. Например:

```
on(release){
 fscommand("exec", "Burroughs: Applications (Mac OS 9):SimpleText");
}
```

Представленный выше оператор содержит полный путь для запуска приложения `SimpleText` на компьютере `Macintosh`. Различные компоненты пути разделяются с помощью символа двоеточия (`:`).

На компьютерах Windows в качестве разделителя компонентов пути применяется символ косой черты (/). Например:

```
fscommand("exec", "C:/Program Files/Accessories/Worldpad.exe");
```

Аргумент `quit` позволяет закрыть исполняемый файл типа `projector`, при этом его значение не задается.

Что касается синтаксиса, то все аргументы (`quit`, `exec` и др.), а также их значения вводятся как строки. Например:

```
fscommand("fullscreen", "true");
```

Когда применяется такой аргумент, фильм воспроизводится в полноэкранный режиме. Аргумент `fullscreen` и его значение `true` необходимо заключать в двойные кавычки, что делает их текстовой строкой. Если не ввести аргументы как строки, Flash не сможет корректно обмениваться данными с операционной системой, следовательно, требуемые действия не будут выполнены.

Лучшим местом для действия `fscommand()` является первый кадр на главной временной шкале. При таком вводе данное действие выполняется до того, как будут вызваны другие сценарии `ActionScript`. Кроме того, при помещении действия `fscommand` в первый кадр фильма, сценариям не потребуется обработчик, и они будут загружены в память как одни из первых параметров фильма.

## Эффект letterbox

Используя действие `fscommandO`, вы можете создавать фильмы с уникальными интерфейсами. В этом разделе мы расскажем о некоторых техниках, которые применяют, чтобы создавать стильные и функциональные фильмы, предназначенные для воспроизведения на различных платформах.

Первая техника включает создание файла типа `letterbox-style projector`. `Letterbox` — это термин, используемый для описания внешнего вида изображения. В кинотеатре вы смотрите фильмы на экране, который значительно отличается от экрана вашего монитора или телевизора, — в отличие от характеристического отношения телевизионного экрана 4:3, экраны кинотеатров имеют характеристическое отношение 16:9. Эти экраны растянуты по горизонтали и позволяют просматривать широкоформатные фильмы. То есть, когда киностудия готовит фильм для показа по телевидению или для выпуска на видеокассетах стандарта VHS, размеры фильма модифицируются в соответствии с отношением 4:3.

При выпуске в формате `letterbox` можно не делать такой модификации, поскольку данный формат позволяет сохранить геометрические пропорции изображения 16:9. В этом случае изображение занимает всю ширину экрана, однако сверху и снизу экран приблизительно на одну треть остается черным. Следовательно, изображение в таком формате будет меньше, чем при изначальном создании с характеристическим отношением 4:3, но геометрические пропорции экрана сохранятся. Обычно формат `letterbox` используют вынужденно для сохранения пропорций изображения, но иногда с его помощью можно получить интересные визуальные эффекты.

Создать эффект letterbox довольно просто. Для этого требуется лишь немного поработать ваш фильм следующим образом.

1. Выберите команду Document меню Modify. В диалоговом окне Document Properties (Свойства документа) установите размеры фильма таким образом, чтобы они соответствовали отношению 16:9 (например, 320x180 или 640x360). Поскольку при изменении размеров уже созданного фильма всегда возникают проблемы с расположением и выравниванием элементов фильма на области действия, размер изображения лучше задавать до создания Flash-фильма.
2. В образце цвета фона Background color на панели Properties выберите черный цвет, даже если вы хотите, чтобы фоновый цвет фильма в дальнейшем был другим. При создании автономного файла типа projector фоновый цвет фильма будет использоваться для определения цвета монитора во время воспроизведения в полноэкранном режиме. Чтобы закрыть диалоговое окно Document Properties, щелкните на кнопке ОК.
3. Создайте новый слой фильма и присвойте ему имя letterbox. Переместите созданный слой ниже всех остальных слоев на временной шкале. С помощью команды New Symbol меню Insert создайте прямоугольник, точно совпадающий с размерами области действия. Выберите цвет заливки прямоугольника, который будет использоваться в качестве фона фильма. Выйдите из режима редактирования и переместите символ с изображением прямоугольника из окна библиотеки на слой letterbox. Выровняйте прямоугольник по центру области действия и заблокируйте слой.
4. Откройте панель Actions, выбрав команду Actions меню Window. Создайте еще один слой и присвойте ему имя actions. Вставьте в первый кадр этого слоя ключевой кадр и введите следующие строки-операторы:

```
fscommand ("allowscale", "false");
fscommand ("fullscreen", "true");
```

Первая строка-оператор служит для сохранения размеров фильма и запрета изменения масштаба его отображения. Если использовать аргумент allowscale со значением true, можно изменять масштаб отображения кадров фильма по всей ширине экрана. Но эффект letterbox более нагляднее при использовании аргумента false

Вторая строка-оператор содержит аргумент fullscreen со значением true, при использовании которого фильм воспроизводится в полноэкранном режиме, то есть поверх окон всех открытых приложений, и занимает всю площадь рабочего стола. Применение аргумента fullscreen необходимо для создания эффекта letterbox, поскольку при этом можно использовать фоновый цвет фильма (черный) в качестве маски, окружающей зону 16:9 области действия.

5. Опубликуйте свой фильм как файл в формате projector и оцените эффект letterbox в действии.

Установив размеры фильма в соответствии с характеристическим отношением 16:9, вы создали начальную основу эффекта. Благодаря черному цвету фона фильма была создана черная маска. Затем, используя графический символ с изображением прямоугольника, соответствующего размерам области действия, вы

создали впечатление, что фоновый цвет области действия не является черным. В этом и заключается эффект letterbox (рис. 29.4).



**Рис. 29.4.** Воспроизведение Flash-фильма с эффектом letterbox в полноэкранном режиме

#### ПРИМЕЧАНИЕ

При создании эффекта letterbox символ с изображением прямоугольника выполняет функции, соответствующие функциям декорации театрального представления. Как в театре, когда фоновые декорации меняют непосредственно во время представления, вы можете изменять цвет или другие характеристики фонового символа, что создаст эффект изменения фона во время воспроизведения фильма.

Для вашего фильма можно также создать специальное меню управления. Создав серию кнопок (строку меню), как описывалось в главе 13, и ассоциировав с кнопками действия `ActionScript`, вы сможете управлять воспроизведением, остановкой, перемоткой фильма, а также осуществлять другие интерактивные действия. Более подробно о возможностях, позволяющих зрителям управлять воспроизведением фильма, речь шла в главе 12.

Кроме всего вышеперечисленного вы можете создать дополнительный эффект строки меню — дать зрителям возможность отображать и скрывать кнопки управления, когда это необходимо. Такой эффект особенно полезен, когда фильм воспроизводится в полноэкранном режиме, и вы не хотите, чтобы строка меню была постоянно видна. Для отображения и скрытия каких-либо элементов фильма существует множество способов. Рассмотрим пример:

```
onClipEvent(keyDown){
 if(Key.getCode()==Key.SPACE){
 _visible=1;
 }
}
onClipEvent(keyUp){
 if(Key.getCode()==Key.SPACE){
 _visible=0;
 }
}
```



Данный сценарий создает эффект появления и скрытия ассоциированной с ним строки меню типа Movie Clip. Каждый раз при нажатии клавиши на клавиатуре генерируются события `keyDown` и `keyUp`. Сценарий проверяет, нажата ли клавиша пробела. Если это так, строка меню появляется на экране. Исчезает она только после того, как клавиша пробела будет отпущена. Пока клип остается видимым, содержащиеся в нем кнопки доступны, что позволяет зрителям получать доступ к меню управления фильмом посредством нажатия пробела.

Чтобы представленный выше сценарий работал корректно, следует изначально установить значение 0 (или `false`) для свойства `_visible` клипа, содержащего кнопки управления. Добавив этот сценарий к соответствующему клипу, вы сможете управлять видимостью строки меню с помощью клавиши пробела.



В качестве примера фильма, где реализован эффект `letterbox` и содержится появляющееся меню, управляемое посредством нажатия клавиши пробела, посмотрите файлы `projectorDemo` (Macintosh) и `projectorDemo.exe` (Windows), хранящиеся в каталоге Chapter 29 на прилагаемом компакт-диске. Исходный файл этого фильма находится в том же каталоге под именем `projectDemo fla`.

## Разработки

Из материала данной главы вы узнали, что никаких особых трудностей при публикации фильма на компакт-диск не существует, — задача, которая может показаться сложной, на самом деле реально выполнима при наличии необходимых ресурсов. Ниже приведен список источников справочной информации, которая поможет вам при записи фильмов и другого содержимого на компакт-диски.

- ▶ [www.macdisk.com/cdromen.php3](http://www.macdisk.com/cdromen.php3). Данный Web-узел, ориентированный в основном на пользователей операционной системы Macintosh, содержит множество полезной информации, необходимой при записи компакт-дисков, совместимых как с Macintosh, так и Windows. Пользователи Windows, например, могут познакомиться с приложением `MacImage`, которое используется для создания универсальных компакт-дисков, совместимых с несколькими операционными системами.
- ▶ [www.roxio.com](http://www.roxio.com). Компания Roxio является производителем приложения `Toast 5 Titanium`, работающего в среде Macintosh и являющегося средством для записи универсальных компакт-дисков. Более полную информацию о приложении `Toast 5 Titanium` вы найдете в практикуме 8.
- ▶ [www.nero.com](http://www.nero.com). Приложение Nero, работающее в среде Windows, разработано для записи универсальных компакт-дисков. Его производителем является компания Ahead Software. Более подробно о приложении Nero будет рассказано в практикуме 8.
- ▶ [www.simplythebest.net/info/cdburninfo.html](http://www.simplythebest.net/info/cdburninfo.html). На этом Web-узле вы найдете информацию о технических деталях процесса записи данных на компакт-диски, а также другие справочные материалы, касающиеся данной темы.

## Резюме

В данной главе раскрыты характеристики и возможности Flash, как программы для создания переносимых продуктов, распространяемых с помощью различных носителей, кроме Web. Вы узнали, как создавать автономные файлы формата Flash projector, которые являются исполняемыми файлами или приложениями, способными воспроизводить фильм без помощи дополнительного программного обеспечения. В этой главе также было рассказано об элементе языка ActionScript — действии `fscommand`, позволяющем создавать сценарии, обеспечивающие непосредственный обмен данными между фильмом и операционной системой.

После создания фильма, включающего необходимые сценарии, и его публикации в формате автономного файла, вы можете приступить к переносу фильма на компакт-диск. О том, как произвести запись данных на компакт-диск, будет подробно рассказано в практикуме 8.

## Практикум 8

# Создание универсального компакт-диска

Создаваемый компакт-диск называется универсальным, поскольку его содержимое может быть прочитано как на компьютерах Macintosh, так и на компьютерах с операционной системой Windows. Выражаясь техническим языком, этот диск содержит данные в формате HFS для системы Macintosh и данные в формате ISO 9660 для систем DOS/Windows. Структура универсального компакт-диска позволяет определить, какие из файлов будут доступны для каждой из систем. При создании универсального компакт-диска вы можете точно определить его характеристики для каждой из систем. Пользователи каждой из систем будут видеть только файлы, предназначенные для их систем. Благодаря таким свойствам универсальный компакт-диск является прекрасным средством распространения Flash-фильмов.

В данном практикуме описывается процесс создания универсального компакт-диска с использованием приложения Roxio Toast 5 Titanium, недавно выпущенного компанией Adaptec. Приложение Toast 5 выбрано из большого количества подобных программ, поскольку оно отличается простотой в использовании и способностью выполнять различные операции, необходимые при создании многоплатформенных компакт-дисков. В нашем практикуме самому приложению Toast 5 мы уделим не очень много внимания, а больше сосредоточимся на общих вопросах записи данных на компакт-диск. Поэтому приводимая информация будет полезна и тем пользователям, которые используют другие приложения для записи данных на компакт-диски.

---

### ПРИМЕЧАНИЕ

Приложение Toast 5 Titanium предназначено для работы только в среде операционной системы Macintosh. Пользователи системы Windows найдут необходимую информацию в разделе «Дополнительные сведения».

---

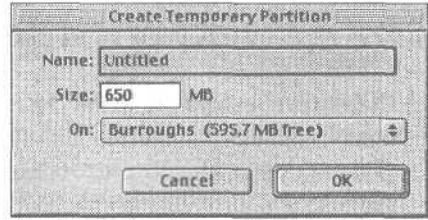
Для выполнения задания практикума необходимо создать два отдельных Flash-фильма в виде файлов формата projector. Один из них предназначен для Macintosh, а второй — для Windows. Если у вас нет собственных фильмов такого формата, воспользуйтесь файлами, хранящимися в каталоге Chapter 29 на прилагаемом компакт-диске. Для записи данных на компакт-диск ваш компьютер должен быть оборудован дисководом CD-R или CD-RW.

## Подготовка файлов

Прежде всего, необходимо запустить приложение Toast 5 Titanium. До начала записи данных на компакт-диск поместим все рабочие файлы во временный каталог на жестком диске компьютера.

Чтобы создать временный каталог на жестком диске, выполните следующие действия.

1. Создайте временный каталог, выбрав в меню Utilities команду Create Temporary Partition (Создать временный раздел). Появится одноименное диалоговое окно.
2. В поле Name введите имя, которое затем будет присвоено компакт-дису. В поле Size укажите размер временного каталога. Он должен быть достаточно большим, чтобы поместить все файлы, которые необходимо записать на компакт-диск. Нажмите кнопку ОК.



На рабочем столе появится значок toast, щелкнув на котором вы получите доступ к созданному каталогу. Скопируйте все Macintosh-файлы в новый раздел и поместите их в таком порядке, как они должны располагаться на компакт-диске. Кроме того, вы можете установить форму, размер и расположение окна, в котором будет открываться компакт-диск на рабочем столе.

Чтобы подготовить Windows-файлы, создайте на рабочем столе новый каталог с тем именем, которое вы хотите присвоить Windows-версии компакт-диска. Скопируйте в этот каталог файл формата Windows projector или любые другие файлы, которые должны быть доступны при работе с Windows.

Первый этап завершен и теперь все готово для организации диска.

## Разбивка диска

Ваши файлы уже организованы в соответствующих каталогах и готовы для переноса на компакт-диск. В окне Toast переместите указатель мыши на кнопку Other (Другой). Около указателя мыши появится значок контекстного меню, которое можно открыть, нажав и удерживая кнопку мыши или произведя щелчок при нажатой клавише Ctrl. В этом меню выберите пункт Custom Hybrid.

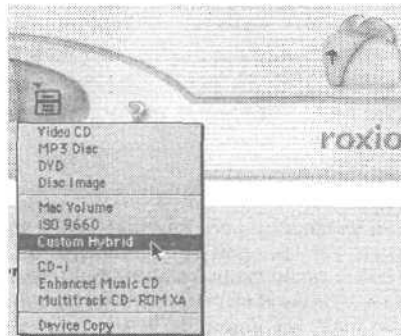
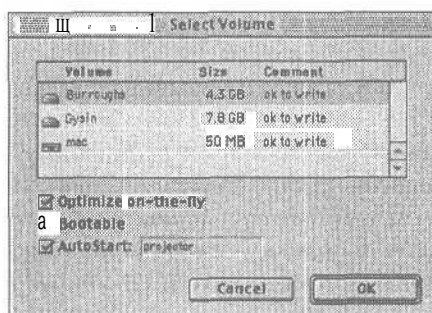


Рис. П8.1. Контекстное меню, открывающееся при нажатии кнопки Other

Диалоговое окно Custom Hybrid позволяет определить внешний вид окна загрузки компакт-диска и содержит две кнопки: одну для установки параметров Macintosh, а вторую для установки параметров ISO.

Чтобы установить параметры для Macintosh-раздела, выполните следующие действия.

1. Щелкните на кнопке Select Mac. Откроется диалоговое окно Select Volume. Выберите раздел, в котором содержатся Macintosh-файлы.
2. Если вы хотите, чтобы исполняемый файл projector (или любой другой файл) запускался автоматически, установите флажок Autostart. В расположенном рядом поле укажите файл, который должен запускаться автоматически. Всю дальнейшую работу приложение Toast произведет самостоятельно.



**Рис. П8.2.** При установке флажка AutoStart файл projector будет запущен автоматически

3. Щелкните на кнопке OK и вернитесь в окно Toast. В разделе Mac окна Toast будет отображаться общий размер данных.

Теперь перейдем к Windows-файлам.

Чтобы указать файлы, которые будут содержаться в Windows-разделе на компакт-диске, выполните следующие действия.

1. Щелкните на кнопке ISO. Откроется диалоговое окно ISO 9660.
2. С помощью мыши перетащите каталог, содержащий Windows-файлы, в раздел вкладки Files. При этом появится значок диска и иерархическая структура Windows-файлов (рис. П8.3). Если вы хотите добавить дополнительные файлы, щелкните на кнопке Add (Добавить) и в открывшемся окне укажите необходимые файлы.

#### ПРИМЕЧАНИЕ

Одним из преимуществ универсального компакт-диска является возможность записи совместно-используемых файлов (файлов, используемых Windows и Macintosh-разделами). Сначала скопируйте такие файлы в Macintosh-раздел. Затем с помощью мыши перетащите их оттуда на вкладку Files окна ISO 9660. Данные файлы будут отмечены голубым цветом, что означает их совместное использование двумя разделами компакт-диска.

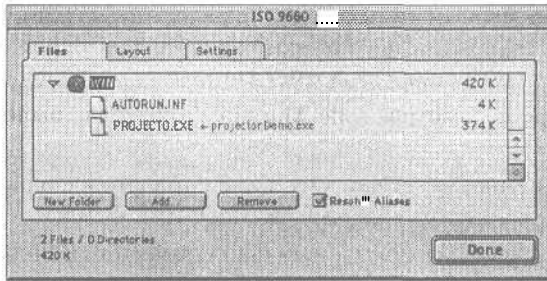


Рис. П8.3. Файлы, предназначенные для копирования в Windows-раздел универсального компакт-диска

3. Перейдите на вкладку Settings (Установки). Откройте меню Naming и выберите пункт Joliet (MS-DOS+Windows). Данная установка позволит вам использовать длинные имена файлов (а также имена, содержащие пробелы) в Windows-разделе на компакт-диске.
4. *Не обязательно.* Для автоматического запуска файла в среде Windows необходимо создать небольшой файл с именем `autorun.inf`. Данный файл содержит инструкцию компьютеру по запуску приложения сразу после чтения содержимого компакт-диска. Для создания такого файла подойдет любой текстовый редактор (например, ВВEdit, Notepad). Данный файл должен содержать следующие строки:

```
[autorun]
open=flashProjector.exe
icon=iconGraphic.ico
```

Имя `flashProjector.exe` замените именем вашего файла, который будет автоматически запускаться после распознавания компакт-диска. Если вы хотите, чтобы и сам диск имел графический значок, укажите во второй строке имя графического файла. Не забудьте поместить указанный файл на компакт-диск. После ввода строк-операторов сохраните файл `autorun.inf` и поместите его в раздел к остальным Windows-файлам.

5. При использовании файла `autorun.inf` необходимо выполнить еще один шаг. Перейдите на вкладку Layout (Размещение), выберите файл `autorun.inf` и щелкните на кнопке Move To Top (Переместить наверх). Данный файл будет помещен в списке файлов первым, что облегчит его поиск при считывании содержимого диска.
6. Для возврата к окну Toast щелкните на кнопке Done (Выполнено). Теперь в разделе ISO окна Toast будет отображаться общий размер данных Windows-раздела.

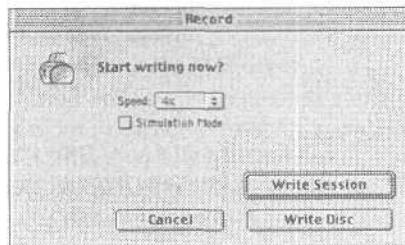
На этом подготовка завершена.

## Запись данных и проверка компакт-диска

На данном этапе все файлы собраны, определена структура диска, и приложение готово начинать запись данных.

Чтобы записать данные на компакт-диск, выполните следующие действия.

1. Убедитесь в наличии записываемого диска в дисковом CD-R или CD-RW. Щелкните на кнопке Record (Запись).
2. После небольшой паузы появится диалоговое окно Record. В раскрывающемся списке Speed выберите оптимальную скорость для вашего записывающего устройства. Если вы хотите произвести тестирование носителя на возможность записи, установите флажок Simulation Mode. Для записи щелкните на кнопке Write Disc.
3. После завершения записи появится диалоговое окно с вопросом, желаете ли вы проверить диск. *Произведение* данной операции рекомендуется, причем занимает оно совсем немного времени. Завершив проверку, извлеките записанный диск из дисковода.



### ПРИМЕЧАНИЕ

Тестирование диска — это важная часть процесса создания компакт-диска. Наилучшим способом проверки является запуск диска на различных компьютерах. При тестировании обратите внимание на то, как работает диск: все ли его свойства функционируют, нет ли медленно работающих приложений.

## Дополнительные сведения

Теперь вы знаете, как создавать универсальный компакт-диск, который можно использовать в качестве носителя для распространения Flash-фильмов. В предыдущих разделах были раскрыты только основные вопросы, связанные с этим процессом. Далее представлена дополнительная информация, которая может оказаться полезной при создании содержимого и записи данных на компакт-диск.

Пользователи операционной системы Windows для записи компакт-дисков могут применять приложение Nero, выпускаемое компанией Ahead Software ([www.nero.com](http://www.nero.com)). Это приложение способно работать в среде Windows 95, 98, Me, NT 4, 2000 и XP. Кроме того, с его помощью записываются компакт-диски различных форматов, включая универсальный ISO/HFS. Для создания универсального компакт-диска с помощью приложения Nero ваш компьютер должен быть оснащен жестким диском SCSI, содержащим раздел HFS с файлами Macintosh, которые будут перенесены на компакт-диск. Более полную информацию о возможностях приложения Nero и требованиях к записывающим устройствам CD-R/CD-RW можно получить, посетив Web-узел компании Ahead Software.

*Многосеансовая запись* — это способ записи на компакт-диск за несколько сеансов, используемый для создания универсальных компакт-дисков. В таком режиме,

когда за один сеанс работы для записи данных используется только определенная часть диска, может работать большинство записывающих приложений, включая Toast. Оставшееся пространство диска может быть использовано в последующих сеансах, то есть во время одного сеанса вы можете записать данные для системы Windows, а во время второго — для Macintosh.

*Расширенные компакт-диски* — это компакт-диски, содержащие как данные, так и звуковую информацию для цифровых аудиопроигрывателей. Такой диск можно создавать как универсальный: в одном сеансе, используя часть диска, записать звуковые данные, а в следующем на оставшейся части записать универсальные данные (Flash-фильмы, HTML-документы и т. д.). Проигрыватели музыкальных компакт-дисков не могут распознать несколько разделов, поэтому им доступен только первый фрагмент информации на диске, то есть музыкальные данные. Но на компьютере будут доступны все данные, содержащиеся на диске. Эту технику используют музыканты, артисты и художники, работающие с мультимедиа, поскольку им необходимо на одном носителе распространять информацию различного типа.

Чтобы на компьютере пользователя компакт-диск запустился *автоматически*, вы должны активизировать флажок Autostart (Macintosh) и Autorun (Windows). Многие пользователи по разным причинам отключают данную функцию. Поэтому, если на компьютере пользователя отсутствует автоматический запуск файлов, рекомендуем добавить файл на компакт-диск с инструкцией по запуску фильма формата projector.





# Приложения

*В этом разделе вы найдете дополнительные сведения, которые позволят вам углубить навыки работы с Flash и ActionScript. В приложении вы найдете пошаговые инструкции для создания навигационных элементов управления и анимационных изображений, которые информируют пользователя о ходе загрузки. В справочнике по ActionScript приведены основные операторы этого языка, их синтаксис и способ применения.*

|                                                              |     |
|--------------------------------------------------------------|-----|
| <b>Приложение А.</b> Создание дополнительных элементов ..... | 618 |
| <b>Приложение Б.</b> Справочник по языку ActionScript .....  | 639 |

## Приложение А

# Создание дополнительных элементов

Изучать Flash можно по-разному. Вы можете овладеть теорией, чтобы использовать полученные навыки для создания собственных продуктов. Другой способ изучения Flash состоит в том, чтобы научиться создавать какие-то элементы, а в процессе этого освоить все теоретические знания. Для кого-то предпочтительнее первый способ, а для кого-то — второй. Поэтому мы постарались охватить их оба, помогая вам узнать как теорию, так и технику создания отдельных элементов. Однако, как вы, наверное, уже убедились, Flash обладает *настолько* разнообразными возможностями, что некоторые из них тематически не *подходят* ни к одной из глав. Не беспокойтесь, мы позаботимся, чтобы вы узнали и о них.

В данном разделе собраны пошаговые инструкции для создания некоторых элементов Flash, о которых еще не рассказывалось в этой книге. К сожалению, если бы мы писали инструкции ко всему, что может быть создано при помощи Flash, нам бесспорно хватило бы работы до следующего ледникового периода. Поэтому приложение содержит только некоторые из них.

В приложении рассматриваются следующие темы:

- ▶ создание навигационных элементов управления;
- ▶ использование анимации для отображения процесса загрузки;
- ▶ использование массива для создания динамического приветствия.

## Создание навигационных элементов управления

Будет ли дизайн любого интерактивного приложения удачным (независимо от того, каким образом и для каких целей оно создано), зависит от системы навигации в нем. В данном случае речь идет не только о кнопках — система навигации определяется не только тем, как пользователь может перемещаться по Web-узлу, но и тем, как представлено содержимое этого узла. Поэтому эффективная система навигации может иметь различные формы.

Для чего нужна эффективная (а иногда и новаторская) навигационная система? Да хотя бы для того, чтобы у пользователей не возникало проблем с поиском необходимой им информации и они могли легко получить к ней доступ.

В следующем разделе вы узнаете, как можно создать несколько полезных навигационных элементов управления, которые позволят значительно упростить пользователю процесс работы с вашим приложением.

## Создание раскрывающегося меню


Часто применение раскрывающегося меню позволяет удобно разместить несколько элементов меню под одной кнопкой. Обычно такое меню работает следующим образом. Когда пользователь помещает указатель мыши на основную кнопку, возникает целый набор дополнительных кнопок (обычно расположенных вертикально). Теперь указатель мыши можно перевести на любую из появившихся кнопок (мы так и будем именовать их — *дополнительные кнопки*) и щелкнуть на ней. Если указатель мыши убрать с дополнительных кнопок, они исчезнут и останется только основная кнопка меню.



Пример раскрывающегося меню содержится в каталоге APPENDIX A на компакт-диске. (Он приведен в двух форматах: SWF и FLA.)

## Создание визуальных элементов



Создавать раскрывающиеся меню довольно легко. Для этого нужно создать клип, несколько кнопок и элементарный сценарий ActionScript.

1. Поскольку меню является самостоятельным клипом (символом типа Movie Clip), сначала необходимо создать этот клип. Выберите команду New Symbol меню Insert, установите переключатель типа символа в положение Movie Clip, укажите имя меню (меню) в поле Name и щелкните на кнопке ОК. Вернитесь к главной временной шкале.
  2. Теперь приступим к созданию кнопки главного меню — назовем ее *основная кнопка*. Выберите команду New Symbol меню Insert, установите переключатель типа символа в положение Button, в поле Name введите имя primary menu button (основная кнопка меню) и щелкните на кнопке ОК.
  3. Оказавшись в режиме редактирования символа, создайте какую-нибудь кнопку меню. Для примера мы выбрали самую простую прямоугольную кнопку. После создания кнопки вернитесь к главной временной шкале.
- 
4. Теперь можно приступать к созданию дополнительных кнопок. Выберите команду New Symbol меню Insert, установите переключатель типа символа в положение Button, введите имя secondary menu button (дополнительная кнопка меню) в поле Name и щелкните на кнопке ОК.
  5. В режиме редактирования символов создайте первую кнопку (кнопки будут располагаться вертикально).
  6. Процесс, описанный в пунктах 4–5, следует повторять до тех пор, пока вы не получите необходимое количество дополнительных кнопок. Помните, что при создании каждой кнопки ей должно быть присвоено уникальное имя.

## Соединение визуальных элементов

Создав основную и дополнительные кнопки, вам необходимо соединить их вместе.

1. В окне библиотеки дважды щелкните на значке клипа menu, созданного вами в самом начале (до сих пор он оставался не востребовавшимся). В первый ключевой кадр единственного слоя (который следует переименовать в primary menu button) поместите основную кнопку.

2. Выберите второй кадр слоя primary menu button и вставьте в него новый кадр, выбрав команду Frame в меню Insert.
3. Добавьте новый слой и назовите его secondary button 1.
4. Выбрав команду Keyframe меню Insert, поместите ключевой кадр во второй кадр слоя secondary button 1.
5. Выберите только что созданный ключевой кадр и вставьте в него символ secondary button. Убедитесь, что левый край дополнительной кнопки прилегает к правому краю основной кнопки, находящейся в слое primary menu button. На рисунке показано, как должна выглядеть дополнительная кнопка по отношению к основной.
 
6. Добавьте новый слой, назвав его secondary button 2, выберите второй кадр и поместите в него ключевой кадр.
7. Поместите символ следующей созданной вами дополнительной кнопки во второй ключевой кадр слоя secondary button 2. Убедитесь, что верхний край этой дополнительной кнопки прилегает к нижнему краю первой дополнительной кнопки, но не перекрывает его.
8. Продолжайте добавлять дополнительные слои, помещая изображения дополнительных кнопок во второй ключевой кадр.
9. Выполнив все эти действия, вы должны получить следующее изображение.
 

## Создание сценария ActionScript

Связав воедино все визуальные элементы, приступайте к созданию сценария, который будет управлять работой вашего меню.

1. Выберите основную кнопку и откройте панель Actions (Window ▸ Actions).

### ПРИМЕЧАНИЕ

Создание сценария ActionScript в соответствии с этой инструкцией выполняется в режиме Normal. Чтобы создать сценарий вручную, вам придется работать в режиме Expert.

2. Добавьте действие gotoO (находящееся в подкатегории Movie Control). Затем активизируйте флажок Roll Over в секции Event (убедитесь, что не выбраны другие флажки). Для этого должна быть открыта область отображения параметров, а в области отображения сценария выбрана строка, содержащая обработчик события сценария.
3. Затем выберите рабочую строку сценария, убедитесь, что в раскрываемом списке Type выбран пункт Frame Number (Номер кадра), введите значение 2 в поле Frame и удостоверьтесь, что установлен переключатель Go to and Play. Таким образом вы переместите головку воспроизведения в кадр 2, благодаря чему при помещении указателя мыши на основную кнопку меню на экране будут появляться дополнительные кнопки.

4. Если вы на данном этапе протестируете клип, то увидите, что выполняется цикл, во время которого дополнительные кнопки быстро появляются и исчезают при переходе головки воспроизведения от кадра 1 к кадру 2 и затем обратно к кадру 1. Вам необходимо «заставить» клип остановиться на кадре 1 и подождать, пока пользователь поместит указатель мыши на основную кнопку меню (действие которой переведет головку воспроизведения в кадр 2, вызвав появление дополнительных кнопок меню). Этого можно добиться при помощи действия `stop()`. Следовательно, вам необходимо создать в клипе новый слой и назвать его `actions` (действия).
5. Выберите первый пустой ключевой кадр слоя `actions` и откройте панель `Actions` (`Window ▶ Actions`).
6. Добавьте действие `stop()`, находящееся в подкатегории `Movie Control` категории `Actions`.
7. То же самое необходимо проделать с кадром 2, чтобы головка воспроизведения не возвращалась автоматически к кадру 1. Для этого выберите второй кадр слоя `actions` и вызовите команду `Keyframe` меню `Insert`.
8. Откройте панель `Actions` (`Window ▶ Actions`) и добавьте действие `stop()` из подкатегории `Movie Control`.
9. Теперь пора создать механизм, который будет убирать с экрана дополнительные кнопки меню, как только пользователь переместит с них указатель мыши. Для этого следует создать невидимую прямоугольную кнопку, размеры которой будут превышать размеры всего меню (когда видимы все его дополнительные кнопки). Этой кнопке будет назначено действие `goto()` для перемещения головки воспроизведения обратно к кадру 1. Благодаря этому будет создаваться впечатление, что дополнительные кнопки исчезают. Кнопка будет находиться под меню и активизироваться тогда, когда пользователь убирает указатель мыши с дополнительных кнопок. Для создания этой невидимой кнопки вызовите команду `New Symbol` меню `Insert`, установите переключатель типа символа в положение `Button`, укажите имя `invisible button` (невидимая кнопка) в поле `Name`, нажмите кнопку `OK` (и перейдите тем самым в режим редактирования символа).
10. Выберите кадр `Hit`, вставьте ключевой кадр и создайте прямоугольник, который по ширине и высоте будет превышать на 25 пикселей размеры меню.

---

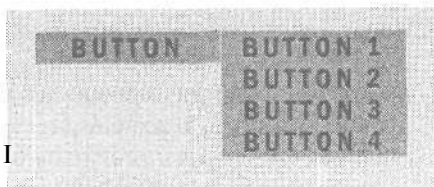
#### ПРИМЕЧАНИЕ

Поскольку эта кнопка будет невидимой, не имеет значения, какой цвет вы будете использовать при ее создании.

---

11. Вернитесь к временной шкале клипа `menu`. Добавьте новый слой и назовите его `invisible button`. Щелкните на этом слое и перетащите его, чтобы он оказался под всеми существующими слоями.
12. Выберите кадр 2 слоя `invisible button` и поместите в него ключевой кадр.

13. Пока только что созданный ключевой кадр остается выбранным, щелкните на значке символа `invisible button` и перетащите его из окна библиотеки на область действия. Расположите его так, чтобы он находился под меню. Сцена должна иметь примерно следующий вид.



14. Теперь нам нужен сценарий ActionScript, с помощью которого головка воспроизведения перейдет обратно к кадру 1 (симулируя исчезновение дополнительных кнопок меню), как только пользователь уберет указатель мыши с дополнительных кнопок. Для этого сначала выберите невидимую кнопку, а затем откройте панель Actions (`Window ▸ Actions`).
15. Добавьте действие `goto()`. Затем в разделе Event области отображения параметров активизируйте флажок `Roll Over` (убедитесь, что другие флажки не выбраны). При этом в области отображения сценария должна быть выбрана строка, содержащая обработчик события.
16. Затем в раскрывающемся списке `Type` выберите пункт `Frame Number` (Номер кадра), введите 1 в поле `Frame` и удостоверьтесь, что установлен переключатель `Go to and Play`.

Поздравляем! Вы создали простое раскрывающееся меню. Вернитесь к основной временной шкале и перетащите клип `menu` на область действия. Чтобы посмотреть на работу меню, выберите команду `Test Movie меню Control`.

## Создание перемещаемого всплывающего окна

При разработке Web-узлов, написанных на языке HTML, можно создавать всплывающие окна при помощи Dynamic HTML (DHTML). При этом вы можете самостоятельно устанавливать размер таких окон. С их помощью очень удобно обеспечивать пользователя информацией, которую необходимо разместить на небольшой площади.

Во Flash подобные окна можно использовать, создавая клипы, содержащие окна, которые будут появляться при щелчке мышью на кнопках (или других элементах) и положение которых можно будет изменять при помощи щелчка и последующего перетаскивания мышью. Суть в том, что, в отличие от HTML, где положение окна фиксируется браузером, его аналог во Flash представляет собой перемещаемое всплывающее окно, положение которого вы можете определять самостоятельно.

Из следующих разделов вы узнаете, как создавать такие окна.

### ПРИЛОЖЕНИЕ-

Поскольку клип является частью Flash-фильма, пользователи не смогут переместить мышью содержащееся в нем окно за пределы окна отображения SWF-файла.



Пример перемещаемого всплывающего окна в двух форматах, SWF и FLA, содержится в каталоге APPENDIX A на прилагаемом компакт-диске.

## Создание всплывающего окна и кнопок

Первым шагом в создании перемещаемого всплывающего окна будет создание кнопки, после щелчка на которой пользователь сможет перемещать на экране появившееся всплывающее окно. Следуя этой инструкции, вы создадите простую прямоугольную кнопку, прилегающую к правому верхнему углу всплывающего окна.

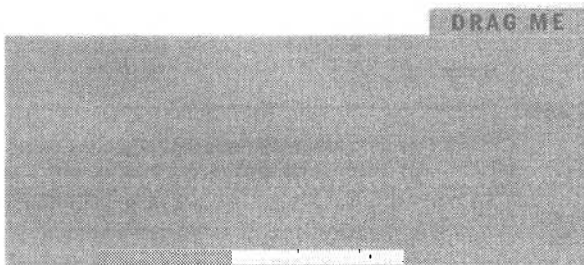
1. Выберите команду **New Symbol** меню **Insert**. Когда появится диалоговое окно **Create New Symbol**, убедитесь, что переключатель типа символов установлен в положение **Button**, укажите имя **drag button** (кнопка перемещения) в поле **Name** и щелкните на кнопке **OK**.
2. В режиме редактирования символа создайте простую прямоугольную кнопку следующего вида. После чего вернитесь к главной временной шкале.



### ПРИМЕЧАНИЕ

Вы, конечно, можете предусмотреть любое сочетание клавиш, нажатие которых будет аналогично щелчку на кнопке перемещения, но для корректного функционирования всплывающего окна это необязательно.

3. Теперь, когда есть нужная кнопка, пора приступить к созданию самого окна. Выберите команду **New Symbol** меню **Insert**. Когда появится диалоговое окно **Create New Symbol**, установите переключатель типа символов в положение **Movie Clip**, введите имя **pop-up window** (всплывающее окно) в поле **Name** и щелкните на кнопке **OK**.
4. В режиме редактирования символов переименуйте единственный существующий слой созданного клипа в **window**. Выберите первый кадр (который автоматически становится ключевым кадром) слоя **window** и создайте прямоугольник, который будет служить окном.
5. Добавьте второй слой и назовите его **actions**.
6. Выберите первый ключевой кадр слоя **actions**, откройте панель **Actions (Window ▸ Actions)** и добавьте действие **stop()** из подкатегории **Movie Control**.
7. Добавьте новый слой и назовите его **drag button**.
8. Выберите первый ключевой кадр слоя **drag button**, переместите символ **drag button** из окна библиотеки на область действия. Расположите изображение кнопки так, чтобы оно находилось вне прямоугольника у его правого верхнего угла. На данном этапе клип (содержащий всплывающее окно и кнопку) должен выглядеть примерно следующим образом.





**СОВЕТ**

Вы можете самостоятельно решать, куда поместить кнопку по отношению к всплывающему окну. Однако помните, что она должна быть как можно ближе к окну, чтобы ассоциироваться с ним у пользователя.

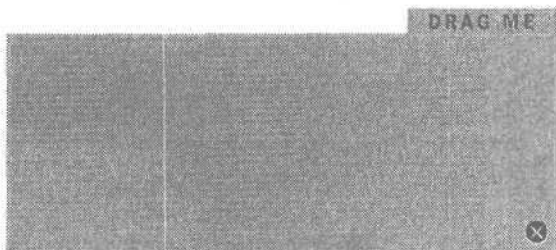
9. Теперь следует дополнить объект Button сценарием ActionScript, который позволит пользователю перемещать всплывающее окно по экрану после щелчка на кнопке перемещения. Сначала выберите эту кнопку. Затем откройте панель Actions, выбрав команду Actions меню Window.
10. В режиме Normal или Expert (решать вам) свяжите с кнопкой следующий сценарий:

```
on (press) {
 startDrag("");
}
on (release) {
 stopDrag();
}
```

**ПРИМЕЧАНИЕ**

Если вы вводите сценарий в режиме Normal, действия `startDrag()` и `stopDrag()` находятся в подкатегории Movie Clip Control категории Actions.

11. Теперь у вас есть кнопка, щелкнув на которой, посетитель сможет перемещать и ее, и само окно. Приступим к созданию кнопки, которая будет закрывать всплывающее окно. Для этого выберите команду New Symbol меню Insert. Когда появится диалоговое окно Create New Symbol, установите переключатель типа символов в положение Button, введите имя close button (закрывающая кнопка) в поле Name и щелкните на кнопке ОК.
12. В режиме редактирования символов создайте кнопку, закрывающую всплывающее окно.
13. Вернитесь к временной шкале клипа pop-up window, создайте новый слой с именем close button, выберите первый ключевой кадр нового слоя и перетащите символ close button из окна библиотеки на область действия. Поместите закрывающую кнопку в окне там, где посчитаете нужным. На данном этапе клип pop-up window будет выглядеть примерно так.

**СОВЕТ**

Как и в случае с кнопкой перемещения, вы можете по собственному усмотрению разместить закрывающую кнопку по отношению к всплывающему окну. Однако помните, что кнопка должна быть в окне, чтобы ассоциироваться с ним у пользователя.

14. Теперь к закрывающей кнопке мы добавим сценарий ActionScript. Сначала выберите названную кнопку, а затем команду Actions меню Window.
15. На открывшейся панели Actions в режиме Normal или Expert введите следующий сценарий:

```
on (release) {
 _root.window._visible - false;
}
```

---

#### ПРИМЕЧАНИЕ

Свойству `_visible` можно также присвоить значение 0 (вместо `false`).

---

По сути, при щелчке пользователем на закрывающей кнопке свойству `_visible` клипа `pop-up window` присваивается значение `false`, в результате чего клип становится невидимым.

## Интеграция окна в фильм

Созданный клип со всплывающим окном необходимо связать с основной временной шкалой фильма. Для этого понадобится кнопка (связанная с небольшим сценарием ActionScript), после щелчка на которой будет появляться всплывающее окно. Вам также придется связать сценарий с первым ключевым кадром основной временной шкалы. Этот сценарий будет делать всплывающее окно изначально невидимым. Ведь не может же оно быть на экране еще до того, как пользователь щелкнет мышью на кнопке?

Для интеграции окна в основную временную шкалу вам необходимо выполнить следующие шаги.

1. Перейдите к основной временной шкале. Существующий первый слой переименуйте в слой `actions` и поместите действие `stop 0` в его первый ключевой кадр.
2. Выберите команду `New Symbol` меню `Insert`. Когда появится диалоговое окно `Create New Symbol`, установите переключатель типа символов в положение `Button`, укажите имя `open button` (кнопка открытия) в поле `Name` и щелкните на кнопке `OK`.
3. В режиме редактирования символов создайте кнопку, при щелчке на которой будет появляться всплывающее окно.
4. Закончив, вернитесь к основной временной шкале. Создайте новый слой с именем `open button`. Перетащите символ `open button` из окна библиотеки на область действия, при этом должен быть выбран первый ключевой кадр только что созданного слоя.
5. Разместив открывающую кнопку по своему усмотрению, свяжите с ней сценарий, который будет вызывать появление всплывающего окна на экране. Для этого выберите эту кнопку и вызовите команду `Actions` меню `Window`.

6. В открывшейся панели Actions введите следующий сценарий (в режиме Normal или Expert):

```
on (release) {
 window._visible - true;
```

---

#### ПРИМЕЧАНИЕ

**СВОЙСТВУ** `_visible` можно присвоить значение `1` вместо `true`.

---

При щелчке пользователя на открывающей кнопке свойству `_visible` присваивается значение `true`, в результате чего всплывающее окно становится видимым.

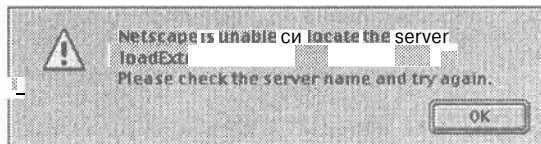
7. Теперь необходимо добиться того, чтобы всплывающее окно было невидимо в начале воспроизведения фильма. Для этого в первый ключевой кадр основной временной шкалы нужно добавить простой сценарий. Сначала в слое actions выберите первый ключевой кадр и вызовите команду Actions меню Window.
8. На панели Actions введите следующий сценарий *перед* уже добавленным вами действием `stop()`.
- ```
window._visible - false
```
9. Когда головка воспроизведения попадает в первый кадр, свойству `_visible` присваивается значение `false`, в результате чего всплывающее окно становится невидимым. После щелчка пользователем на открывающей кнопке оно будет открыто.
10. Создайте новый слой и назовите его pop-up window.
11. В слое pop-up window выберите первый ключевой кадр и перетащите символ pop-up window из окна библиотеки на область действия.
12. Мы подошли к самому важному моменту. Символу pop-up window требуется присвоить имя его экземпляра. Для этого выберите клип в области действия и откройте панель Properties, выбрав команду Properties меню Window. Введите имя window в поле Instance Name.

Вот вы и создали перемещаемое всплывающее окно. Чтобы посмотреть на его работу, выберите команду Test Movie меню Control.

Открытие локального HTML-файла с компакт-диска Flash

Одно из величайших достоинств Flash заключается в том, что программа позволяет создавать проекты, работающие как на компьютерах под управлением Windows, так и на компьютерах Macintosh. Это позволяет распространять Flash-фильмы на универсальных компакт-дисках, благодаря чему сфера их распространения значительно увеличивается. Компакт-диски могут хранить много файлов и позволяют избежать проблем со скоростью подключения и установки связи, характерных для распространения Flash-фильмов через Web.

Работая во Flash с фильмом, находящимся на компакт-диске, вы поймете, что процесс работы с таким фильмом похож на работу с продукцией, предназначенной для Интернета, за исключением нескольких небольших отличий. С одним из них вы столкнетесь, попытавшись получить доступ к HTML-файлу, хранящемуся на компакт-диске. Если во Flash-фильме есть содержимое, которое следует просматривать в Web-браузере, вам необходимо указать для него URL. В случае, когда Flash-фильм находится на компакт-диске, возникает проблема, потому что Flash приспособлен устанавливать доступ к файлу в Web-среде, а не на локальном компьютере. Действие `getURL()` отлично срабатывает в Web-среде, но не с локальными файлами на компакт-диске. Если вы уже пытались получить доступ к такому файлу, то вам наверняка знакомо возникающее при этом сообщение об ошибке.



Другое отличие состоит в том, что при работе с Интернетом ваш SWF-фильм обычно хранится в одном месте. Указав соответствующий URL, вы направляете пользователей к фильму, который хотите им показать. К сожалению, при работе с компакт-диском все обстоит по-другому. Компьютеры имеют различные устройства чтения компакт-дисков. Пользователи могут давать своим жестким дискам различные имена. Одно из решений проблемы заключается в том, чтобы четко указать путь к файлу, как показано ниже.

```
GetURL("file:///Macintosh%20HD/Desktop%20Folder/FlashCD/index.html");
```

Для одного компьютера этот способ прекрасно срабатывает. Однако, если вы собираетесь распространять ваши фильмы среди многих пользователей, вы должны быть уверены в том, что любой из них сможет получить доступ к локальным HTML-файлам вашего проекта. Для этого нужно определить полный путь к файлу, который вы хотите вывести на экран. Только тогда ваш фильм сможет правильно отыскать нужные файлы на компакт-диске. Именно поэтому при разработке фильмов, предназначенных для распространения на компакт-дисках, решающим фактором их правильной работы является указание полного пути. В случае, когда все файлы распространяемого проекта хранятся на компакт-диске, правильно указав его корневой каталог, вы тем самым обеспечите доступ ко всем файлам, связанным с фильмом. Структура компакт-диска не меняется, чего не скажешь о конфигурации пользовательских компьютеров. Следовательно, без указания полного пути никак не обойтись.

Определение полного пути

Для получения полного пути к файлам необходимо добавить соответствующую функцию.

1. Создайте новый фильм (File ▶ New). Сохраните созданный файл под именем `HTML_Launcher`. Щелкните на пустом ключевом кадре слоя Layer 1 и вызовите команду Actions меню Window. На панели Actions введите следующее выражение:
`trace(_url);`

Выберите команду Test Movie меню Control. В окне Output появится строка — это и есть полный путь к файлу HTML_Launcher. Свойство _url возвращает полный путь к SWF-файлу как строку. Обратите внимание на то, что выражение HTML%5Flauncher.swf записано в конце строки, содержащей URL. Поскольку Flauncher.swf — это имя самого файла, выражение перед ним и является полным путем к файлу.

ПРИМЕЧАНИЕ

Последовательность символов %5F в URL является эквивалентом знака подчеркивания (_). Дополнительную информацию о работе с закодированными аналогичным образом строками можно получить, обратившись к функциям escape() и unescape() на панели References Flash MX.

2. URL для фильма извлечен, и у вас есть, с чем работать. Получив доступ к фильму, вы должны найти все связанные с ним файлы.

Возвратитесь к панели Actions, на которой уже содержится действие trace(), удалите его и введите следующее выражение:


```
function moviePath(){
    var lastSlash=_url.lastIndexOf("/");
    var path=_url.substring(0,lastSlash+1);
    return path;
}
```

Эта функция возвращает полный путь к папке, в которой находится файл HTML_Launcher. Рассмотрим используемый для этого сценарий подробнее.

- > В первой строке объявляется функция moviePath(). Этой функции не требуются аргументы, поэтому скобки после нее остаются пустыми.
- > В следующей строке кода инициализируется переменная lastSlash. Метод lastIndexOf сохраняет номер позиции последнего символа (/) в URL фильма.
- > В третьей строке инициализируется переменная path. С помощью метода subString() создается новая строка на основе URL фильма. Эта строка включает начало URL, позиция 0, и заканчивается позицией, в которой символ (/) встречается в последний раз, номер которой увеличен на 1. Выражение lastSlash+1 гарантирует, что последний символ (/) будет включен в создаваемую строку. В противном случае URL фильма был бы записан неправильно.
- > Оператор return в четвертой строке кода позволяет функции передать строку, содержащуюся в переменной path, другим частям фильма.

Теперь функцию можно вызвать из вашего фильма. Следует отметить, что сама функция находится в первом кадре первой сцены фильма. Поэтому считается она одной из первых в начале воспроизведения и доступна в любое время воспроизведения.

Запуск внешнего HTML-файла

 Изучение этого раздела требует предварительной подготовки. Для начала на прилагаемом компакт-диске откройте каталог APPENDIX A и скопируйте файл `LOADEXTERNAL.HTML` в каталог, в котором находится файл `HTML_Launcher`. Вместо того чтобы копировать файл, вы, при желании, можете создать собственный HTML-документ. Кроме того, вам понадобится кнопка. Вы можете создать ее такой, какой захотите. После завершения указанных действий можно перейти непосредственно к выполнению инструкции запуска.

1. Перетащите кнопку запуска на область действия. Выберите ее и вызовите команду `Actions` меню `Window`. В открывшейся панели `Actions` введите следующее выражение:

```
on(release){
  var link=_root.moviePath();
  getURL(link+"loadExternal.html");
}
```

В этом выражении вызывается функция `moviePath()` и с помощью действия `getURL()` запускается HTML-файл. В переменной `link` сохраняется значение, возвращаемое функцией `moviePath()`, — полный путь к фильму. Поскольку функция была создана во временной шкале основного фильма, в выражении присутствует ссылка `_root`, благодаря которой обращение к функции становится корректным. Затем действие `getURL()` запускает HTML-файл, дополняя путь к файлу (сохраненный в переменной `link`) именем файла. В качестве аргументов действия `getURL()` могут быть использованы строки. Поскольку в переменной `link` уже содержится строка, возвращаемая функцией `moviePath()`, она легко дополняется именем HTML-файла, образуя путь к нужному файлу.

2. Сохраните фильм и выберите команду `Test Movie` меню `Control`. После щелчка на кнопке запуска HTML-файл откроется в окне браузера, установленного на вашем компьютере.
3. Такой способ запуска файла является очень гибким. Чтобы подтвердить это, проведем следующий эксперимент. Закройте приложение `Flash` и перейдите к каталогу, в котором сохранены ваш фильм и HTML-файл. Скопируйте или переместите файлы `HTML_Launcher` и `LOADEXTERNAL.HTML` в другой каталог. Откройте SWF-файл, щелкните на кнопке и посмотрите, что произойдет. Поскольку написанный вами сценарий динамически определит полный путь к фильму, последний может находиться в любом месте. При помощи свойства `_url` всегда можно установить местонахождение фильма. Можно убрать имя файла из URL и использовать полные пути для получения доступа к файлам поддержки, другим фильмам и т. д.

Важно помнить, что функция обращается к каталогу, в котором находится основной фильм, а указать нужный файл вы можете сами. Например, если ваш фильм находится в корневом каталоге компакт-диска, а HTML-файл — в каталоге `webDocs`, нужно немного изменить сценарий. Теперь он будет содержать следующее выражение.

```
getURL(link+"webDocs/loadExternal.html");
```

Поскольку расположение файлов на компакт-диске не может измениться, подобное «программирование» пути вполне безопасно. Правильно указав путь к файлу в сценарии, вы всегда сможете получить к нему доступ.

МХ Эта инструкция касалась загрузки внешнего HTML-файла. Однако Flash MX поддерживает различные внешние носители. Теперь вы можете загрузить графические файлы JPEG и звуковые MP3 из внешнего источника непосредственно во Flash-фильмы. Воспользуйтесь описанной в этом разделе технологией, чтобы в вашем фильме всегда можно было определить местонахождение внешних файлов. Например:

```
var photo=root.moviePath();
loadMovie(photo+"loader_images/image1.jpg", "frame");
```

В данном случае вызывается функция `moviePath()`. Возвращаемое ею значение сохраняется в переменной `photo`. Путь к файлу в его окончательном виде передается в действие `loadMovie()`, которое импортирует файлы из каталога `loader_images` и загружает их в экземпляр клипа `frame`. Дополнительную информацию о загрузке изображения в формате JPEG из внешнего источника можно получить в главе 17; о загрузке внешних файлов MP3 рассказывается в главе 22.

Анимации, используемые для отображения процесса загрузки

С увеличением сложности самой программы Flash изменились и создаваемые в пей файлы. Теперь для их загрузки уже требуется определенное время. Вследствие этого широкое применение получили анимационные изображения, используемые для отображения процесса загрузки. В сущности, это небольшая анимация, которая воспроизводится, пока загружается сам фильм, и развлекает зрителей. В противном случае им пришлось бы смотреть на пустой экран в окне браузера и гадать, не вышел ли из строя компьютер. По окончании процесса загрузки демонстрация анимации прекращается и начинается воспроизведение фильма.

Существуют два вида анимации, используемой для отображения процесса загрузки: простая и сложная. Простая — это анимация (возможно, повторяющаяся несколько раз) без указания на то, какая часть фильма уже загружена. Сложная не только воспроизводит повторяющуюся анимацию, но и отображает информацию о степени загрузки самого фильма в виде индикатора загрузки.

Создание простой анимации для отображения процесса загрузки

Самым простым способом занять вашу аудиторию во время загрузки фильма является применение простой повторяющейся анимации.

СОВЕТ

Поскольку анимация воспроизводится, пока загружается основной фильм, она сама должна быть небольшой по размеру (имеется в виду размер файла) и загружаться мгновенно. В противном случае аудитории придется ждать, пока загрузится сама анимация, а это делает ее применение бессмысленным.



На прилагаемом компакт-диске в каталоге APPENDIX A есть пример простой анимации, предназначенной для отображения процесса загрузки (в формате SWF и FLA).

Процесс создания простой анимации для отображения процесса загрузки включает использование клипа и сценария ActionScript.

1. Для начала убедитесь, что фильм, к которому вы хотите добавить анимацию, открыт.
2. Откройте панель Scene, выбрав команду Scene меню Windows. (Для удобства предположим, что весь фильм состоит из одной сцены.)
3. Переименуем существующую сцену в main movie (главный фильм).
4. Перед тем как продолжить, выберите последний кадр главного фильма, откройте панель Properties (Window ▶ Properties) и введите метку final в поле Label. Скоро вам станет ясно, зачем это нужно.
5. Добавьте вторую сцену (щелкнув на кнопке Add Scene), назовите ее preloader. Убедитесь, что на панели Scene сцена preloader находится над сценой main movie.
6. Выберите сцену preloader.
7. Выберите единственный существующий слой (Layer 1) и переименуйте его в actions.
8. В слое actions выберите кадр 5 и вставьте ключевой кадр посредством команды Keyframe меню Insert.
9. Откройте панель Properties (Window ▶ Properties). При этом должен быть выбран вновь созданный ключевой кадр.
10. В поле Label введите метку check (проверка).
11. Выберите кадр 10 слоя actions и поместите в него другой ключевой кадр.
12. На панели Properties добавьте метку loop (цикл) в только что созданный ключевой кадр.
13. Выберите кадр 5 (с меткой check) и откройте панель Actions (Window ▶ Actions).
14. Убедитесь, что вы находитесь в режиме Expert (чтобы перейти в этот режим, выберите команду Expert Mode в меню Options панели Actions). В текстовом поле ActionScript введите следующий сценарий.

```
if (_framesloaded<_totalframes) {
    gotoAndPlay("loop");
} else {
    gotoAndPlay("main movie", 1);
}
```

Этот сценарий будет проверять количество загруженных кадров, сравнивая его с общим количеством кадров в фильме. Если число загруженных кадров в момент выполнения сценария меньше общего количества кадров в фильме, головка воспроизведения возвращается к кадру с меткой loop. Как только число загруженных кадров окажется равным общему количеству кадров в фильме, головка воспроизведения перейдет к кадру 1 сцены main movie.

15. Теперь необходимо вставить в фильм сценарий, который обеспечит повторную проверку состояния загрузки фильма (после первоначальной проверки). Сначала выберите кадр 10 (с меткой loop).
16. Откройте панель Actions, добавьте в область отображения сценария действие goto 0 из подкатегории Movie Control.
17. На области отображения параметров в раскрывающемся списке Scene выберите пункт <current scene>, в раскрывающемся списке Type выберите пункт Frame Number, в поле Frame введите 1 и убедитесь, что установлен переключатель Go to and Play. Таким образом, головка воспроизведения автоматически перемещается к первому кадру. От этого кадра головка перейдет к кадру 5 и повторно выполнится сценарий, проверяющий загрузку всех кадров.
18. После ввода сценариев, необходимых для проверки загрузки главного фильма, вам следует создать ту повторяющуюся анимацию, которая будет развлекать аудиторию во время загрузки.
19. Создайте клип с простой анимацией. Например, можно создать попеременно увеличивающееся и уменьшающееся слово Loading. Единственное существенное ограничение в данном случае относится к размеру самого файла клипа, который должен быть небольшим, чтобы его можно было загружать и начинать воспроизведение практически мгновенно.

СОВЕТ

Создайте анимацию таким образом, чтобы она воспроизводилась плавно (то есть чтобы не происходило резких изменений по достижении последнего кадра и переходе обратно к первому).

20. Возвратитесь к сцене preloader.
21. Создайте новый слой и назовите его preloader animation.
22. Щелкните на слое preloader animation и перетащите его под слой actions.
23. При выбранном первом ключевом кадре слоя preloader animation щелкните на значке вашего клипа, содержащего анимацию, и перетащите его из окна библиотеки на область действия. Здесь вы можете поместить его, куда захотите.
24. И наконец, последнее. Вам необходимо поместить действие stop 0 в последний ключевой кадр сцены main movie, чтобы по окончании фильма головка воспроизведения не возвращалась к кадру 1 в первой сцене (то есть сцене preloader). Для этого переключитесь на сцену main movie, добавьте новый слой и назовите его actions.
25. В только что созданном слое выберите первый ключевой кадр, откройте панель Actions (Window ▸ Actions), добавьте действие stop() из подкатегории Movie Control в область отображения сценария.

Итак, вы только что создали простую анимацию, используемую для отображения процесса загрузки. Анимационный клип будет повторяться, пока идет проверка (и, если необходимо, пере проверка) состояния загрузки сцены main movie.

ПРИМЕЧАНИЕ

Поскольку вы запускаете весь фильм с жесткого диска, сцена main movie у вас загрузится практически мгновенно и анимация отображения процесса загрузки едва ли будет воспроизводиться. Если вы хотите посмотреть на ее работу, запустите свой фильм, выберите команду Test Movie меню Control, а затем выберите команду Bandwidth Profiler меню View. Выберите также команду Show Streaming меню View. Помните, что при тестировании фильма вы можете выбрать в меню Debug скорость имитируемого модемного соединения.

Создание сложной анимации для отображения процесса загрузки

Простая анимация, конечно, может несколько занять аудиторию, пока загружается фильм. Однако, если при этом нет никаких указаний на степень загрузки фильма, весь этот процесс может показаться довольно скучным. Вот тут-то и пригодится сложная анимация, позволяющая информировать пользователя о ходе загрузки.

Существует множество способов создания сложной анимации для отображения загрузки. Следуя нижеприведенной инструкции, вы создадите анимацию, в которой показывается, как по мере загрузки фильма заполняется индикатор.



На прилагаемом компакт-диске в каталоге APPENDIX A есть пример сложной анимации для отображения процесса загрузки (в формате SWF и FLA).

ПРИМЕЧАНИЕ

Анимация, которую вы создадите, следуя этой инструкции, выглядит несколько необычно. Индикатор заполняется не слева направо, а с центра и сразу в обе стороны. Мы решили создать такой оригинальный индикатор исключительно по эстетическим соображениям. Существует слишком много индикаторов со стандартным заполнением, а этот поможет вам привлечь внимание пользователей.

Следуя этой инструкции, вы научитесь применять очень полезные методы `getBytesLoaded` и `getBytesTotal`, при помощи которых определяется, какая часть фильма загрузилась, и ведется управление анимацией индикатора загрузки.

1. Открыв фильм, в который вы хотите ввести сложную анимацию, создайте новую сцену (перед самим фильмом) и назовите ее `preloader`.
2. Создайте новый клип и назовите его `preloader bar`. Для этого выберите команду New Symbol меню Insert, установите переключатель типа символа в положение Movie Clip, введите указанное имя в поле Name и щелкните на кнопке ОК.
3. В режиме редактирования символа нарисуйте горизонтальный прямоугольник, залитый каким-то цветом (по вашему выбору). По сути, этот прямоугольник и является индикатором загрузки. Поэтому вам лучше сделать его длинным (скажем, длиной 5 см или 2 дюйма) и тонким. После чего возвратитесь к главной временной шкале.

ПРИМЕЧАНИЕ

Не группируйте обводку и заливку прямоугольника.

4. Откройте окно библиотеки (Window ▶ Library), выберите клип `preloader bar` и создайте его копию, выбрав команду Duplicate меню Options.
5. В открывшемся окне Duplicate Symbol введите имя `preloader bar outline` в поле Name и щелкните на кнопке ОК.
6. Откройте клип `preloader bar outline`, либо дважды щелкнув на его значке мышью в окне библиотеки, либо выбрав его в меню Edit Symbols, расположенном на строке Scene and Symbol.
7. Выберите заливку прямоугольника и удалите ее. Теперь перед вами пустой контур.
8. Вернитесь назад к клипу `preloader bar`. Измените имя слоя, содержащего изображение горизонтального прямоугольника, на `bar`. Затем добавьте ключевой кадр в кадр 2.
9. Добавьте новый слой и назовите его `actions`.
10. Выбрав первый ключевой кадр слоя `actions`, в режиме Expert добавьте следующий сценарий:

```
setProperty (this, xscale, (_root.getBytesLoaded()/
_root.getBytesTotal()*100));
```

11. Этот сценарий определяет степень загрузки фильма в процентах (для этого общее количество байтов фильма делится на количество уже загруженных байтов).
12. Переключитесь назад на главную временную шкалу. Переименуйте существующий слой в `loader` и добавьте дополнительный слой, назвав его `actions`.
13. При выбранном первом ключевом кадре слоя `loader` перетащите символ `preloader bar outline` из окна библиотеки на область действия. При помощи панели Align (Window ▶ Align) выровняйте его, чтобы он оказался в центре области действия.
14. В слое `loader` выберите второй кадр и вызовите команду Keyframe меню Insert.
15. Пока выбран второй ключевой кадр слоя `loader`, перетащите символ `preloader bar` из окна библиотеки на область действия. При помощи панели Align (Window ▶ Align) выровняйте его, чтобы он оказался в центре области действия. Вы должны поместить его в то же место, где уже находится клип `preloader bar outline`. Очень важно, чтобы они совпадали.
16. Разместив все графические элементы анимации, переходите к созданию сценария, который проверяет загрузку кадров фильма. Добавьте ключевой кадр ко второму кадру слоя `actions` и, выбрав его, в режиме Expert добавьте следующий сценарий:

```
if (_root.getBytesLoaded() != _root.getBytesTotal()) {
    gotoAndPlay (2);
}
else {
    gotoAndPlay ("intro", 1);
}
```

17. С помощью этого сценария воспроизведение фильма останавливается на кадре 2 (и сценарий повторяется непрерывно) до тех пор, пока загруженное количество байтов не станет равным общему размеру фильма. После этого начнется показ самого фильма.

ПРИМЕЧАНИЕ

В этом примере действие обращается к первому кадру в сцене под названием intro. При создании собственного индикатора загрузки вместо intro вам следует ввести имя второй сцены вашего фильма (именно в этой сцене содержится главный фильм, к которому вы создаете индикатор загрузки).

18. Теперь вы можете посмотреть, как работает ваш фильм, выбрав команду Test Movie меню Control. Как и в случае с предыдущим примером, вы запускаете весь фильм с жесткого диска, поэтому сцена главного фильма загружается практически мгновенно и анимация для отображения загрузки едва ли будет воспроизводиться. Если вы хотите посмотреть ее в работе, выберите команду Bandwidth Profiler меню View. Также выберите команду Show Streaming меню View.

Использование массива для создания динамического приветствия

Сценарии ActionScript помогают сделать ваши фильмы менее предсказуемыми. Например, обычно текст и графика на Web-узле появляются всегда на одном и том же месте, в одно и то же время и изображаются одними и теми же цветами. Однако вы можете удивить своих посетителей приятным разнообразием. В этом разделе вы узнаете, как использовать сценарии ActionScript для создания массива, отображающего каждый раз различные приветствия при посещении посетителем вашего Web-узла.

Массив представляет собой информационную структуру, состоящую из отдельных «отсеков», в каждом из которых хранится частица (или частицы) информации. В этом смысле массив напоминает шкаф. Массив — это большая структура (в нашей аналогии — шкаф), которая состоит из небольших контейнеров (ящиков). В каждом контейнере может содержаться информация (вещи). Например, в шкафу могут быть ящики для носков, нижнего белья, футболок и свитеров. Для каждой вещи есть свой ящик. Массив во многом подобен шкафу. Только роль ящиков в нем выполняют элементы массива. Каждый элемент массива содержит частицу данных: строку, число, переменную или даже другой массив. Поскольку массив является единым целым, куда его ни перемещают, вся информация переносится вместе с ним. Массив позволяет манипулировать информацией как одним целым, что намного удобнее, чем иметь дело с ее отдельными частями. Короче говоря, массив упрощает хранение и работу с наборами данных, поскольку в нем они представляются в виде строго организованной структуры.

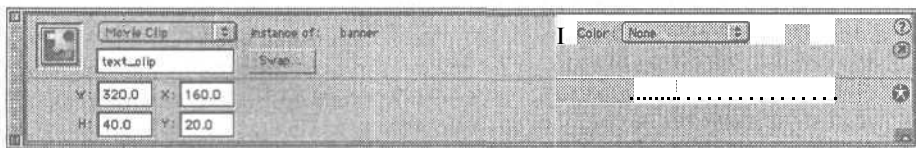
Возможности применения массивов просто поражают. Этой теме можно посвящать не одну, а сразу несколько книг. В этом разделе вы узнаете, как можно создать массив в ActionScript и использовать его для достижения интересных динамических эффектов во Flash-фильмах.



Для выполнения приведенных ниже инструкций вам необходимо скопировать несколько файлов с компакт-диска, прилагаемого к этой книге. Откройте каталог APPENDIX A и скопируйте весь каталог ARRAY на рабочий стол вашего компьютера. Для этого проекта нужно несколько файлов SWF. В нем создается интерактивный интерфейс, или Web-узел международного клуба, или Web-узел какой-то организации. Вы внесете в этот проект изменения с тем, чтобы посетители, просматривающие первую часть фильма, каждый раз видели в левом верхнем углу новый перевод слова *hello* (привет). Откройте файл INTERNATIONAL_CLUB.FLA и выберите команду Test Movie меню Control. Нажимая кнопки, расположенные внизу экрана, вы увидите, что вверху появляются различные слова. Однако, щелкнув на желтой кнопке, вы вообще не увидите никакого текста. Именно для этого случая вы создадите массив.

Для создания массива, выводящего на экран приветствие, необходимо выполнить следующие действия.

1. Откройте файл BANNER.FLA, находящийся в каталоге ARRAY. На данный момент весь каталог ARRAY должен быть сохранен на вашем компьютере. Щелкните в центре области действия. Если она еще не открыта, нажмите Cmd/Ctrl+F3, чтобы открыть панель Properties. Вы увидите, что на самом деле щелкнули не на области действия, а на экземпляре клипа banner, названного text_clip. Свой массив вы создадите в сценарии, присоединенном к этому клипу.



2. При выбранном клипе banner откройте панель Actions, выбрав команду Actions меню Windows.
3. Введите следующий сценарий, который при загрузке клипа в область действия создаст массив greeting (приветствие).

```
onClipEvent (load){
    greeting=new Array();
}
```

4. Щелкните в позиции, следующей сразу после точки с запятой, чтобы начать новую строку, и введите в ней следующее:

```
greeting[0]="Hello";
```

Теперь сценарий будет выглядеть следующим образом:

```
onClipEvent (load){
    greeting=new Array();
    greeting[0]="Hello!";
}
```

Новая строка очень важна, поскольку с ее помощью создается первый элемент массива, в котором хранится строка Hello!. В массиве элементы идентифицируются их порядковыми номерами, начиная с 0.

5. Затем вы можете добавить в массив еще несколько элементов:

```
onClipEvent (load){
    greeting=new ArrayC:
    greeting[0]="Hello!";
    greeting[1]="Bonjour!";
    greeting[2]="Hola!";
}
```

6. Теперь массив носит более интернациональный характер. После того как вы поместите в массив все строки, в нем будет уже три элемента.
7. Итак, в массиве появилось несколько элементов; ими можно воспользоваться. Щелкните в позиции, находящейся сразу после последней точки с запятой, нажмите клавишу Return/Enter и добавьте следующие выражения:

```
choice=Math.floor(Math.random()*greeting.length);
text=greeting[choice];
```

Для чего нужны эти строки? Хороший вопрос. Они являются одним из самых важных элементов описываемого сценария.

- > В переменной `choice` хранится произвольное число, определяемое выражением, находящимся по другую сторону от оператора присваивания (=). Это число всегда будет относиться к диапазону чисел от 0 до последнего номера в массиве.
 - > Метод `length()` извлекает длину массива или количество элементов в нем. Полученное число умножается на произвольное число от 0 до 0,999 и округляется до ближайшего меньшего числа. Поскольку размер текущего массива равен 3, это всегда будет целое число от 0 до 2. Данная информация используется в следующей строке.
 - > При помощи оператора доступа к массиву мы извлекаем элемент массива, номер которого сохранен в переменной `choice`, и помещаем его в переменную `text`. Например, если переменная `choice` имеет значение 2, переменная `text` будет иметь значение, содержащееся во втором элементе массива `greeting`, то есть `Hola!`. Ну, а значение переменной `text` используется в текстовом поле клипа. В нашем случае в поле появится слово `Hola!`.
8. Чтобы проверить полученный результат, выберите команду Control меню Movie. На экране появится фильм, а в нем — одно из трех приветствий, сохраненных в массиве. Закройте фильм и вновь запустите его. Ваши шансы увидеть то же самое приветствие составляют один к трем. Но будем надеяться, что в этот раз вы увидите приветствие на другом языке. Если вы все еще не полностью уверены в работе массива, запускайте фильм вновь и вновь. Каждый раз вы, скорее всего, будете видеть различные приветствия.
9. Добавьте в сценарий еще несколько строк, содержащих приветствия на других языках мира. Окончательный вид сценария показан на рис. АЛ.

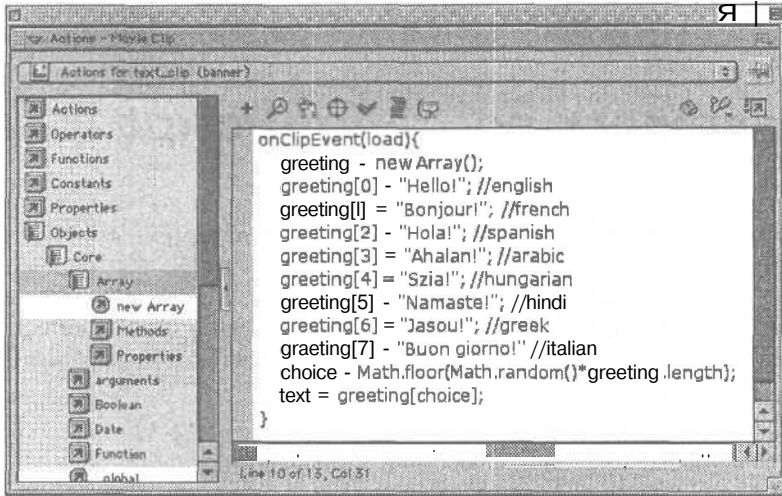


Рис. А.1. В последнем сценарии количество приветствий достигает восьми

10. Сохраните фильм и вновь протестируйте его. Повторив эту процедуру несколько раз, вы увидите приветствия на самых разных языках.
11. Чтобы проверить работу данного фильма совместно с другими фильмами, закройте файл `BANNER.FLA` и откройте `INTERNATIONAL_CLUB.SWF`. Щелкните на желтой кнопке, возвращающей приветствие.

Вы можете увидеть и другие маленькие хитрости `ActionScript`, использованные в этом фильме. Для этого посмотрите сценарии, связанные с коричневой, красной и зеленой кнопками файла `INTERNATIONAL_CLUB.SWF`.

Приложение Б

Справочник по ActionScript

Данная глава не похожа на остальные главы этой книги, но мы надеемся, что и она окажется полезной. Вместо пошагового описания инструкций, применявшегося ранее в этой книге, мы организовали материал так, чтобы вы могли получать информацию по мере необходимости. Например, если существует объект, синтаксис которого вы хотели бы уточнить, или вы хотите что-либо сделать, но не знаете, какие элементы ActionScript лучше для этого использовать, — в этих и многих других случаях вам поможет данный справочник.

В разделах этой главы описываются следующие элементы языка ActionScript.

- ▶ **Действия.** Встроенные команды ActionScript. Они используются для навигации и управления временной шкалой.
- ▶ **Функции.** Подпрограммы, используемые для выполнения общих задач.
- ▶ **Свойства.** Элементы, которые устанавливают и извлекают параметры временных шкал, кнопок и клипов.
- ▶ **Операторы.** Применяются для вычисления и изменения значений выражений.
- ▶ **Объектные ссылки и ссылки на временную шкалу.** Средства, позволяющие получить доступ к элементам фильма.
- ▶ **Предопределенные объекты.** Применяются для манипулирования данными, связанными со встроенными объектами (MovieClip, Sound и т. п.).

Благодаря хорошо продуманной структуре главы у вас не должно возникнуть трудностей, связанных с поиском. Подумайте, что вы хотите сделать. Если необходимо изменить вид чего-либо, тогда вам, скорее всего, нужен раздел «Свойства». Если нужно установить контроль над воспроизведением фильма главной временной шкалы, обратитесь к разделу «Действия». Может случиться, что вы, наоборот, точно знаете термин, но не помните, к какой категории он относится. Такая проблема тоже легко разрешается. Если вы, например, хотите найти действие `startDrag()`, где вы начнете поиск? Подумайте, при помощи действия `startDrag()` вы можете перемещать клип. Это — разновидность управления временной шкалой, следовательно, вам нужен раздел «Действия». В самих разделах термины приводятся в алфавитном порядке.

Справа от каждого элемента в скобках указана версия Flash, начиная с которой его можно использовать.

Этот справочник по ActionScript не полон. До выхода Flash MX ActionScript уже был довольно сложным языком, а с появлением всех нововведений этой программы он стал еще сложнее. Полный справочник по этому языку, наверное, имел бы больший объем, чем вся эта книга. Здесь же вы найдете руководство по применению элементов ActionScript, которые, по нашему мнению, являются *самыми важными*. Без них вам не обойтись.

Действия

Действия представляют собой встроенные команды ActionScript. С помощью действий можно создавать условные операторы и циклы, а также управлять воспроизведением Flash-фильмов.

break (4+)

Синтаксис break

Это действие может оказаться полезным, если вы хотите выйти из цикла до его завершения. Действие break выполняется внутри цикла, цикл разрывается и управление передается первому следующему за ним выражению.

Пример

```
onClipEvent(load){
    taxRate=.0875;
    grandTotal=subTotal*taxRate;
    for(i=1;i<10;i++){
        if(eval("_root.selQty"+i)==0){
            this.attachMovie("alertClip","alert".1);
            break;
        }
        totalQty+=eval("_root.selQty"+i);
    }
}
```

В данном сценарии действие break используется для разрыва цикла и создается новый экземпляр клипа alertClip, если одна из переменных selQty1, selQty2... selQty9 равна нулю. На практике экземпляр клипа alertClip представляет собой своего рода предупреждающее окно с сообщением, информирующим аудиторию о возникновении ошибки ввода данных и о том, как с ней можно справиться. Действие break размещается в сценарии так, чтобы выполнение последнего остановилось перед выполнением выражения, которое обновило бы значение переменной totalQty. Кроме того, действие break вызывается *после* добавления клипа alert. См. также описание действия continue.

continue (4+)

Синтаксис continue

Это действие прерывает выполнение текущей итерации цикла. Затем для операторов while или do..while условие цикла проверяется до его возобновления. Для оператора for сначала увеличивается счетчик цикла, а затем проверяется условие цикла. В конструкции for...in (см. ниже) выполнение цикла возобновляется для следующего свойства. Во избежание ошибок при использовании действия continue внимательно следите за выполнением оператора цикла.

Пример

```
onClipEvent(load){
    taxRate=.0875;
    grandTotal=subTotal*taxRate;
    for(i=1;i<10;i++){
        if(eval("_root.selQty"+i)==0){
            continue;
        }
        totalQty+=eval("_root.selQty"+i);
    }
}
```

В этом сценарии действие `continue` используется для прерывания текущей итерации цикла, если одна из переменных `selQty1`, `selQty2...selQty9` равна нулю. См. также описание действия `break`.

do...while (4+)

Синтаксис

```
do{
    выражение;
} while(условие);
```

В этом цикле истинность условия проверяется после выполнения очередной итерации. Другими словами, тело цикла будет гарантированно выполнено один раз.

Пример

```
x=0;
trace("целые числа от 0 до 9:");
do{
    trace(x);
    x++;
}while(x<10);
```

Данный сценарий выводит на экран фразу целые числа от 0 до 9:, за которой следуют числа 0 1 2 3 4 5 6 7 8 9. При выполнении цикла в одиннадцатый раз значение `x` становится равным 10. Условие `x<10` уже не является истинным, и выполнение цикла прекращается.

duplicateMovieClip (4+)

Синтаксис `duplicateMovieClip("источник", "копия_экземпляра", уровень)`

При помощи действия `duplicateMovieClip()` создается копия экземпляра клипа. Действие принимает три аргумента: `источник` — имя экземпляра, с которого следует создать копию (строка); `копия_экземпляра` — имя, которое присваивается копии экземпляра (тоже строка); `уровень` — уровень главного фильма, на который следует поместить полученную копию. Аргумент `уровень` используется не для того, чтобы присвоить новому клипу свойство фильма `_level`, а чтобы установить уровень копии по отношению к клипу `источник`. Если, например, клип `источник` находится на уровне 1, его копию можно поместить ниже на уровень 0 или выше на уровень 2. Клипы, находящиеся на уровнях, номер которых больше, будут перекрывать клипы с меньшим номером уровня, то есть появляться поверх них. Копии клипов наследуют свойства оригиналов, но не переменные из временной шкалы клипа источника. Чтобы удалить копию клипа, воспользуйтесь действием `removeMovieClip()`.

for (5+)

Синтаксис

```
for(инициализирующее_выражение;условное_выражение;модифицирующее_выражение){
    выражение;
}
```

Действие `for` предлагает альтернативный вариант цикла `while`. Вместо двух дополнительных строк, в которых инициализируется переменная и увеличивается счетчик цикла, в цикле `for` используется одна-единственная строка, в которую помещаются все элементы цикла.

При обнаружении в сценарии цикла `for` первым выполняется `инициализирующее_выражение`, в котором обычно устанавливается счетчик цикла. Это происходит только один раз перед запуском цикла. Затем анализируется `условное_выражение`,

содержащее условие прекращения цикла. Пока оно равно `true`, цикл не прекращается. Каждый раз после прохождения всех строк тела цикла выполняется модифицирующее_выражение, в котором происходит изменение счетчика цикла. Как только проверка условного_выражения даст результат `false`, все строки тела цикла и модифицирующее_выражение будут пропущены и управление будет передано первому выражению, следующему за телом цикла.

Если нужно, чтобы в цикле было несколько переменных (то есть более одного счетчика), разделите их запятыми.

Пример

Два нижеприведенных цикла эквивалентны. Каждый из них выполняется до тех пор, пока переменные не станут равны 5. Обратите внимание на то, насколько компактнее синтаксис цикла `for`.

```
for(hi=10,lo=10;hi!=lo;hi--,lo++){
    trace("hi: "+hi);
    trace("lo: "+lo);
}

hi=10;
lo=0;
while(hi!=lo){
    trace("hi: "+hi);
    trace("lo: "+lo);
    hi--;
    lo++;
}
```

for...in (5+)

Синтаксис `for(переменная in объект){`
 выражение;
`}`

Цикл `for...in` выполняет выражение в отношении ряда свойств, принадлежащих объекту или массиву. Цикл выполняет выражение для каждого свойства (сохраненного в переменной), относящегося к указанному объекту. Этот цикл можно использовать для того, чтобы установить или обновить все свойства объекта либо извлечь из них необходимую информацию.

Пример

```
for(item in _level0){
    clip=_root[item];
    if(typeof clip=="movieClip"){
        trace("Путь: "+targetPath(clip));
        //Действия для подготовки клипов и т. д.
    }
}
```

Данный сценарий проверяет все элементы временной шкалы объекта. Если элемент является клипом, путь к нему выводится в окне Output. Кроме того, сценарий можно использовать как шаблон для выполнения некоторых других действий с клипами, например для того, чтобы установить координаты клипа, сделать его видимым или невидимым и т. д.

fscommand (3+)

Синтаксис fscommand("аргумент", "значение")

Действие fscommand() можно использовать для установки связи с приложением, служащим хостом для вашего Flash-фильма. С его помощью обеспечивается взаимодействие между фильмом и автономным приложением Flash Player. Дополнительную информацию об использовании действия fscommand() в ваших проектах вы найдете в главе 29. Это действие также может обеспечивать связь между фильмом и Web-браузером при помощи команд JavaScript или VBScript.

function (5+)

Синтаксис function имя(аргумент1, аргумент2...аргументХХ
выражение;
}

При помощи действия function создается функция. Функция — это подпрограмма обработки данных, которую можно выполнять много раз. Функции применяются для создания стандартных процедур, выполняющих часто встречающиеся задачи. Чтобы создать функцию, необходимо указать ее имя и действия, которые она будет выполнять (выражение). В случае необходимости вы также можете указать аргументы для передачи функции каких-либо параметров.

Пример

Простая функция не требует аргументов. Следующая функция перемещает головку воспроизведения основной временной шкалы к первому кадру.

```
function rewind(){
    gotoAndStop(1);
}
```

Для более сложных функций необходимо указывать аргументы, такие функции более гибки и предоставляют больше возможностей разработчику.

Нижеприведенная функция сперва проверяет, указаны ли все необходимые аргументы. Если это не было сделано, в окне Output появляется соответствующее сообщение и выполнение функции прекращается. Если аргументы указаны правильно, функция возвращает результат.

```
function times (op1, op2){
    if(op1==undefined || op2==undefined){
        trace("Требуются аргументы");
        return;
    }else{
        return op1*op2;
    }
}
```

При втором вызове в вышеприведенном примере действие return возвращает значение. Чтобы функция возвратила результат, который можно использовать, присвойте ее переменной, как это сделано в следующем выражении:

```
prod=_root.times(2,2);
```

При выполнении функции из предыдущего примера переменная prod будет иметь значение 4.

getURL (2+/4+)

Синтаксис `getURL("имя_файла". "окно", "метод_передачи")`

Действие `getURL` позволяет Flash работать за пределами фильма. С его помощью можно:

- ▶ открыть документ, находящийся по указанному URL в сети Интернет;
- ▶ передать данные на сервер;
- ▶ установить связь с фильмом программы Director.

Действие `getURL()` принимает до трех аргументов. Аргумент `имя_файла` указывает файл, который должен быть открыт, или команду, которая должна быть передана. Аргумент `окно` позволяет обратиться к определенному окну в браузере или фрейму. Все значения этого аргумента указываются как строки.

- ▶ `_self` — документ загружается в текущем окне, где в настоящий момент воспроизводится фильм;
- ▶ `_blank` — документ загружается в новом окне;
- ▶ `_parent` — данное значение указывает на текущее окно и позволяет загрузить документ во фрейм, в котором в настоящий момент воспроизводится фильм;
- ▶ `_top` — данный элемент целесообразно использовать тогда, когда ваш фильм воспроизводится во фрейме, а новый документ, на который указывает URL, должен заполнить все окно браузера.

Третий аргумент `метод_передачи` (который поддерживают только четвертая и более поздние версии Flash Player) позволяет передавать информацию приложению на сервер при помощи методов GET и POST. Дополнительную информацию об этих методах можно найти в главе 17.

gotoAndPlay (2+)

Синтаксис `gotoAndPlay(номер_кадра)`
`gotoAndPlay("метка_кадра")`
`gotoAndPlay("имя_сцены",номер_кадра)`
`gotoAndPlay("имя_сцены","метка_кадра")`

Это действие позволяет переместиться к определенному кадру, сцене или метке кадра и возобновить воспроизведение фильма с этого места. Аргумент `номер_кадра` задается как целое число, а аргументы `метка_кадра` и `имя_сцены` — как строки (то есть в кавычках). Если аргумент `имя_сцены` не указан, действие выполнится по отношению к текущей сцене.

gotoAndStop (2+)

Синтаксис `gotoAndStop(номер_кадра)`
`gotoAndStop("метка_кадра")`
`.gotoAndStop("имя_сцены",номер_кадра)`
`gotoAndStop("имя_сцены","метка_кадра")`

Это действие позволяет переместиться к определенному кадру, сцене или метке кадра и остановить воспроизведение фильма в этом месте. Аргумент `номер_кадра` задается как целое число, а аргументы `метка_кадра` и `имя_сцены` — как строки (то есть в кавычках). Если аргумент `имя_сцены` не указан, действие выполнится по отношению к текущей сцене.

if (4+)

Синтаксис `if (у слов ие){
 выражение;
}`

Действие `if` используется для создания условного оператора. Условие является выражением, возвращающим значение типа `boolean`. Если условие истинно, выполняется выражение. В противном случае выражение игнорируется. Действие `if` можно использовать в сочетании с действием `else`, позволяющим указать выражения, которые будут выполняться, когда условие равно `false`.

loadMovie (3+)

Синтаксис `loadMovie("URL", "уровень", "метод_передачи")`

Действие `loadMovie()` применяется для загрузки дополнительных SWF-файлов. Его аргументы всегда указываются как строки (то есть в кавычках). Аргумент `URL` позволяет ввести полный либо относительный путь к файлу, который вы хотите загрузить. В аргументе `уровень` вы можете указать, куда следует загрузить фильм. Фильм можно загрузить как новый документ (`уровень_level1`, `_level4` и т. д.) или заменить им фильм на одном из уже существующих уровней. Например, фильм основной временной шкалы можно заменить, загрузив новый фильм на уровнях `_level0` или `_root`. В аргументе `уровень` также можно указать имя клипа, в который нужно вставить новый SWF-файл. При загрузке фильма можно также передавать ему переменные. В аргументе `метод_передачи` укажите метод `GET` или `POST`, чтобы передать переменные в новый фильм. Метод `GET` рекомендуется использовать при загрузке небольшого количества переменных, в противном случае лучше воспользоваться методом `POST`.

loadMovieNum (4+)

Синтаксис `loadMovieNum("URL",уровень,"метод_передачи")`

Действие `loadMovieNum()` почти идентично действию `loadMovie()`. Основное различие между ними состоит в том, что в первом случае аргумент `уровень`, определяющий уровень загрузки документа, задается как число, а не как строка. Это облегчает динамическую загрузку фильмов, но делает невозможной загрузку новых фильмов в экземпляры клипов. Если указанный уровень не существует, то программа создаст его для нового фильма. Если такой уровень все-таки есть, фильм, который находится на нем, будет заменен.

Пример Два следующих выражения эквивалентны.

```
loadMovieNum("act1.swf",1,"GET");
loadMovie("act1.swf","_level1","GET");
```

В следующем цикле перебираются все уровни с фильмами от 0 до 8 и осуществляется проверка, все ли они существуют. В переменной `top` сохраняется номер последнего уровня, на котором имеется какой-то фильм. По завершении цикла на следующий уровень загружается файл `high.swf`.

```
l=0;
top=0;
do{
    if(eval("_level"+l){
        trace("Найдено@"+l);
```

```

        top++;
    } else {
        trace("Не найдено@"+1);
    }
} while(1++<8);
trace("Верхнийуровень"+(top-1));
loadMovieNum("high.swf", top);

```

Если бы во время выполнения сценария на уровнях 0 и 1 были фильмы, файл high.swf загрузился бы на уровень `_level2`.

loadVariables (4+)

Синтаксис `loadVariables("URL", "уровень", "метод_передачи")`

Действие `loadVariables()` позволяет загрузить переменные из файла, указанного в аргументе URL, в фильм или клип, определенный в аргументе уровень (следует указать либо уровень клипа, например `_level2`, либо его имя). Flash может считывать различные данные из текстовых документов, файлов CGI, ASP, PHP и других серверных файлов. Все переменные передаются как строки и должны находиться в том же домене, что и фильм, который запрашивает их при вызове его в Web-браузере. В аргументе `метод_передачи` укажите метод GET или POST, чтобы передать переменные в фильм-получатель. Метод GET рекомендуется использовать при загрузке небольшого количества переменных, в противном случае лучше воспользоваться методом POST.

loadVariablesNum (4+)

Синтаксис `loadVariablesNum("URL", уровень, "метод_передачи")`

Действия `loadVariables()` и `loadVariablesNum()` практически идентичны. Основное различие между ними состоит в том, что для действия `loadVariablesNum()` аргумент уровень, определяющий уровень загрузки документа, задается как число, а не как строковое значение. Это облегчает динамическую загрузку фильмов, принимающих переменные. Однако действие `loadVariablesNum()` не позволяет передавать переменные в клипы.

nextFrame (2+)

Синтаксис `nextFrame()`

С помощью этого действия головка воспроизведения перемещается на один кадр вперед и останавливается в новой позиции. Это действие является альтернативой использованию выражения `gotoAndStop(_currentframe+1)`.

nextScene (2+)

Синтаксис `nextScene()`

Действие `nextScene()` полезно при выполнении перемещений между сценами фильма. При вызове этого действия головка воспроизведения переходит к первому кадру следующей сцены фильма и воспроизведение фильма останавливается. Порядок сцен определяется расположением элементов на панели Scene. Действие `nextScene()` не будет работать, если его связать с клипом; его следует связывать с кадрами главной временной шкалы фильма.

play (2+)

Синтаксис play()

Действие play() выполняет одну из наиболее важных функций: с его помощью воспроизводится анимация. Воспроизведение продолжается до тех пор, пока его не остановит другое действие или пока не закончатся кадры на временной шкале. Перейдя к новой сцене посредством действия nextScene() или prevScene(), вы можете использовать действие playO, чтобы начать воспроизведение этой сцены.

prevFrame (2+)

Синтаксис prevFrame()

С помощью данного действия головка воспроизведения перемещается на один кадр назад и останавливается. Это действие является альтернативой использованию выражения gotoAndStop(_currentframe-1).

prevScene (2+)

Синтаксис prevSceneO

При вызове данного действия головка воспроизведения переходит к первому кадру предыдущей сцены фильма и воспроизведение фильма на этом месте останавливается. Порядок сцен определяется расположением элементов на панели Scene. Действие prevScene() не будет работать, если его связать с кликом; его следует связывать с кадрами фильма главной временной шкалы.

print (5+)

Синтаксис print("имя_клипа", "печатаемая_область")

Это действие применяется для вывода на печать всего фильма или его части. В аргументе имя_клипа вы указываете имя клипа или фильма, кадры которого хотите напечатать. Чтобы определить, какие именно кадры должны быть напечатаны, пометьте каждый такой кадр меткой #p на панели Properties. Если этого не сделать, будут напечатаны все кадры клипа или фильма, указанного в аргументе имя_клипа. Принтеры PostScript способны работать как с векторными, так и с растровыми изображениями. Все остальные принтеры перед печатью преобразуют векторные изображения в растровые. Действие printO не позволяет напечатать элементы, отображаемые с использованием альфа-канала, и отобразить цветовые эффекты — для этого воспользуйтесь действием printAsBitmap().

Аргумент печатаемая_область может принимать следующие строковые значения.

- ▶ bmovie. Flash использует содержимое помеченного кадра для определения печатаемой области. Чтобы пометить кадр, определяющий печатаемую область, укажите для него метку #b и поместите в него графические элементы, которые будут служить в качестве обрамляющей фигуры, границы которой соответствуют печатаемой области.
- ▶ bmax. Применяет комбинацию содержимого всех печатаемых кадров, чтобы определить печатаемую область. Этим значением можно воспользоваться, если данные, выводимые на печать, создаются динамически и отличаются от кадра к кадру.
- ▶ bframe. Обработывается каждый печатаемый кадр отдельно. Изображение будет заполнять всю страницу, поэтому масштаб элементов каждого кадра будет соответственно изменяться.

Пример

В этом сценарии на печать выводятся все кадры экземпляра клипа `directions`. Печатаемые области в них определяются кадром с меткой `#b`.

```
print("_root.directions", "bmoive");
```

Если вы хотите напечатать отдельные кадры, укажите для них метку `#p`.

printAsBitmap (5+)

Синтаксис `printAsBitmap("имя_клипа", "печатаемая_область")`

Это действие подобно действию `printO`, но за одним исключением. С его помощью вместо векторных печатаются растровые изображения. Несмотря на невысокое качество печати, именно это действие следует использовать, если вы хотите сохранить при печати эффекты прозрачности (напечатать элементы, для которых указаны значения альфа-канала) и цветовые эффекты. Характеристику аргументов действия см. в описании действия `print()`.

printAsBitmapNum (5+)

Синтаксис `printAsBitmapNum(уровень, "печатаемая_область")`

Данное действие практически идентично действию `printAsBitmap()`, только в аргументе `уровень` вместо строкового значения вы указываете номер уровня. Это действие используется, если необходимо вывести на печать фильм, находящийся на некотором уровне. Вместо того чтобы указывать номер уровня как целое число, вы можете воспользоваться переменной, принимающей числовые значения.

Пример

В этом примере номер уровня задается как переменная `printNfo`, содержащая числовые данные, возвращаемые функцией `output()`.

```
printNfo=output();
printAsBitmapNum(printNfo, "bframe");
```

printNum (5+)

Синтаксис `printNum(уровень, "печатаемая_область")`

Это действие практически идентично действию `print()`, только в аргументе `уровень` вместо строкового значения вы указываете номер уровня. Данное действие используется, если необходимо вывести на печать фильм, находящийся на некотором уровне. Вместо того чтобы указывать номер уровня как целое число, вы можете воспользоваться переменной, принимающей числовые значения.

Пример Два следующих выражения эквивалентны.

```
printNum(3, "bmax");
print("_level3", "bmax");
```

removeMovieClip (4+)

Синтаксис `removeMovieClipC "имя_клипа")`

Действие `removeMovieClip()` используется для удаления экземпляра клипа из Flash-фильма. В аргументе `имя_клипа` укажите имя экземпляра клипа, который вы хотите удалить, или путь к нему. Это действие выполняется только для клипов, которые вы поместили в фильм при помощи действия `attachMovie()` или `duplicateMovieClip()`.

Пример

Если этот сценарий связать с клипом, созданным посредством действия `duplicateMovieClip()`, то экземпляр этого клипа при щелчке на нем мышью будет удален. Поскольку `this` относится к текущему клипу, его имя указано *не* как строковое значение.

```
onClipEvent(mouseUp){
    if(this.hitTest(_root._xmouse,_root._ymouse,true){
        removeMovieClip(this);
    }
}
```

setProperty (4+)

Синтаксис `setProperty("имя_клипа",свойство,значение)`

Действие `setProperty()` применяется для того, чтобы присвоить значение одного из predeterminedных свойств Flash экземпляру клипа. Укажите нужный экземпляр в аргументе `имя_клипа`. В аргументе `свойство` задайте соответствующее свойство (например, `_alpha` или `_rotation`), а в аргументе `значение` — значение свойства.

startDrag (4+)

Синтаксис `startDrag("имя_клипа",блокирование,влево,вверх,вправо,вниз)`

Действие `startDrag()` делает клип, указанный в аргументе `имя_клипа`, *перемещаемым*. То есть этот экземпляр клипа можно перемещать на области действия, щелкнув на нем мышью и перетаскивая его. Если аргумент `блокирование` имеет значение `true`, указатель блокируется на точке регистрации перемещаемого клипа. Если аргумент имеет значение `false`, можно перемещать клип, предварительно поместив указатель в любую точку на нем. Это необязательный аргумент. Если вы его не укажете, по умолчанию будет использоваться значение `false`. Аргументы `влево`, `вверх`, `вправо`, `вниз` определяют координаты границ перемещения клипа. Эти координаты определяются относительно временной шкалы, на которой размещается клип, указанный в аргументе `имя_клипа`. Координаты определяются относительно начала (0;0) временной шкалы, на которой размещается клип. Если клип относится к основной временной шкале, началом будет левый верхний угол области действия. Если клип находится на экземпляре другого клипа, началом будет служить точка регистрации этого другого клипа. Аргументы `влево`, `вверх`, `вправо`, `вниз` тоже необязательные. Чтобы отменить возможность перемещения клипа, воспользуйтесь действием `stopDrag()`.

Пример См. описание действия `stopDrag()`.

stop (2+)

Синтаксис `stop()`
`имя_клипа.stop()`

При вызове действия `stop()` воспроизведение фильма останавливается. Чтобы остановить воспроизведение указанного клипа, введите ссылку на него в аргументе `имя_клипа`.

stopDrag (4+)*Синтаксис* stopDrag()

Действие stopDrag() прекращает выполнение действия startDrag() в фильме. Поскольку во Flash может быть только один перемещаемый клип, этому действию не требуются аргументы. Оно влияет на текущий клип, с которым связано действие startDrag().

Пример

В следующем сценарии экземпляр клипа blob делается перемещаемым. Клип остается перемещаемым до тех пор, пока пользователь не отпустит кнопку мыши, то есть не закончит перемещение клипа.

```
blob.onMouseDown=function(){
    startDrag("blob",false,0,0,550,400);
}

blob.onMouseUp=function(){
    stopDrag();
}
```

switch (6)*Синтаксис*

```
switch(условие){
case(a):
    //выражения, выполняемые, если условие=a
    break;
case(b):
    //выражения, выполняемые, если условие=b
    break;
case(n):
    //выражения, выполняемые, если условие=n
    break;
default:
    //выражения, выполняемые по умолчанию, если проверка условия возвращает
    false
}
```

Действие switch представляет собой условную конструкцию, которая является альтернативой конструкции if...else. Это действие применяется, когда существует несколько вариантов истинности условия, каждый из которых имеет ряд выражений, выполняемых последовательно. Аргумент условие — это проверяемое условное выражение. В каждую ветвь case поместите выражения, которые будут выполняться при истинности условия. Кроме того, в сценарий следует включить действия break, которые не позволят перейти к следующей ветви, если проверка условия возвращает true. Выражения ветви default будут выполняться, если ни в одном из предыдущих случаев проверка условия не возвратила true.

Пример

В этом примере в клипе отслеживается текущее положение головки воспроизведения на основной временной шкале. В зависимости от номера текущего кадра основной временной шкалы клип должен задержаться в кадрах 1, 2 или 3 своей временной шкалы.

```
onClipEvent(enterFrame){
    switch(_root.currentframe){
    cased):
```

```

        gotoAndStop(1);
        break;
    case(10) :
        gotoAndStop(2);
        break;
    case(20);
        gotoAndStop(3);
        break;
    default:
        stop();
}
}

```

unloadMovie (3+)

Синтаксис `unloadMovie("имя_клипа")`

Действие `unloadMovie()` применяется для удаления уровней, содержащих фильмы, которые загружены при помощи действий `loadMovie()` и `loadMovieNum()`. Действие `unloadMovie()` также может помочь удалить содержимое экземпляра клипа из области действия. В аргументе `имя_клипа` (указывается как строка) задается фильм или экземпляр клипа, который вы хотите удалить.

Пример

В этом выражении удаляется фильм на уровне 2:

```
unloadMovie("_level2");
```

В следующем выражении удаляется содержимое экземпляра клипа:

```
unloadMovie("feature1");
```

При удалении содержимого экземпляра клипа сам экземпляр остается в области действия в виде пустого кадра. Этот кадр можно заполнить новым фильмом (SWF) при помощи действия `loadMovie()`.

unloadMovieNum(4+)

Синтаксис `unloadMovieNum(уровень)`

Действие `unloadMovieNum()` почти идентично действию `unloadMovie()`. Основное различие между ними состоит в том, что для действия `unloadMovieNum()` уровень загрузки документа определяется как число, а не как строка. Из-за этого данное действие нельзя применять для удаления содержимого экземпляра клипа. Оно применяется для удаления уровней вместе с их фильмами.

Пример Два следующих выражения эквивалентны.

```

unloadMovieNum(2);
unloadMovie("_level2");

```

updateAfterEvent (5+)

Синтаксис `updateAfterEvent()`

Действие `updateAfterEvent()` обновляет экран тотчас после совершения некоторого события. Не будучи связанным с частотой смены кадров фильма, данное действие наделяет программу способностью реагировать на нажатие клавиш клавиатуры и действия мыши, происходящие *между* кадрами. Действие можно использовать для ввода данных с клавиатуры или других событий, связанных с вводом пользователем данных.

Пример

Приведенная ниже функция, связанная с экземпляром `cursor`, превращает этот экземпляр в пользовательский указатель мыши.

```
cursor.onMouseMove=function(){
    cursor._x=_xmouse;
    cursor._y=_ymouse;
    updateAfterEvent();
}
```

while (4+)

Синтаксис `while(условие){`
 выражение;
 `}`

При помощи действия `while()` создаются циклы. В цикле `while()` необходимо указать условие, которое будет проверяться. Если проверка условия возвращает `true`, будет выполняться выражение, которое содержится в теле цикла. Если проверка условия возвращает значение `false`, выражение, содержащееся в теле цикла, пропускается. В цикле `while()` условие всегда проверяется перед выполнением любых действий.

Пример

Сперва принято указывать переменную(-ые) цикла, а уже затем условие и само тело цикла. С помощью приведенного ниже цикла создаются копии клипа и произвольным образом помещаются на область действия.

```
var z:
z=0;
while(z<10){
    ry=Math.random()*400;
    duplicateMovieClip("lite", "lite"+z z);
    _root ["lite"+z]._x=50*z;
    _root["lite"+z]._y=ry;
    z++;
}
```

with (5+)

Синтаксис `with(объект){`
 выражение;
 `}`

Появившись во Flash 5, действие `withO` заменило действие `tellTarget O`, применявшееся во Flash 3 и 4. Это действие используется для получения доступа и управления свойствами объекта или экземпляра клипа, указанного в аргументе объект. Если вы хотите получить доступ к свойствам экземпляров клипа, действие `with()` позволяет избежать указания длинных путей к объекту в ссылках.

Пример

Этот сценарий устанавливает новое значение прозрачности (значение альфа-канала) и новые размеры для экземпляра клипа `menu`.

```
with(menu){
    _alpha-40;
    _xscale+=25;
    _yscale+=25;
}
```

А вот идентичный сценарий без использования действия `with()`.

```
_root.navigation.bar.menu._alpha=40;
_root.navigation.bar.menu._xscale+=25;
_root.navigation.bar.menu._yscale+=25;
```

Свойства

В ActionScript *свойства* — это элементы, при помощи которых вы получаете доступ к данным о состоянии, положении, характеристиках и т. д. различных элементов фильма. Свойства можно ассоциировать практически со всеми компонентами фильма. Когда вы указываете значения свойств для отдельных элементов, каждый из них получает свой уникальный набор характеристик.

MX Во Flash MX управление кнопками происходит с помощью нового объекта `Button`. С помощью объекта `Button` можно управлять любым символом кнопки, имеющим имя экземпляра, изменяя его положение, размеры и т. п.

alpha (4+)

Синтаксис *экземпляр._alpha=значение*

При помощи свойства `_alpha` устанавливаются значения прозрачности для экземпляров объектов `Button` и клипов. Это свойство может иметь значения от 100 (объект непрозрачен) до 0 (объект прозрачен). Прозрачные кнопки и клипы остаются активными в области действия и действуют как обычные объекты.

currentFrame (4+)

Синтаксис *экземпляр_клипа._currentFrame*

Это свойство возвращает номер кадра клипа, который воспроизводится на момент вызова свойства. Если не указать значение аргумента `экземпляр_клипа`, свойство вернет номер текущего кадра фильма, с которым связан этот сценарий.

dropTarget(4+)

Синтаксис *перемещаемый_экземпляр._dropTarget*

Это свойство доступно только для чтения. Оно возвращает путь к клипу, в который был помещен другой клип, указанный в аргументе `перемещаемый_экземпляр`. Клип считается помещенным в другой, если точка регистрации перемещаемого клипа находится в границах главного клипа. Свойство возвращает путь, записанный с использованием символов (/). Если вы хотите, чтобы для записи пути использовался точечный синтаксис, воспользуйтесь функцией `eval()`. Например:

```
onClipEvent(mouseUp){
    trace("Использование символа /: "+this._dropTarget);
    trace("Точечный синтаксис: "+eval(this._dropTarget));
}
```

Пример

Когда вы «отпускаете» клип, который перетаскивали мышью, следует проверка, находится ли он в экземпляре `hit`. Если нет, то перемещаемый клип (его точка регистрации) помещается в точку с координатами (100;100).

```
onClipEvent(mouseUp){
    stopDrag();
    if(eval(this.dropTarget)!=_root.hit){
```

```

        this._x=100;
        this._y=100;
    }
}

```

_framesloaded (4+)

Синтаксис `_framesloaded`
 имя_файла.`_framesloaded`

Это свойство возвращает количество кадров, которые были загружены из основной временной шкалы фильма или внешнего SWF-файла. С его помощью перед началом воспроизведения можно проверить, все ли кадры фильма загружены.

Пример

```

if(_framesloaded==totalframes){
    gotoAndPlay("start");
}else{
    gotoAndPlay(1);
}

```

В приведенном сценарии контролируется процесс загрузки кадров фильма основной временной шкалы. Если все кадры загружены, головка воспроизведения переходит к кадру с меткой `start`; в противном случае — к кадру 1.

_height (4+)

Синтаксис `экземпляр._height=значение`

Это свойство применяется, чтобы проверить или установить высоту отображения экземпляра клипа или кнопки. Аргумент значение данного свойства следует указывать в пикселах.

_name (4+)

Синтаксис `экземпляр._name="имя"`;

С помощью этого свойства указываются имена экземпляров клипов и кнопок. Вы можете проверить, имеет ли экземпляр имя и изменить его в случае необходимости. Чтобы присвоить экземпляру новое имя, введите соответствующее строковое значение в аргументе `имя`.

NaN (5+)

Синтаксис `NaN`

`NaN` означает «not a number» («не номер»). Это свойство применяется для указания элементов, которые не являются числами или которые нельзя обрабатывать как числовые данные.

_quality (5+)

Синтаксис `_quality="значение"`

Это свойство устанавливает параметры качества изображения для вашего фильма. Качество изображения определяется сглаживанием, которое применяется в фильме. Аргумент значение принимает одно из четырех возможных строковых значений.

- ▶ **LOW**: отменяются любые функции сглаживания и устанавливается приоритет скорости воспроизведения над качеством.
- ▶ **MEDIUM**: сглаживание применяется только по отношению к векторным изображениям.

- ▶ "HIGH": сглаживание применяется к векторным и к растровым изображениям, если в последних нет анимации.
- ▶ "BEST": в этом случае фильм воспроизводится с наилучшим качеством, которое только достижимо, без учета возможности компьютера поддерживать требуемую скорость воспроизведения.

_rotation (4+)

Синтаксис *экземпляр._rotation=значение*

С помощью свойства `_rotation` осуществляется поворот экземпляра кнопки или клипа на указанное в аргументе значение число градусов. Если аргумент значение является положительным числом, экземпляр поворачивается по часовой стрелке; если отрицательным — против. Вращение осуществляется относительно исходной ориентации кнопок и клипов на их временной шкале.

_soundbuftime (4+)

Синтаксис *_soundbuftime=значение*

Свойство `_soundbuftime` определяет, сколько секунд потокового звука следует загрузить перед началом его воспроизведения. Значение по умолчанию составляет 5 секунд.

_target(4+)

Синтаксис *экземпляр._target*

Это свойство возвращает путь к экземпляру кнопки или клипа. Для отображения пути используется синтаксис с применением символа (/), как во Flash 4. Для преобразования его синтаксиса в синтаксис с использованием точек (мы называем его точечный синтаксис) применяйте функцию `targetPath()`.

_totalframes (4+)

Синтаксис *экземпляр_клипа._totalframes*

Свойство `_totalframes` возвращает количество кадров в клипе или в фильме основной временной шкалы. Это свойство применяется при создании анимационных изображений, которые воспроизводятся, пока идет процесс загрузки всех кадров фильма.

Пример См. в описании свойства `_framesloaded`.

_url(4+)

Синтаксис *экземпляр._url*

Свойство `_url` возвращает URL для SWF-файла, содержащего экземпляр кнопки или клипа, указанный в аргументе экземпляр. Возвращаемое свойством строковое значение содержит полный путь к требуемому файлу.

Пример `trace(this._url);`

Если файл загружается с жесткого диска, будет выведен следующий результат:

```
file:///C:/windows/Desktop/pathTest.swf
```

Для файла, загружаемого с Web-узла, результат будет таким:

```
http://www.domain.com/folder/file.swf
```


_visible (4+)

Синтаксис *экземпляр._visible=Boolean*

Данное свойство определяет, виден ли экземпляр кнопки или клипа, указанный в аргументе экземпляр, в области действий. Если это свойство имеет значение true (используется по умолчанию), экземпляр является видимым. Если свойство имеет значение false, экземпляр является скрытым. Скрытые (невидимые) экземпляры *клипов* реагируют на события и действия, а экземпляры *кнопок*, которые были сделаны невидимыми при помощи свойства `_visible`, — нет.

_width (4+)

Синтаксис *экземпляр._width=значение*

Это свойство применяется, чтобы установить ширину экземпляра клипа или кнопки, которые указаны в аргументе экземпляр. Аргумент значение следует указывать в пикселах.

_x (3+)

Синтаксис *экземпляр._x*
экземпляр._x=значение

С помощью свойства `_x` возвращается (в первом случае) или устанавливается (во втором случае) значение координаты положения экземпляра клипа или кнопки, па которые указывает аргумент экземпляр, по оси X. Координата определяется для точки регистрации и, следовательно, зависит от того, где находится экземпляр (см. ниже).

- ▶ Если экземпляр находится в фильме основной временной шкалы, координата определяется относительно левого края области действия. Свойство `_x` клипа или кнопки, удаленного па 225 пикселей от левого края области действия, будет иметь значение 225.
- ▶ Если экземпляр находится в клипе, его координаты определяются относительно точки регистрации этого клипа. Например, если он находится па 50 пикселей правее точки регистрации, его свойство `_x` будет иметь значение 50; если па 50 пикселей левее — -50.

_xmouse (5+)

Синтаксис *экземпляр._xmouse*

Это свойство возвращает координату положения указателя мыши по оси X относительно экземпляра клипа или кнопки.

- ▶ `_root._xmouse` возвращает координату положения указателя мыши относительно левого края области действий.
- ▶ `moverClip._xmouse` возвращает координату положения указателя мыши относительно точки регистрации объекта `moverClip`.

Пример

Приведенный сценарий позиционирует экземпляр клипа па одной горизонтальной линии с указателем мыши относительно основной временной шкалы.

```
onClipEvent(enterFrame){
    _x=_root._xmouse;
}
```

_xscale(4+)

Синтаксис *экземпляр._xscale=значение*

Это свойство применяется для горизонтального масштабирования экземпляра клипа или кнопки. Все значения масштабирования основываются на начальном размере экземпляра и вычисляются в процентном отношении к нему. Изображение выравнивается симметрично относительно точки регистрации экземпляра.

Пример

В приведенном сценарии при совершении события `enterFrame` экземпляр клипа масштабируется, то есть его ширина увеличивается на 50%. Поскольку масштабирование осуществляется симметрично, по 25% приходится на увеличение ширины влево и вправо.

```
onClipEvent(enterFrame){
    _xscale=_xscale+50;
}
```

_y(3+)

Синтаксис *экземпляр._y*
 экземпляр._y=значение

С помощью свойства `_y` возвращается (в первом случае) или устанавливается (во втором случае) значение координаты клипа или кнопки, которое указано в аргументе `экземпляр`, по оси Y. Значение координаты относится к точке регистрации и зависит от того, где находится экземпляр (см. ниже).

- ▶ Если экземпляр находится в фильме основной временной шкалы, координата определяется относительно верхнего края области действия. Свойство `_y` экземпляра клипа или кнопки, находящегося на 100 пикселей ниже верхнего края области действия, будет иметь значение 100.
- ▶ Если экземпляр находится в клипе, его координаты определяются относительно точки регистрации этого клипа. Например, если экземпляр находится на 50 пикселей ниже точки регистрации, его свойство `_y` будет иметь значение 50; если на 50 пикселей выше - -50.

_ymouse (5+)

Синтаксис *экземпляр._ymouse*

Это свойство возвращает координату положения указателя мыши по оси Y относительно экземпляра клипа или кнопки:

- ▶ `_root._ymouse` возвращает координату положения указателя мыши относительно верхнего края области действия;
- ▶ `launcher._ymouse` возвращает координату положения указателя мыши относительно точки регистрации кнопки `launcher`.

_yscale (4+)

Синтаксис *экземпляр._yscale=значение*

Это свойство применяется для вертикального масштабирования экземпляра клипа или кнопки. Все значения масштабирования основываются на начальном размере экземпляра и вычисляются в процентном отношении к нему. Изображение выравнивается симметрично относительно точки регистрации экземпляра.

Функции

Функции — самостоятельные фрагменты кода в языке ActionScript. Эти уникальные подпрограммы позволяют обрабатывать передаваемые им данные. Функции ActionScript можно использовать для выполнения стандартных задач по обработке и извлечению информации во Flash-фильмах.

eval (5+)

Синтаксис eval (выражение)

Когда выражение содержит или возвращает сроку, функция eval () преобразует его в идентификатор, содержащий значение переменной, свойства или ссылку на клип, содержащуюся в строке. Если выражение не содержит данных, функция возвращает undefined.

Пример

Функцию eval () можно использовать для обработки переменных или экземпляров, создаваемых динамически.

```
J-1:
while(j<=5){
    this.duplicateMovieClip("cloud"+j,j);
    eval("_root.cloud"+j)._x=100*j;
    eval("_root.cloud"+j)._y=100*j;
    eval("_root.cloud"+j)._alpha=20*j;
    j++;
}
```

В этом примере создается пять копий клипа. Каждый новый экземпляр (cloud1, cloud2 и т. д.) обрабатывается как строка и преобразуется в ссылку на клип, которой можно манипулировать.

getProperty (4+)

Синтаксис getProperty(экземпляр, свойство)

Эта функция применяется, чтобы извлечь свойство указанного экземпляра клипа. Аргумент экземпляра должен содержать ссылку на экземпляр клипа.

getTimer (4+)

Синтаксис getTimer()

Функция возвращает количество миллисекунд, прошедших с начала воспроизведения фильма. Она может быть полезна при использовании с различными действиями или событиями, вызываемыми через заданный промежуток времени.

Пример

```
onClipEvent(load){
    start=getTimerO;
}
onClipEvent(enterFrame){
    if(start+10000<getTimer()){
        orb.gotoAndStop(orb._currentframe+1);
        start=getTimerO;
    }
}
```

Приведенный сценарий представляет собой хороший пример простого таймера. В значении переменной start записывается количество миллисекунд, прошедших с момента первой загрузки клипа. В каждом кадре количество прошедших

миллисекунд сравнивается со значением переменной `start`, увеличенным на 10000. Если текущее количество миллисекунд больше `start+10000` (то есть фильм идет уже 10 секунд), головка воспроизведения клипа `orb` переходит к следующему кадру и останавливается, а значение счетчика сбрасывается. Через 10 секунд головка воспроизведения клипа `orb` переходит к следующему кадру, счетчик вновь переустанавливается и т. д.

newline (4+)

Синтаксис `newline`

При помощи функции `newline` в строку или текст вставляется символ разрыва строки (возврата каретки). Она используется в текстовых полях.

String (5+)

Синтаксис `String(выражение)`

Функция `String()` применяется для преобразования выражения в строковое значение. Значения типа `Boolean` возвращаются как `Boolean`. Например, `String(true)`; возвращает значение `true`.

Числа возвращаются в виде строковых значений. Например, `String(365)`; возвращает значение `365`.

Для клипа функция возвращает строковое значение, указывающее полный путь к клипу. Например, `String(this)`; возвращает значение `_root.main.sub1`.

Для объектов функция `String(this)` возвращает строковое значение (то же, что и метод `toString()`), которое зависит от содержимого объекта.

targetPath (5+)

Синтаксис `targetPath(экземпляр_клипа)`

Эта функция возвращает полный путь к экземпляру_клипа как строковое значение, записанное с использованием точечного синтаксиса.

Пример

Приведенный ниже код присваивает переменной `te` значение полного пути к клипу. Эту переменную можно использовать в последующем, когда необходимо сделать ссылку на путь к клипу.

```
onClipEvent(load){
    me=targetPath(this);
}
```

Операторы

Операторы применяются для вычисления и изменения значений выражений.

--(декрементирование) (4+)

Синтаксис `--A`
`A--`

Оператор `--` можно использовать как перед некоторым выражением (в данном случае `A`), так и после него. В первом случае из выражения `A` вычитается 1 и возвращается новое значение; во втором — тоже вычитается 1, но возвращается исходное значение выражения `A`.

Пример X=2;
 y=-x;

Оператор декрементирования помещен перед выражением, в результате получим $y = 1$, $x = 1$.

 X=2;
 y=x--;

Оператор декрементирования помещен после выражения, и в результате получим $y = 2$, $x = 1$.

++(инкрементирование)

Синтаксис ++A
 A++

Этот оператор можно использовать как перед выражением (в данном случае A), так и после него. В первом случае к выражению A прибавляется 1 и возвращается новое значение; во втором прибавляется 1, но возвращается исходное значение выражения A.

Пример X=1;
 y=++x;

Оператор инкрементирования помещен перед выражением, в результате получим $y = 2$, $x = 2$.

 X=1;
 y=x++;

Оператор инкрементирования помещен после выражения, в результате получим $y = 1$, $x = 2$.

!(логическое НЕ)

Синтаксис !A

Оператор ! всегда возвращает значение типа Boolean, противоположное значению следующего за ним выражения A.

Пример !1 возвращает 0.
 !0 возвращает 1.

!=(неравенство) (5+)

Синтаксис A!=B

Оператор != проверяет, действительно ли два выражения A и B не равны, и возвращает соответствующее логическое значение. Если выражения не равны, оператор возвращает true, в противном случае — false.

Пример

Оператор != можно использовать в сценарии анимации, которая отображает процесс загрузки. Если загрузились не все кадры фильма, головка воспроизведения переходит к кадру 1.

```
if(_framesloaded!=_totalframes){
    gotoAndPlay(1);
}else{
    gotoAndPlay(5);
}
```

!== (строгое неравенство) (6)*Синтаксис* A!==B

Этот оператор, также как и оператор !=, проверяет, действительно ли два выражения A и B не равны, и возвращает соответствующее логическое значение. Единственно отличие состоит в том, что оператор !== учитывает также тип данных выражений.

Пример У нас есть следующие выражения:

```
str="1";
num=new Number(1);
```

В первом случае возвращается значение false, поскольку значения переменных str и num равны:

```
str!=num;
```

Во втором случае возвращается значение true, поскольку переменные str и num не удовлетворяют строгому равенству — они относятся к различным типам данных (строка и число):

```
str!==num;
```

% (деление по модулю) (4+)*Синтаксис* A%B

Этот оператор возвращает остаток, полученный при делении выражения A на выражение B.

Пример 17%4 возвращает 1.

Если 17 разделить на 4, получим 4 и в остатке 1. Оператор % возвращает остаток деления двух выражений.

%= (присвоение по модулю) (4+)*Синтаксис* A%=B

Этот оператор возвращает остаток, полученный при делении выражения A на выражение B, и присваивает его выражению A.

Пример

```
x=2;
y=13;
y%=x;
trace(y);
```

В этом примере в окне Output будет выведен результат 1. Переменная y делится на переменную x и остаток (1) присваивается переменной y.

&& (логическое И) (4+)*Синтаксис* A&&B

Этот оператор оценивает два выражения A и B и возвращает логическое значение true или false в зависимости от значения каждого выражения.

true&&false возвращает false.

true&&true возвращает true.

false&&false возвращает false.

Чтобы оператор && возвратил true, оба выражения должны быть равны true.

- (вычитание) (4+)*Синтаксис* A-B

Этот оператор либо вычитает выражение B из выражения A, либо присваивает выражению противоположное значение.

Пример 1,5-0.4 возвращает 1,1.
 4-3 возвращает 1.
 -(4-3) возвращает -1.
 -(1-2) возвращает 1.

*** (умножение) (4+)***Синтаксис* A*B

Этот оператор умножает выражение A на выражение B.

= (присвоение результата умножения) (4+)Синтаксис* A*=B

Этот оператор умножает выражение A на выражение B и присваивает произведение выражению A.

Пример Запись a*=b эквивалентна a=a*b.

В следующем коде переменной a присваивается новое значение 200.

```
a=10
b=20
a*=b
```

?: (условный) (4+)*Синтаксис* условие?выражение1:выражение2

Условный оператор ?: является альтернативой конструкции if...else и требует всего одну строку кода. Оператор производит оценку условия и, если оно истинно, возвращает выражение1, в противном случае возвращает выражение2.

Пример

```
if(Key.getCode()==Key.RIGHT){
    nextFrame();
}else{
    stop();
}
```

В этом сценарии головка воспроизведения переходит на следующий кадр, если нажата правая клавиша мыши. Однако его можно записать и в следующем виде:

```
Key.RIGHT?nextFrame():stop();
```

// (однострочный комментарий) (1+)*Синтаксис* //комментарий

Если в коде сценария присутствует оператор //, это значит, что текст между символом оператора и символом конца строки, на которой находится оператор, следует игнорировать. В комментарии вы можете внести свои замечания и пояснения к сценарию. По умолчанию текст комментариев окрашивается в светло-серый цвет.

/*...*/ (многострочный комментарий) (5+)*Синтаксис* `/*начало_комментария...конец_комментария*/`

Этот оператор используется, если вы помещаете в сценарий большое количество текста, например блок из нескольких строк, которые должны игнорироваться. Программа будет пропускать весь текст между первой и последней звездочками. Не пропустите один из этих операторов при включении в код комментариев.

/= (присвоение результата деления) (4+)*Синтаксис* `A/-B`

Этот оператор возвращает значение, полученное в результате деления выражения A на выражение B, и присваивает его выражению A.

Пример Запись `x/=u` эквивалентна `x=x/u`.

Переменной x присваивается значение частного, полученного при выполнении операции `x/u`.

|| (логическое ИЛИ) (4+)*Синтаксис* `A||B`

Этот оператор проверяет выражение A и выражение B и возвращает значение типа `Boolean`. Если одно из выражений является истинным, оператор возвращает `true`. Если оба выражения ложны, возвращается значение `false`.

Для `1||0` оператор возвращает значение `true`.Для `0||1` оператор возвращает значение `true`.Для `1||1` оператор возвращает значение `true`.Для `0||0` оператор возвращает значение `false`.**+ (сложение) (4+)***Синтаксис* `A+B``"строка1"+"строка2"`

Оператор возвращает сумму выражений A и B. Его также можно использовать для объединения строк (только во Flash Player 5 и более поздних версиях).

+= (присвоение результата сложения) (4+)*Синтаксис* `A+=B``C+="строка"`

Этот оператор возвращает сумму выражений A и B и присваивает ее значение выражению A. Данный оператор работает с числами и строками.

Выражение `x+=1` эквивалентно `x=x+1`.`var x=10. y=20``x+=y` возвращает 30.

Оператор может также соединить строку с переменной и создать тем самым новое значение:

```
x="Action";
trace(x+="Script");
```

В приведенном примере в окне Output будет выведено слово `ActionScript`.

< (меньше) (4+)*Синтаксис* A<B

Если выражение A меньше выражения B, оператор < возвращает true. В противном случае (то есть если A больше или равно B) он возвращает false. Этот оператор может также обрабатывать строки. Прописные буквы, расположенные ближе к A, имеют меньшие значения, наибольшие значения имеют строчные буквы, расположенные ближе к z.

<= (меньше или равно) (4+)*Синтаксис* A<=B

Если выражение A меньше или равно выражению B, оператор <= возвращает true. В противном случае (то есть если A больше B) он возвращает false. Оператор может также обрабатывать строки. Прописные буквы, расположенные ближе к A, имеют меньшие значения, наибольшие значения имеют строчные буквы, расположенные ближе к z.

= (присвоение) (4+)*Синтаксис* идентификатор=A

Этот оператор присваивает значение выражения A некоторому идентификатору. Аргумент идентификатор представляет собой имя, присвоенное свойству объекта, переменной или элементу массива.

-= (присвоение результата вычитания) (4+)*Синтаксис* A-=B

Этот оператор возвращает разность выражений A и B и присваивает ее значение выражению A.

Пример Выражение `y-=x` эквивалентно `y=y-x`.
 `var x=10, y=20`

`y-=x` возвращает 10.

== (равенство) (4+)*Синтаксис* A==B

Если выражение A равно выражению B, оператор возвращает true. В противном случае оператор возвращает false.

=== (строгое равенство) (6)*Синтаксис* A===B

Подобно оператору равенства этот оператор сравнивает выражения A и B и возвращает логическое значение в зависимости от того, равны они ли нет. Их отличие состоит в том, что оператор строгого равенства учитывает также тип данных выражений.

Пример Возьмем следующие выражения:

```
str="1";
num=new Number(1);
```

Оператор равенства `str==num` возвращает значение true, так как значения переменных равны (единица). Но оператор строгого равенства `str===num` возвращает значение false, поскольку переменные `str` и `num` содержат данные различных типов.

> (больше) (4+)*Синтаксис* A>B

Если выражение A больше выражения B, оператор > возвращает `true`. В противном случае (то есть если A меньше или равно B) оператор возвращает `false`. Оператор может также обрабатывать строки. Заглавные буквы, расположенные ближе к A, имеют меньшие значения, наибольшие значения имеют строчные буквы, расположенные ближе к z.

>= (больше или равно) (4+)*Синтаксис* A>=B

Если выражение A больше выражения B или равно ему, оператор >= возвращает `true`. В противном случае оператор возвращает `false`. Оператор может также обрабатывать строки. Заглавные буквы, расположенные ближе к A, имеют меньшие значения, наибольшие значения имеют строчные буквы, расположенные ближе к z.

typeof (5+)*Синтаксис* typeof A

Оператор `typeof` возвращает тип данных, содержащихся в выражении A. Это могут следующие типы данных: `Boolean`, `function`, `movieclip`, `number`, `object` и `string`.

Пример

```
obj - new Object;
trace("2: "+typeof 2);
trace("text: "+typeof "text");
trace("true: "+typeof true);
trace("obj: "+typeof obj);
```

В результате выполнения этого кода в окне будет отображено следующее:

```
2: number
text: string
true: Boolean
obj: object
```

То есть 2 — это число, `text` — значение типа `string`, `true` — логическое значение, а `obj` — пользовательский объект. Для элементов, не содержащих данных, оператор возвращает значение `undefined` или `null`.

Объектные ссылки и ссылки на временную шкалу

Объектные ссылки — это элементы, указывающие на клипы, уровни фильма, переменные, объекты и функции Flash-фильмов. Правильное использование этих элементов является залогом успеха в указании точных путей к различным объектам. Если встал вопрос об указании пути к какому-либо объекту, его правильность всегда можно проверить при помощи панелей `Movie Explorer`, `Debugger` или кнопки `Insert Target Path` панели `Actions`.

_global (6)

Синтаксис `global.имя_переменной`
`global.имя_функции`

Эта ссылка позволяет создавать глобальные переменные, объекты или функции. Определение *глобальный* означает, что переменная, объект или функция доступны

из любой временной шкалы вашего фильма. Вместо того чтобы указывать путь, например `_root.myFunction`, ссылка `_global` позволяет просто вызвать функцию `myFunction`. Для того чтобы к функции или переменной можно было получить доступ по имени из любой временной шкалы, при их инициализации укажите ссылку `_global`.

Пример

В приведенном коде переменная `startGame` объявляется как глобальная.

```
_global.startGame=getTimer();
```

Эту переменную можно вызвать в любом месте фильма, просто указав ее имя, `startGame`. Если вы не воспользовались ссылкой `_global` при создании этой переменной в фильме главной временной шкалы, к ней придется обращаться, указывая путь `_root.startGame`.

_level (4+)

Синтаксис `Level X`

Эта ссылка указывает на временную шкалу фильма, загруженного на уровне X. Приложение Flash Player может одновременно хранить и воспроизводить множество фильмов. Главная временная шкала (первый загруженный фильм) находится на уровне 0. Все последующие фильмы загружаются на более высоких уровнях: уровне 1, уровне 2 и т. д. Ссылка `_level` применяется, чтобы загрузить или выгрузить дополнительный фильм или управлять его воспроизведением.

_parent (4+)

Синтаксис `экземпляр._parent`

При помощи ссылки `_parent` можно обратиться к временной шкале фильма, в котором содержится клип или кнопка. Например, при помощи ссылки `_parent` можно обратиться к временной шкале главного фильма из экземпляра клипа или кнопки, в нем находящегося. В фильме главной временной шкалы эту ссылку использовать нельзя, поскольку он расположен на самом высоком уровне из всех возможных (`_level0` или `_root`). Ссылка `_parent` особенно полезна при использовании относительных путей к объекту.

Пример

Если экземпляр клипа `arrow` находится в фильме главной временной шкалы, вызвав нижеприведенный сценарий во временной шкале клипа `arrow`, вы остановите воспроизведение фильма.

```
_parent.stop();
```

Если в клипе `arrow` содержится вложенный экземпляр клипа `Letcher`, вызвав следующий сценарий во временной шкале клипа `Letcher`, вы остановите воспроизведение фильма главной временной шкалы, сделаете экземпляр `arrow` полупрозрачным и повернете на 180° экземпляр `head`, также находящийся в клипе `arrow`.

```
_parent._parent.stop();
_parent.alpha=50;
_parent.head.rotation=180;
```

_root (4+)

Синтаксис `_root.экземпляр_клипа`
 `_root.действие`
 `_root.свойство`

Ссылка `_root` указывает на главную временную шкалу фильма (`level0`); ее можно использовать для создания полного пути к объекту, указанному в аргументе `экземпляр_клипа`, или для управления главной временной шкалой из любой части фильма.

this (5+)

Синтаксис `this.действие`
 `this.свойство`
 `this.nestedClipA.nestedClipB`

Ссылка `this` используется как самовывоз экземпляров символов для текущей временной шкалы. Если вызвать `this` из обработчика событий мыши, связанного с кнопкой, она укажет на временную шкалу фильма, в котором эта кнопка находится. Если эта ссылка вызвана из обработчика событий клипа, она укажет на временную шкалу этого клипа. Ссылку `this` можно использовать для управления воспроизведением, установкой/получением свойств текущей временной шкалы. Она также применяется при создании пути к объекту, начинающемуся с указания текущей временной шкалы.

Предопределенные объекты

Предопределенные объекты (`Array`, `Color`, `Date`, `Key`, `Math`, `Mouse`, `MovieClip`, `Object` (`Generic`), `Sound`, `String` и `TextField`) управляют свойствами и данными, связанными со встроенными Flash-объектами. Полный список встроенных объектов вы можете получить, воспользовавшись панелью `Reference Flash MX`.

Объект Array (5+)

Синтаксис `new Array()`
 `new Array(размер)`
 `new Array(элемент0, элемент1, элемент2...элементх)`

Массив представляет собой упорядоченный набор данных. Свойства и методы объекта `Array` можно использовать для хранения, извлечения и обработки элементов, находящихся в массиве.

Для создания объекта `Array` можно воспользоваться функцией `new Array()`. Например:

```
list=new Array0;
```

Указав эту функцию без аргументов, вы создадите новый пустой массив. Однако функция может принимать и необязательные аргументы. Указав целое число в аргументе `размер`, вы установите размер массива, то есть количество элементов в нем.

```
week=new Array(7);
```

В данном случае создается пустой массив `week`, в который можно записать максимум 7 элементов. Также допускается заполнение массива сразу при его создании.

```
groceries=new Array("pasta","bread","milk");
```

Массив `groceries` содержит три элемента типа `string`. В массиве могут содержаться и смешанные данные.

```
home=new Array();
home[0]="Chicago";
home[1]=60657;
home[2]=["Lakeview","Wrigleyville"];
```

В этом примере при создании массива `home` применяется оператор доступа к элементу массива (`[]`). Массив `home` содержит три элемента: "Chicago" (тип `string`), 60657 (тип `number`) и другой массив с элементами "Lakeview" и "Wrigleyville" типа `string`. Все элементы массива сохраняются в памяти последовательно, и обращаться к элементу массива следует по смещению его адреса, то есть индексу (индекс первого элемента — 0, второго — 1 и т. д.). При доступе к элементу массива по индексу следует указывать оператор `[]`. Например, выражение `trace(home[1])` возвращает значение 60657, то есть второй элемент массива.

Свойства

`Array.length` возвращает количество элементов в массиве.

Методы

`Array.concat()` создает новый массив, добавляя дополнительные элементы в уже существующий.

`Array.pop()` удаляет последний элемент массива, но возвращает значение этого последнего элемента.

`Array.push()` добавляет дополнительные элементы в конец массива.

`Array.shift()` удаляет первый элемент массива, но возвращает значение этого первого элемента.

`Array.slice()` создает новый массив, используя некоторую часть уже существующего.

`Array.sort()` упорядочивает элементы массива в соответствии с правилами, установленными пользователем.

`Array.toString()` возвращает элементы массива в одной строке, разделяя их при помощи запятых.

`Array.unshift()` добавляет дополнительные элементы в начало массива.

Array.concat (5+)

Синтаксис `имя_массива.concat(элемент1,элемент2...элементх)`

Этот метод используется для добавления дополнительных элементов в конец существующего массива. При этом создается совершенно новый массив.

Пример

Создадим массив `letter1`.

```
letter1=new Array("a","b","c");
```

Создадим массив `letter2`, добавив два элемента в массив `letter1`.

```
letter2=letter1.concat("d","e");
```

В массиве `letter2` содержатся следующие элементы.

```
["a","b","c","d","e"]
```

Array.length (5+)*Синтаксис* *имя_массива*.length

Это свойство можно использовать как для установки, так и для извлечения размера массива, то есть количества элементов, содержащихся в массиве, указанном в аргументе *имя_массива*. Поскольку нумерация индексов элементов в массиве начинается с 0, размер массива всегда на 1 больше, чем индекс последнего его элемента. Значение свойства `length` изменяется, если в массив добавляются элементы или удаляются из него. Таким образом, свойство `length` всегда отражает текущее состояние массива, даже если он на данный момент пуст.

*Пример*Создадим массив `alphabet`.

```
alphabet=new Array("a","b","c");
```

В окне Output будет выведено значение 3, поскольку в массиве `alphabet` всего три элемента.

```
trace(alphabet.length);
```

Array.pop (5+)*Синтаксис* *имя_массива*.pop()

Этот метод применяется для удаления последнего элемента массива, указанного в аргументе *имя_массива*, и возвращает его значение. При вызове данного метода последний элемент массива удаляется, а значение, находившееся в этом элементе, возвращается. Метод противоположен методу `shift()`, который удаляет первый элемент массива и возвращает его значение.

*Пример*В следующем выражении удаляется последний элемент массива `alphabet`.

```
alphabet=new Array("a","b","c");
popped=alphabet.pop();
```

В окне Output будет выведена надпись с удален, поскольку именно элемент `c` был последним в массиве:

```
trace(popped+"удален");
```

Array.push (5+)*Синтаксис* *имя_массива*.push(*элемент1*, *элемент2*...*элементх*)

Этот метод добавляет указанные элементы в массив. В отличие от метода `concat()` новый массив при этом не создается, а в отличие от метода `unshift()` элементы помещаются в конец, а не в начало массива.

Пример Создадим массив `alphabet`.

```
alphabet=new Array("a","b","c");
```

В следующем выражении в массив `alphabet` добавляются элементы `d`, `e` и `f`.

```
alphabet.push("d","e","f");
```

Array.shift (5+)*Синтаксис* `имя_массива.shift()`

Этот метод применяется для удаления первого элемента массива, указанного в аргументе `имя_массива`, и возвращает значение удаляемого элемента. При вызове данного метода первый элемент массива удаляется, а значение этого элемента возвращается. Метод противоположен методу `Array.pop()`, который удаляет последний элемент массива и возвращает его значение.

Пример

В следующем выражении удаляется первый элемент массива `alphabet`.

```
alphabet=new Array("a","b","c");
shifted=alphabet.shift();
```

В окне Output будет выведена строка `a` удален, поскольку именно элемент `a` был первым элементом массива.

```
trace(shifted+"удален");
```

Array.slice (5+)*Синтаксис* `имя_массива.slice(индекс_1,индекс_2)`

Метод `Array.slice()` применяется для создания нового массива на основе определенной части уже существующего, указанного в аргументе `имя_массива`. Чтобы выделить эту часть массива, вам следует указать два аргумента: `индекс_1` и `индекс_2`. Оба аргумента содержат индексы, заданные с учетом того, что начальный элемент массива имеет индекс 0. Аргумент `индекс_1` представляет первый элемент выделенной части существующего массива, которая будет включена в новый массив. Если этот аргумент содержит отрицательное число, отсчет будет вестись с конца существующего массива. Например, `-1` — номер последнего элемента массива, `-2` — номер предпоследнего элемента и т. д. Аргумент `индекс_2` — это последний элемент выделенной части существующего массива, который будет включен в новый массив. Аргумент `индекс_2` может быть задан как отрицательное число — так же как и аргумент `индекс_1`. Если не указать этот аргумент, будет использовано его значение по умолчанию — `имя_массива.length` — и все оставшиеся элементы существующего массива перейдут в новый массив. Короче говоря, в новый массив входят элементы, находящиеся между элементами `индекс_1` и `индекс_2` (включительно) существующего массива.

Пример Создадим новый массив:

```
nums=new Array("one","two","three","four","five");
```

Следующие выражения эквивалентны. Во всех создается новый массив с двумя последними элементами массива — `four` и `five`.

```
sub1=nums.slice(3,5);
sub2=nums.slice(3);
sub3=nums.slice(-2,5);
```

Array.sort (5+)*Синтаксис* `имя_массива.sort()`
`имя_массива.sort(функция_сортировки)`

При помощи метода `Array.sort()` элементы массива, указанного в аргументе `имя_массива`, сортируются в определенном порядке. Если функция сортировки не

указана, элементы будут сортироваться в таком порядке: вначале — целые числа, затем — символы верхнего регистра (в алфавитном порядке) и, наконец, символы нижнего регистра.

Если вы укажете функцию сортировки, массив будет отсортирован в соответствии с правилами, устанавливаемыми этой функцией. Функция сортировки должна удовлетворять вашим нуждам. При ее создании следует соблюдать некоторые простые правила. Функция должна предоставлять возможность сравнивать два значения. Если вы хотите, чтобы при сортировке первое значение предшествовало второму, функция должна возвращать -1 . Если при сортировке второе значение должно предшествовать первому, функция должна возвращать 1 . Для равных значений, не требующих сортировки, функция будет возвращать 0 . Элементы массива сортируются в таком порядке: вначале отрицательные числа, затем — положительные. Например, отсортируем значения А и Б в возрастающем порядке.

$A < B$ — функция возвращает -1 .

$A > B$ — функция возвращает 1 .

$A == B$ — функция возвращает 0 .

Чтобы отсортировать значения в убывающем порядке, вам просто нужно поменять значения, возвращаемые функцией. Когда $A < B$, функция будет возвращать 1 ; если $A > B$ — -1 ; при равенстве значений — 0 .

Array.unshift (5+)

Синтаксис `имя_массива.unshift(элемент1, элемент2...элементх)`

Этот метод добавляет элементы в массив, указанный в аргументе `имя_массива`. В отличие от метода `concat()` новый массив при этом не создается, а в отличие от метода `push()` элементы помещаются в начало, а не в конец массива.

Пример `Создадим массив alphabet.
alphabet=new Array("d","e","f");`

В следующем выражении в массив `alphabet` добавляются элементы `a`, `b` и `c`.

```
alphabet.unshift("a","b","c");
```

Объект Color (5+)

Синтаксис `new Color(имя_клипа)`

В объекте `Color` хранится информация, позволяющая работать со значениями составляющих цвета и прозрачности клипа или фильма, указанного в аргументе `имя_клипа`. Перед вызовом методов объекта `Color` необходимо создать новый объект посредством функции `new Color()`.

Методы

`Color.getRGB()` возвращает текущее RGB-значение объекта `Color`.

`Color.getTransform()` возвращает текущее значение параметра трансформации объекта `Color`.

`Color.setRGB()` задает текущее RGB-значение объекта `Color`.

`Color.setTransform()` задает текущее значение параметра трансформации объекта `Color`.

Color.getRGB (5+)*Синтаксис* объектColor.getRGB

Этот метод возвращает RGB-значение для указанного объекта Color.

Пример

При щелчке мышью на клипе, с которым связан приведенный ниже сценарий, RGB-значения цвета указанного объекта переносятся из одного клипа в другой. Метод getRGB() извлекает значение цвета объекта orbColor клипа orb и применяет его к текущему экземпляру клипа.

```
onClipEvent(mouseUp){
    if(this.hitTest(_root._xmouse._root._ymouse,true)){
        colorTrans=new Color(this);
        colorTrans.setRGB(_root.orb.orbColor.getRGB());
    }
}
```

Color.getTransform (5+)*Синтаксис* объектColor.getTransform()

Этот метод извлекает текущие параметры трансформации, установленные для указанного объекта Color.

Color.setRGB (5+)*Синтаксис* colorObject.setRGB

Этот метод применяется, чтобы задать RGB-значение для указанного объекта Color.

Пример

В приведенном сценарии устанавливается красный цвет для экземпляра клипа, с которым он связан.

```
onClipEvent(load){
    clipColor=new Color(this);
    clipColor.setRGB(0xFF0000);
}
```

См. также описание метода Color.getRGB().

Color.setTransform (5+)*Синтаксис* объектColor.setTransform(Объект1)

Метод setTransform() позволяет манипулировать цветовыми составляющими объекта Color: красной, зеленой, синей и альфа-значением. Вы можете устанавливать долю (в процентах) или значение каждой составляющей. Чтобы воспользоваться этим методом, сначала необходимо создать объект и присвоить ему свойства, приведенные в табл. Б.1. Созданный объект становится объектом, указанным в аргументе Объект1. При вызове метода setTransformO его свойства применяются к объекту Color.

Таблица Б.1. Свойства объекта Object1

Свойство	Диапазон	Описание
ra	от -100 до 100	Доля красного
rb	от -255 до 255	Значение красного
ra	от -100 до 100	Доля зеленого

Свойство	Диапазон	Описание
gb	от -255 до 255	Значение зеленого
ba	от -100 до 100	Доля синего
bb	от -255 до 255	Значение синего
aa	от -100 до 100	Степень прозрачности (значение альфа-канала в процентах)
ab	от -255 до 255	Альфа-значение

Доля в процентах — это значение, применяемое к исходному значению цветовой составляющей. Например, если установить значение `ba=50`, голубая составляющая цвета объекта станет равной всего лишь половине своей исходной величины. Значение также можно увеличивать или уменьшать. Если установить `gb=255`, значение зеленой составляющей цвета объекта подскакивает до максимума. Примеры применения этих свойств и методов можно найти в главе 16.

Объект Date (5+)

Синтаксис `new Date()`
`new Date(год, месяц, число, часы, минуты, секунды, миллисекунды):`

В ActionScript объект `Date` позволяет получить текущую дату в соответствии с местным временем или временем по Гринвичу. Местное время определяется настройками системы компьютера, на котором воспроизводится фильм. В этом справочнике не рассматриваются методы объекта `Date`, связанные с временем по Гринвичу.

Объект `Date` позволяет получить текущие дату и время, сохранить их и использовать в фильмах. Существует два способа создания объекта `Date`. Чтобы создать объект для текущих даты и времени, можно воспользоваться функцией `new Date()`. Если необходимо создать объект для указанной даты, используют аргументы объекта `Date`.

- ▶ Год — целое число, обозначающее год. Чтобы указать год с 1900 по 1999, используйте двузначные числа от 0 до 99 (например, число 73 означает 1973 год). Годы с 1000 по 1899, а также с 2000 и далее обозначаются четырехзначными числами. Трехзначные числа применяются для обозначения годов до 1000 года н. э.
- ▶ Месяц — целое число от 0 (январь) до 11 (декабрь), обозначающее месяц.
- ▶ Число — целое значение от 1 до 31, обозначающее число месяца. Это — необязательный аргумент.
- ▶ Часы — целое число от 0 (полночь) до 23 (11 часов вечера), обозначающее часы. Аргумент часы необязателен.
- ▶ Минуты — целое число от 0 до 59, обозначающее минуты. Это — необязательный аргумент.
- ▶ Секунды — целое число от 0 до 59, обозначающее секунды. Это — необязательный аргумент.
- ▶ Миллисекунды — целое число от 0 до 999, обозначающее миллисекунды, необязательный аргумент. В ActionScript миллисекунды имеют большое значение, поскольку вся информация, касающаяся дат, хранится в виде количества миллисекунд, прошедших начиная с полуночи 1 января 1970 года. Методы

объекта `Date` позволяют манипулировать этими длинными громоздкими числами и переводить их в понятные даты с днями, месяцами, годами и т. д.

Методы

Создав объект `Date`, вы можете воспользоваться следующими его методами, позволяющими извлекать или устанавливать дату.

`Date.getDate()` возвращает число месяца.

`Date.getDay()` возвращает день недели.

`Date.getFullYear()` возвращает год в виде четырехзначного числа.

`Date.getHour()` возвращает часы.

`Date.getMilliseconds()` возвращает миллисекунды.

`Date.getMinutes()` возвращает минуты.

`Date.getMonth()` возвращает месяц года.

`Date.getSeconds()` возвращает секунды.

`Date.getTime()` возвращает количество миллисекунд, прошедших с 1 января 1970 года.

`Date.getYear()` возвращает год относительно 1900 года.

`Date.setDate()` позволяет задать число.

`Date.setFullYear()` позволяет задать год в виде четырехзначного числа.

`Date.setHour()` позволяет задать часы.

`Date.setMilliseconds()` позволяет задать миллисекунды.

`Date.setMinutes()` позволяет задать минуты.

`Date.setMonth()` позволяет задать месяц года.

`Date.setSeconds()` позволяет задать секунды.

`Date.setTime()` задает дату в миллисекундах относительно полуночи 1 января 1970 года.

`Date.setYear()` задает год в виде как двузначного, так и четырехзначного числа.

`Date.toString()` возвращает строку, содержащую дату как по местному времени, так и по Гринвичу.

Пример

В этом сценарии создается новый объект `Date` для текущих даты и времени. Затем в нем используются различные методы для форматирования даты и в соответствующих текстовых полях выводится значение времени.

```
//Создаем новый объект Date;
today=new Date();
//Форматируем каждую единицу времени
//в отдельном текстовом поле
hours=today.getHours();
minutes=today.getMinutes();
seconds=today.getSeconds();
```

Следующий сценарий связан с кнопкой. В нем вычисляется количество дней, остающихся до дня рождения, указанного в текстовых полях `month` и `day`.

```
on(release){
  today=newDate();
  bDay=new Date(today.getFullYear(),month-1,day)
  var wait=Math.ceil((bDay-today)/86400000);
  if(wait<0){
    printout="Поздравляю с прошедшим днем рождения!";
  }else{
    printout=wait+" дней осталось до следующего дня рождения.";
  }
}
```

В приведенном примере используются два объекта `Date`: один — для текущей даты (`today`), а другой — для даты дня рождения, указанной в полях `month` и `day`. Обратите внимание: поскольку в `ActionScript` нумерация месяцев начинается с 0 (0 обозначает январь), необходимо вычесть 1 из значения, введенного в поле `month`. Разница между значениями двух дат вычисляется в миллисекундах. Чтобы получить результат в более понятных числах, его делят на 86400000 (количество миллисекунд в одном дне) и округляют при помощи метода `Math.ceil()` до следующего большего целого числа. Результат выводится в поле `printout`.

Date.getDate (5+)

Синтаксис объектDate.getDate()

Метод `getDate()` возвращает значение дня месяца для объекта `Date`. Данное значение представляет собой целое число от 0 до 31.

Date.getDay (5+)

Синтаксис объектDate.getDay()

Метод `getDay()` возвращает значение дня недели для объекта `Date`. Значение дня недели представляет собой целое число от 0 (воскресенье) до 6 (суббота).

Date.getFullYear (5+)

Синтаксис объектDate.getFullYear()

Метод `getFullYear()` возвращает значение года для объекта `Date` в виде четырехзначного числа.

Date.getHours (5+)

Синтаксис объектDate.getHours()

Метод `getHours()` возвращает значение часов для объекта `Date`. Это значение представляет собой целое число от 0 (полночь) до 23 (11 часов вечера). Обратите внимание на то, что объект `Date` не поддерживает обозначения А.М. (до полудня) и Р.М (после полудня).

Date.getMilliseconds (5+)

Синтаксис объектDate.getMilliseconds()

Метод `getMilliseconds()` возвращает значение миллисекунд для объекта `Date`. Значение миллисекунд представляет собой целое число от 0 до 999.

Date.getMinutes (5+)*Синтаксис* объектDate.getMinutes()

Метод `getMinutes()` возвращает значение минут для объекта `Date`. Это значение представляет собой целое число от 0 до 59.

Date.getMonth (5+)*Синтаксис* объектDate.getMonth()

Метод `getMonth()` возвращает значение месяца для объекта `Date`. Значение месяца представляет собой целое число от 0 (январь) до 11 (декабрь).

Date.getSeconds (5+)*Синтаксис* объектDate.getSeconds()

Метод `getSeconds()` возвращает значение секунд для объекта `Date`, которое представляет собой целое число от 0 до 59.

Date.getTime (5+)*Синтаксис* объектDate.getTime()

Во Flash все объекты `Date` сохраняются как одно число, представляющее собой количество миллисекунд, прошедших с полуночи 1 января 1970 года. Метод `getTime()` возвращает это количество миллисекунд для объекта `Date`. Такой способ обработки дат позволяет легко их сравнивать.

Date.getYear (5+)*Синтаксис* объектDate.getYear()

Метод `getYear()` применяется для получения значения года для объекта `Date` относительно 1900 года. Например, для 1973 года метод возвратит значение 73, а для 2002 года - 102.

Date.setDate (5+)*Синтаксис* объектDate.setDate(число)

Метод `setDate()` позволяет задать значение числа месяца для объекта `Date`. Значение указывается в аргументе число как целое число от 0 до 31.

Date.setFullYear (5+)*Синтаксис* объектDate.setFullYear(год, месяц, число)

При помощи метода `setFullYear()` можно задать точную дату для объекта `Date`. Метод принимает следующие аргументы:

- ▶ год — четырехзначное число, обозначающее год;
- ▶ месяц — число от 0 (январь) до 11 (декабрь), обозначающее месяц;
- ▶ число — число месяца от 1 до 31.

Date.setHours (5+)*Синтаксис* объектDate.setHours()

Данный метод применяется, чтобы установить значение часов для объекта `Date`. Это значение указывается как целое число от 0 (полночь) до 23 (11 часов вечера).

Date.setMilliseconds (5+)*Синтаксис* объектDate.setMilliseconds()

Этот метод применяется, чтобы установить значение миллисекунд для объекта Date. Значение должно быть целым числом от 0 до 999.

Date.setMinutes (5+)*Синтаксис* объектDate.setMinutes()

Метод setMinutes() используется для того чтобы задать значение минут для объекта Date. Значение минут указывается в виде целого числа от 0 до 59. Если указать значение, выходящее за диапазон чисел от 0 до 59, значение часа объекта Date будет изменено на предыдущий или следующий час соответственно. Например, значение 60 минут эквивалентно первой минуте следующего часа.

Date.setMonth (5+)*Синтаксис* объектDate.setMonth()

Метод setMonth() задает значение месяца года для объекта Date. Значение месяца указывается в виде целого числа от 0 (январь) до 11 (декабрь). Если указать значение, выходящее за диапазон чисел от 0 до 11, значение года объекта Date будет изменено на предыдущий или следующий год соответственно.

Date.setSeconds (5+)*Синтаксис* объектDate.setSeconds()

Этот метод позволяет задать значение секунд для объекта Date, которое представляет собой целое число от 0 до 59. Если указать значение, выходящее за диапазон чисел от 0 до 59, значение минут объекта Date будет изменено на предыдущую или следующую минуту соответственно. Например, значение секунд -1 эквивалентно последней (59-ой) секунде предыдущей минуты.

Date.setTime (5+)*Синтаксис* объектDate.setTime()

Этот метод можно использовать, чтобы задать количество миллисекунд для объекта Date относительно 1 января 1970 года. Количество миллисекунд указывается в виде целого числа.

Date.setYear (5+)*Синтаксис* объектDate.setYear(год, месяц, число)

Метод setYear() позволяет установить точную дату для объекта Date. Он принимает следующие аргументы.

- ▶ Год — целое число, обозначающее год: от 0 до 99 — годы с 1900 по 1999; четырехзначные числа — годы с 1000 по 1899 и с 2000 и далее, трехзначные числа — годы до 1000 года н. э.
- ▶ Месяц — число от 0 (январь) до 11 (декабрь), обозначающее месяц.
- ▶ Число — число от 1 до 31, обозначающее день месяца.

Date.toString (5+)*Синтаксис* объектDate.toString()

Этот метод применяется для преобразования информации объекта Date в строку: день недели, месяц, число, часы, минуты, секунды, часовой пояс и год.

Объект Key (5+)

Синтаксис `Key.метод()`
 `Key.свойство`

С помощью объекта Key контролируется нажатие клавиш в фильме. Он предоставляет великолепное средство для создания интерактивных элементов управления в фильме, которые вызываются с клавиатуры. Методы объекта Key применяются для идентификации нажатых клавиш, а его свойства используются при создании комбинаций клавиш быстрого вызова команд с клавиатуры. (Полный список кодов клавиш вы найдете в табл. Б.4 в конце этого раздела.)

Методы

`Key.getAscii()` возвращает значение ASCII последней нажатой клавиши.

`Key.getCode()` возвращает значение кода последней нажатой клавиши.

`Key.isDown()` проверяет, нажата ли в данный момент указанная клавиша.

Свойства В табл. Б.2 приведены свойства объекта Key.

Таблица Б.2. Свойства объекта Key

Синтаксис	Клавиша	Код
<code>Key.BACKSPACE</code>	Backspace	8
<code>Key.CAPSLOCK</code>	Caps Lock	20
<code>Key.CONTROL</code>	Control	17
<code>Key.DELETEKEY</code>	Delete	46
<code>Key.DOWN</code>	Down arrow	40
<code>Key.END</code>	End	35
<code>Key.ENTER</code>	Enter (цифровая клавиатура)	13
<code>Key.ESCAPE</code>	Esc	27
<code>Key.HOME</code>	Home	36
<code>Key.INSERT</code>	Insert	45
<code>Key.LEFT</code>	Left arrow	37
<code>Key.PGDN</code>	Page Down	34
<code>Key.PGUP</code>	Page Up	33
<code>Key.RIGHT</code>	Right arrow	39
<code>Key.SHIFT</code>	Shift	16
<code>Key.SPACE</code>	Spacebar	32
<code>Key.TAB</code>	Tab	9
<code>Key.UP</code>	Up arrow	38

Key.getAscii (5+)

Синтаксис `Key.getAscii()`

Этот метод возвращает ASCII-значение последней нажатой клавиши. В отличие от метода `Key.getCode()`, `Key.getAscii()` различает верхний и нижний регистр букв, но не различает числа, которые были введены с символьной и цифровой клавиатур.

Пример См. описание метода `Key.getCode()`.

Key.getCode (5+)*Синтаксис* Key.getCode()

Данный метод возвращает значение кода последней нажатой клавиши. Коды клавиш — это стандартные значения, представляющие все клавиши в цифровом виде. (Полный список кодов клавиш вы найдете в табл. С.4 этого раздела.)

Пример

Следующая функция может применяться для того, чтобы извлечь значение кода клавиши и ее ASCII-значение.

```
keyCheck=newObject();
keyCheck.onKeyDown=function(){
    trace("Код: "+Key.getCode());
    trace("ascii-значение: "+Key.getAscii());
}
Key.addListener(keyCheck);
```

Объект keyCheck идентифицирует нажатые клавиши или, как в данном случае, реагирует на событие onKeyDown.

Key.isDown (5+)*Синтаксис* Key.isDown(код_клавиши)

В отличие от методов Key.getAscii() и Key.getCode, этот метод проверяет, удерживается ли нажатой клавиша, определенная аргументом код_клавиши. Аргумент код_клавиши может вводиться как цифровой код или как свойство объекта Key (например, Key.SPACE).

Пример

В приведенном сценарии представлена простая навигационная система клипа с использованием клавиш управления курсором.

```
onClipEvent(enterFrame){
    if(Key.isDown(Key.UP)){
        this._y-=10;
    }else if(Key.isDown(key.DOWN)){
        this._y+=10;
    }
    if(Key.isDown(Key.LEFT)){
        this._x-=10;
    }elseif(key.isDown(Key.RIGHT)){
        this._x+=10;
    }
}
```

Обратите внимание на то, что код к клавишам перемещения курсора в вертикальном (со стрелками вверх и вниз) и горизонтальном направлениях (со стрелками влево и вправо) находится в отдельных блоках if...else. Это позволяет пользователю выполнять перемещение клипа в двух направлениях одновременно. Клип может перемещаться одновременно вверх или вниз и вправо или влево, то есть его можно перемещать по диагонали.

Объект Math (5+)

Синтаксис `Math.свойство`
 `Math.метод(выражение)`

Объект `Math` позволяет создавать сценарии, в которых выполняются математические расчеты. При помощи методов этого объекта вы получаете доступ к математическим функциям; последние можно использовать для обработки чисел.

Методы

`Math.abs()` вычисляет модуль числа.

`Math.ceil()` округляет число до следующего большего целого числа.

`Math.floor()` округляет число до предыдущего меньшего целого числа.

`Math.max()` вычисляет большее из двух чисел.

`Math.min()` вычисляет меньшее из двух чисел.

`Math.pow()` возводит число в степень.

`Math.random()` возвращает произвольное число от 0 до 1.

`Math.round()` округляет число до ближайшего целого числа.

Math.abs (5+)

Синтаксис `Math.abs(выражение)`

Метод `Math.abs()` применяется для получения *абсолютной величины* выражения. Абсолютная величина действительного числа равна этому числу, если оно положительно, равна противоположному числу, если оно отрицательно, и равна нулю, если число равно нулю. Этот метод преобразует отрицательные значения в положительные, а положительные оставляет без изменений.

Math.ceil (5+)

Синтаксис `Math.ceil(выражение)`

Метод `Math.ceil()` преобразует число с плавающей точкой, указанное в выражении, в ближайшее большее целое число.

`Math.ceil(0.00001)` возвращает 1.

`Math.ceil(1.9)` возвращает 2.

Math.floor (5+)

Синтаксис `Math.floor(выражение)`

Метод `Math.floor()` преобразует число с плавающей точкой, указанное в выражении и `i`, в ближайшее меньшее целое число.

`Math.floor(0.00001)` возвращает 0.

`Math.floor(1.9)` возвращает 1.

Math.max (5+)

Синтаксис `Math.max(число1, число2)`

Метод `Math.max()` возвращает большее из двух чисел, указанных в аргументах `число1` и `число2`.

Math.min (5+)*Синтаксис* `Math.min(число1,число2)`

Метод `Math.min()` возвращает меньшее из двух чисел, указанных в аргументах `число1` и `число2`.

Math.pow (5+)*Синтаксис* `Math.pow(число,exp)`

При помощи этого метода число возводится в степень `exp`.

Math.random (5+)*Синтаксис* `Math.random()`

Этот метод возвращает произвольное число, большее 0, но меньшее 1. Метод `Math.random()` применяется для произвольного выбора чисел. Эти числа затем можно использовать в функциях, которые произвольным образом размещают клипы на области действий и т. д.

Пример

Приведенное ниже выражение возвращает произвольное целое число от 0 до 10.

```
Math.floor(Math.random()*11);
```

Метод `Math.floor()` округляет результат до ближайшего меньшего целого числа. Поскольку метод `Math.random()` никогда не возвращает 1, то для того, чтобы в результате получить 10, мы умножаем число, возвращаемое этим методом, на 11, которое на единицу больше верхней границы диапазона (0;10). Пусть, например, метод `Math.floor()` возвратил 0,999. Умножив это число на 11 и округлив до ближайшего меньшего целого, мы получим 10.

Точно так же можно определить как верхнюю, так и нижнюю границу для вычисления произвольного числа. В следующем сценарии возвращается произвольное число от 50 до 100, которое используется для установки значения свойства `_alpha`.

```
onClipEvent(load){
    vis=Math.floor(Math.random()*101)+50;
    this._alpha=vis;
}
```

Math.round (5+)*Синтаксис* `Math.round(выражение)`

Этот метод применяется для округления числа с плавающей точкой до ближайшего целого числа. Если значение десятых в дроби равно или больше 0,5, выражение округляется до следующего большего целого числа; если значение десятых меньше 0,5, выражение округляется до ближайшего меньшего целого числа.

Пример

```
Math.round(1.5) возвращает 2.
Math.round(1.4) возвращает 1.
Math.round(-1.5) возвращает -1.
Math.round(-1.6) возвращает -2.
```

Объект Mouse (5+)*Синтаксис* `Mouse.метод`

Объект `Mouse` позволяет отображать или не отображать указатель мыши в приложении Flash Player. Методы этого объекта обычно применяются для замены указателя мыши, используемого системой по умолчанию (стрелка), пользовательским указателем.

Методы

`Mouse.hide()` — указатель мыши не отображается.

`Mouse.show()` — указатель мыши отображается.

Mouse.hide (5+)*Синтаксис* `Mouse.hide()`

Этот метод помогает скрыть указатель мыши, находящийся на окне приложения Flash Player (в границах Flash-фильма). Если указатель мыши поместить за пределы области отображения кадров фильма, отображаться будет указатель, применяемый в системе по умолчанию.

Пример

Если с клипом связать следующий сценарий, этот клип станет настраиваемым указателем мыши. Следует воспользоваться событием `mouseMove`, чтобы обеспечить перемещение клипа, представляющего собой настраиваемый указатель, вместе с указателем мыши.

```
onClipEvent(load){
    Mouse.hide();
}
onClipEvent(mouseMove){
    this._x=_root._xmouse;
    this._y=_root._ymouse;
```

Mouse.show (5+)*Синтаксис* `Mouse.show()`

С помощью этого метода отображается указатель мыши, применяемый по умолчанию, после того как он был скрыт при помощи метода `Mouse.hide()`.

Объект MovieClip (5+/6)*Синтаксис* `экземпляр_клипа.метод`
`путь.экземпляр_клипа.метод`

Объект `MovieClip` не предназначен для создания новых клипов. Во Flash для этого применяется команда `New Symbol` меню `Insert`, при этом переключатель типа символа устанавливается в положение `Movie Clip`. Объект `MovieClip` предоставляет целую серию методов для управления воспроизведением экземпляров клипов. Многие из этих методов подобны действиям, предназначенным для управления воспроизведением клипов и временной шкалой фильмов. Однако некоторые методы предоставляют дополнительные функциональные возможности, не поддерживаемые действиями ActionScript. Методы объектов `MovieClip` могут применяться как по отношению к экземплярам клипов, так и по отношению к временной шкале фильмов. Это относится и к главному фильму, и к SWF-файлам, которые загружаются на новый уровень фильма. Чтобы применить методы

объекта `MovieClip` к клипу, свяжите с ним сценарий при помощи соответствующего обработчика событий `onClipEvent()`. Если нужно обеспечить управление клипом из другого элемента фильма, например из кнопки, экземпляру клипа должно быть присвоено уникальное имя. Это можно сделать на панели `Properties` либо при создании копии клипа, либо при перемещении его из библиотеки. Кроме того, часто приходится указывать путь к клипу, с которым вы хотите работать. Путь необходим, если этот клип находится на другом уровне (`_root`, `_level1`, `_level2` и т. д.) или вложен в другой клип. О способах получения доступа к клипам и временной шкале фильма можно узнать из главы 16.

Методы

`MovieClip.attachMovie()` создает новый экземпляр клипа при помощи клипа, экспортируемого из библиотеки.

`MovieClip.duplicateMovieClip()` создает копию указанного клипа.

`MovieClip.getBytesLoaded()` возвращает количество байтов указанного фильма, которое было загружено.

`MovieClip.getBytesTotal()` возвращает размер указанного фильма в байтах.

`MovieClip.getURL()` загружает внешний файл, найденный по указанному URL.

`MovieClip.globalToLocal()` преобразует координаты области действия в координаты экземпляра клипа.

`MovieClip.gotoAndPlay()` перемещает головку воспроизведения указанного фильма или экземпляра клипа к некоторому кадру его временной шкалы и начинает воспроизведение с этого места.

`MovieClip.gotoAndStop()` перемещает головку воспроизведения указанного фильма или экземпляра клипа к некоторому кадру его временной шкалы и останавливает воспроизведение на этом месте.

`MovieClip.hitTest()` возвращает логическое значение `true` или `false` в зависимости от того, находится ли в области воспроизведения указанного клипа некоторая координатная точка или область воспроизведения другого клипа. (Далее мы будем говорить — *пересекается* ли клип с другим клипом.)

`MovieClip.loadMovie()` загружает внешний SWF-файл в указанный экземпляр клипа.

`MovieClip.loadVariables()` загружает данные, найденные по указанному URL, в экземпляр клипа.

`MovieClip.localToGlobal()` преобразует координаты экземпляра клипа в координаты области действия.

`MovieClip.nextFrame()` переводит головку воспроизведения на один кадр вперед в указанном клипе или фильме и останавливает воспроизведение.

`MovieClip.play()` воспроизводит указанный клип или фильм.

`MovieClip.prevFrame()` переводит головку воспроизведения на один кадр назад в указанном клипе или фильме и останавливает воспроизведение.

`MovieClip.removeMovieClip()` удаляет экземпляр клипа, созданный при помощи метода `duplicateMovieClip()` или `attachMovie()`.

`MovieClip.startDrag()` делает экземпляр клипа перемещаемым.

`MovieClip.stop()` останавливает воспроизведение указанного клипа или фильма.

`MovieClip.stopDrag()` отменяет возможность перемещения клипа.

`MovieClip.unloadMovie()` выгружает внешний SWF-файл, доступ к которому был получен с помощью действия `loadMovie()` или метода `MovieClip.loadMovie()`.

`MovieClip.valueOf()` возвращает полный путь к указанному экземпляру клипа с использованием точечного синтаксиса записи пути.

MovieClip.attachMovie (5+)

Синтаксис `экземпляр_клипа.attachMovie("id","имя",уровень)`

Метод `attachMovie()` вызывает новый экземпляр клипа, указанного в аргументе `имя`, и размещает его на области действия в точке начала координат клипа, указанного в аргументе `экземпляр_клипа`. Точкой начала координат является левый верхний угол области отображения кадров главного фильма или точка регистрации экземпляра клипа. Чтобы вызвать новый клип, сначала необходимо экспортировать его из библиотеки, выбрав команду `Linkage` меню `Options`. Экспортируемое имя (указанное в окне `Linkage Properties`) используется в аргументе `id`, положение в главном фильме задается в аргументе `уровень`. Аргументы `id` и `имя` вводятся как строки, а аргумент `уровень` — как целое число.

Пример

Приведенный сценарий связан с фильмом, который экспортировался из библиотеки под именем `sun` и имеет имя экземпляра `yellow`. Фильм размещается на уровне главного фильма (`_root`) в левом верхнем углу области действия. В этом примере фильм `yellow` перемещается в центр области действия, имеющей размеры 550x400 пикселей.

```
_root.attachMovie("sun","yellow",2);
_root.yellow._x=275;
_root.yellow._y=200;
```

MovieClip.duplicateMovieClip (5+)

Синтаксис `экземпляр_клипа.duplicateMovieClip("новое_имя",уровень)`

Этот метод создает копию указанного экземпляра клипа (аргумент `экземпляр_клипа`), присваивает ей новое имя (аргумент `новое_имя`) и определяет ее положение в главном фильме (аргумент `уровень`). Аргумент `новое_имя` вводится как строка, а `уровень` — как целое число. Несмотря на некоторые отличия в синтаксисе, функциональные возможности этого метода идентичны возможностям действия `duplicateMovieClip()`.

Пример См. описание действия `duplicateMovieClip()`.

MovieClip.getBytesLoaded (5+)

Синтаксис `имя_фильма.getBytesLoaded()`

Этот метод применяется, чтобы определить количество загруженных байтов для фильма, указанного в аргументе `имя_фильма`. Если вы хотите получить количество загруженных байтов главного фильма, этот аргумент можно не указывать, но неплохо бы указать путь `_root` или `this`. Поскольку клип считается частью фильма, в котором он находится, нельзя узнать количество загруженных байтов для экземпляра клипа. Метод `MovieClip.getBytesLoaded()` предназначен для отслеживания процесса загрузки главных фильмов и SWF-файлов.

Пример

С помощью следующего сценария в поле `loaded` выводится количество килобайтов, соответствующее размеру загрузившейся части главного фильма. Чтобы получить значение в килобайтах, количество байтов, возвращенное методом `MovieClip.getBytesLoaded()`, было разделено на 1024.

```
loaded=Math.round(_root.getBytesLoaded()/1024)+"kB";
```

MovieClip.getBytesTotal (5+)

Синтаксис `экземпляр_клипа.getBytesTotal()`

Этот метод возвращает размер указанного фильма или экземпляра клипа в байтах.

Пример

С помощью следующего сценария в поле `total` выводится количество килобайтов, соответствующих размеру главного фильма (общий размер фильма в килобайтах). Чтобы получить значение в килобайтах, количество байтов, возвращенное методом `MovieClip.getBytesTotal()`, было разделено на 1024.

```
total=Math.round(_root.getBytesTotal()/1024)+"kB";
```

MovieClip.getURL (5+)

Синтаксис `имя_получателя.getURL("имя_источника", "окно", "метод_передачи")`

Этот метод объекта `MovieClip` выполняет те же функции, что и действие `getURL()`. Он позволяет передавать переменные из одного фильма (аргумент `имя_источника`) непосредственно в другой фильм (аргумент `имя_получателя`). Полное описание метода и пример к нему вы найдете в описании действия `getURL()`.

MovieClip.globalToLocal (5+)

Синтаксис `экземпляр_клипа.globalToLocal(объект)`

Метод `MovieClip.globalToLocal()` преобразует координаты области действия в координаты экземпляра клипа. Он сообщает, где в области действия находится определенная точка (относительно точки регистрации экземпляра клипа). В аргументе объект вы должны указать имя объекта, который содержит свойства `x` и `y`. Этот аргумент создается для хранения значений координат области действия, которые должны быть преобразованы в координаты экземпляра клипа, но он не может определять положение клипа. Преобразование основывается на системе координат клипа, указанного в аргументе `экземпляр_клипа`.

Пример

Рассмотрим приведенный сценарий построчно, чтобы понять механизм преобразования.

Создадим новый объект `pnt`:

```
pnt=new Object();
```

Присвоим его свойствам `x` и `y` значения текущих координат `x` и `y` данного экземпляра клипа.

```
pnt.x=this._x;
pnt.y=this._y;
```

Выведем текущие значения `x` и `y` в текстовом поле. Они отражают глобальные координаты клипа или его положение относительно области действия.

```
_root.txt1="Глобальные координаты: "+this._x+", "+this._y;
```

Выполним преобразование глобальных координат объекта `pnt` в локальные.

```
this.globalToLocal(pnt)
```

Выведем локальные координаты текущего клипа.

```
_root.txt2="Локальные координаты: "+pnt.x+", "+pnt.y;
```

Поскольку свойства были заданы как координаты положения клипа, преобразование глобальных координат в локальные выдает координаты (0;0) или координаты точки регистрации этого клипа.

MovieClip.gotoAndPlay (5+)

Синтаксис `экземпляр_клипа.gotoAndPlay(номер_кадра)`
 `экземпляр_клипа.gotoAndPlay("метка_кадра")`

Этот метод переводит головку воспроизведения указанного фильма или экземпляра клипа (аргумент `экземпляр_клипа`) в определенный кадр его временной шкалы и начинает воспроизведение с этого места. Вы можете определить нужный кадр, указав либо его номер (целое число), либо метку (строка).

MovieClip.gotoAndStop (5+)

Синтаксис `экземпляр_клипа.gotoAndStop(номер_кадра)`
 `экземпляр_клипа.gotoAndStop("метка_кадра")`

Этот метод переводит головку воспроизведения указанного фильма или экземпляра клипа (аргумент `экземпляр_клипа`) в определенный кадр его временной шкалы и останавливает воспроизведение в этом месте. Вы можете определить нужный кадр, указав либо его номер (целое число), либо метку (строка).

MovieClip.hitTest (5+)

Синтаксис `экземпляр_клипа.hitTest(л.у, значение)`
 `экземпляр_клипа.hitTest("путь")`

Этот метод проверяет, пересекается ли экземпляр_клипа с некоторой координатной точкой или другим экземпляром клипа, и возвращает соответствующее значение типа `Boolean`. Чтобы проверить наличие пересечения с точкой, необходимо указать ее координаты `x` и `y` и ввести логическое значение в аргументе `значение`. Если этот аргумент имеет значение `false` (это значение используется по умолчанию), пересечение будет определяться с учетом обрамления указанного экземпляра клипа. Если аргумент равен `true`, при определении пересечения будут учитываться только пиксели изображения экземпляра клипа. Аргумент `значение` следует задавать равным `true` при работе с клипами неправильной формы или с теми, в которых есть большие пустые участки между изображениями. Чтобы установить наличие пересечения экземпляра клипа с другим клипом, необходимо указать только путь к этому клипу как строковое значение.

Пример

В приведенном примере метод `MovieClip.hitTest()` используется для проверки наличия пересечения текущего клипа с экземпляром клипа `border`. Если они пересекаются (`true`), в поле `display` появится надпись "Пересечение"; в противном случае надпись будет отсутствовать.

```
onClipEvent(enterFrame){
    if(this.hitTest("_root.border")){
```

```

        _root.display="Пересечение";
    }else _root.display="";
}

```

MovieClip.loadMovie (5+)

Синтаксис *экземпляр_клипа.loadMovie("URL", "метод_передачи")*

Этот метод выполняет те же функции, что и действие `loadMovie()`. Единственное его отличие заключается в том, что он не принимает аргумент `уровень`. Вместо этого метод объекта `MovieClip` загружает SWF-файл, на который указывает `URL`, в экземпляре `клипа`. В разделе, посвященном действию `loadMovie()`, вы найдете соответствующий пример, а также узнаете о том, как можно передавать переменные при помощи необязательного аргумента `метод_передачи`.

MovieClip.loadVariables (5+)

Синтаксис *экземпляр_клипа.loadVariables("URL", "метод_передачи")*

Этот метод выполняет те же функции, что и действие `loadVariables()`. Его основное отличие заключается в том, что он не принимает аргумент `уровень`, а вместо этого загружает переменные, найденные по указанному `URL`, в экземпляре `клипа`. В разделе, посвященном действию `loadVariables()`, вы найдете соответствующий пример, а также узнаете о передаче переменных при помощи необязательного аргумента `метод_передачи`.

MovieClip.localToGlobal (5+)

Синтаксис *экземпляр_клипа.localToGlobal(объект)*

Метод `MovieClip.localToGlobal` преобразует координаты экземпляра клипа в координаты области действия. Он сообщает, где в экземпляре клипа находится определенная точка относительно левого верхнего угла области действия. В аргументе объект вы должны указать имя объекта, который содержит свойства `x` и `y`. Этот аргумент создается для хранения значений координат, которые должны быть преобразованы в координаты области действия или глобальные координаты, но он не может определять положение клипа. Преобразование выполняется относительно системы координат указанного экземпляра клипа.

Пример

Рассмотрим приведенный ниже сценарий построчно, чтобы понять механизм преобразования.

Создадим новый объект `pnt`.

```
pnt=new Object();
```

Присвоим его свойствам `x` и `y` значения текущих координат `x` и `y` данного экземпляра клипа.

```
pnt.x=0;
pnt.y=0;
```

Выведем текущие значения `x` и `y` в текстовом поле. Они отражают глобальное положение клипа (положение в области действия).

```
_root.txt1="Глобальные координаты: "+this._x+","+this._y;
```


Выполним преобразование локальных координат объекта `pnt` в глобальные.

```
this.localToGlobal(pnt)
```

Выведем локальные координаты текущего клипа.

```
_root.txt2="Локальные координаты: "+pnt.x+", "+pnt.y;
```

Поскольку свойства были заданы как координаты точки (0;0), преобразовав локальные координаты в глобальные, получаем координаты (200; 130), то есть координаты положения клипа в области действия.

MovieClip.nextFrame (5+)

Синтаксис `экземпляр_клипа.nextFrame()`

Метод `MovieClip.nextFrame()` переводит головку воспроизведения указанного экземпляра клипа или фильма (аргумент `экземпляр_клипа`) на один кадр вперед по временной шкале и останавливает воспроизведение.

MovieClip.play (5+)

Синтаксис `экземпляр_клипа.play()`

Метод `MovieClip.play()` применяется для воспроизведения указанного экземпляра клипа или фильма.

MovieClip.prevFrame (5+)

Синтаксис `экземпляр_клипа.prevFrame()`

Метод `MovieClip.prevFrame()` переводит головку воспроизведения указанного экземпляра клипа или фильма (аргумент `экземпляр_клипа`) на один кадр назад по временной шкале и останавливает воспроизведение.

MovieClip.removeMovieClip (5+)

Синтаксис `экземпляр_клипа.removeMovieClip()`

Этот метод объекта `MovieClip` выполняет те же функции, что и действие `removeMovieClip()`. С его помощью из области действий удаляется любой экземпляр клипа, созданный посредством действий `MovieClip.attachMovie()` и `MovieClip.duplicateMovie()`. Метод не принимает аргумент `имя_клипа`, вместо которого следует указывать путь к экземпляру `клипа`, чтобы удалить его с области действия.

Пример

В данном сценарии клип `alert` удаляется из текущего фильма.

```
onClipEvent(mouseUp){
    if(this.hitTest(_root._xmouse, _root._ymouse, true)){
        _root.alert.removeMovieClip();
    }
}
```

MovieClip.startDrag (5+)

Синтаксис `экземпляр_клипа.startDrag(блокирование, влево, вверх, вправо, вниз)`

Как и действие `startDrag()`, метод `MovieClip.startDrag()` делает клип, указанный в аргументе `экземпляр_клипа`, перемещаемым, позволяя перетаскивать его мышью на области действия. Аргумент `блокирование` содержит значение типа `Boolean`, которое определяет, должна ли точка регистрации клипа при первом щелчке мышью переместиться и совпасть с указателем мыши (`true`) или остаться в своем

исходном положении (`false`). Аргументы влево, вверх, вправо, вниз вводятся как целые числа и определяют координаты границ перемещения клипа. В главе 16 вы найдете пример использования действия `startDragO` для создания бегунка.

MovieClip.stop (5+)

Синтаксис `экземпляр_клипа.stop()`

Метод `MovieClip.stopO` позволяет остановить воспроизведение указанного экземпляра клипа или фильма.

MovieClip.stopDrag (5+)

Синтаксис `экземпляр_клипа.stopDrag()`

Как и действие `stopDrag()`, метод `MovieClip.stopDrag()` отменяет возможность перемещения клипа в текущем фильме.

Пример См. описание действия `startDragO`.

MovieClip.unloadMovie (5+)

Синтаксис `экземпляр_клипа.unloadMovie()`

Этот метод позволяет выгрузить SWF-фильм, доступ к которому был получен с помощью действия или метода `MovieClip.loadMovie()`.

MovieClip.valueOf (5+)

Синтаксис `экземпляр_клипа.valueOf()`

Данный метод возвращает полный путь к указанному клипу, используя для его записи точечный синтаксис.

Методы рисования объекта MovieClip (6)

Методы рисования для объекта `MovieClip` появились во Flash MX. Они предназначены для динамического рисования клипов с помощью ActionScript. Дополнительную информацию об этих методах можно получить, обратившись к панели References, вызвать которую можно при помощи комбинации клавиш Shift+F1, или в документации к Flash MX.

`MovieClip.beginFill()` задает характеристики заливки и окрашивает выбранным цветом клип.

`MovieClip.createEmptyMovieClip()` создает новый пустой клип.

`MovieClip.curveTo()` создает кривую, имеющую характеристики (сплошная, пунктирная или какая-то другая, толщина, цвет и др.), установленные на текущий момент.

`MovieClip.endFill()` применяет заливку, использованную при последнем вызове метода `beginFillO`.

`MovieClip.lineStyle()` определяет характеристики линий.

`MovieClip.lineTo()` создает прямую линию, имеющую характеристики, установленные на текущий момент.

`MovieClip.moveTo()` задает исходную точку для создания линий или кривых.

MovieClip.beginFill (6)

Синтаксис `экземпляр_клипа.beginFill(rgb,alpha)`

Этот метод применяется для заполнения цветом экземпляра клипа, указанного в аргументе `экземпляр_клипа`. Для заполняемого цвета необходимо указать два следующих элемента: шестнадцатеричное RGB-значение в формате `0xrrggbb` (аргумент `rgb`) и значение от 0 до 100, определяющее степень прозрачности заливки (аргумент `alpha`).

MovieClip.createEmptyMovieClip (6)

Синтаксис `экземпляр_клипа1.createEmptyMovieClip("экземпляр_клипа2",уровень)`

Этот метод используется на начальном этапе создания клипа с помощью ActionScript. Посредством метода `createEmptyMovieClip()` на временной шкале указанного клипа (аргумент `экземпляр_клипа1`) создается новый клип. При этом необходимо указать два аргумента: имя создаваемого экземпляра клипа (аргумент `экземпляр_клипа2`) и глубину уровня (аргумент `уровень`). На одном уровне главного фильма может быть только один клип, поэтому следует указывать незапятнанный уровень.

MovieClip.curveTo (6)

Синтаксис `экземпляр_клипа.curveTo(упрх,упру,х,у)`

Метод `curveTo()` предназначен для создания в клипе кривых и дуг. Кривая линия рисуется, начиная с текущей точки, определенной последним вызовом метода `MovieClip.lineTo()`, `MovieClip.curveTo()` или `MovieClip.moveTo()`. Если не указаны иные координаты, кривая начинается в точке с координатами (0;0). Чтобы нарисовать кривую, нужно указать четыре следующих элемента: координаты `x` и `y` (`упрх`, `упру`) для управляющей точки, с помощью которой кривой придается ее форма, и координаты (`x` и `y`) для точки окончания кривой. Кривая будет иметь такие же характеристики, какие были установлены при последнем вызове метода `MovieClip.lineTo()`.

MovieClip.endFill (6)

Синтаксис `экземпляр_клипа.endFill()`

Этот метод применяет заливку, выбранную при последнем вызове метода `MovieClip.beginFill()`, к указанному экземпляру клипа.

MovieClip.lineStyle (6)

Синтаксис `экземпляр_клипа.lineStyle(толщина,rgb,alpha)`

С помощью этого метода можно задать характеристики линий в ваших векторных изображениях. Толщина линии (аргумент `толщина`) определяется как целое число от 0 до 255. В аргументе `rgb` задается шестнадцатеричное RGB-значение цвета (в формате `0xrrggbb`), которое определяет цвет линии. Степень прозрачности (аргумент `alpha`) определяется целым числом от 0 до 100. Все характеристики линий фиксируются и сохраняются до нового вызова метода `MovieClip.lineStyle()`.

MovieClip.lineTo (6)

Синтаксис экземпляр_клипа.lineTo(x,y)

Метод `MovieClip.lineTo()` предназначен для создания в указанном экземпляре клипа прямых линий. Прямая линия начинается в текущей точке, определенной последним вызовом метода `MovieClip.lineTo()`, `MovieClip.curveTo()` или `MovieClip.moveTo()`. Если не указаны иные координаты, линия начинается в точке (0;0). Для точки завершения необходимо задать соответствующие координаты в аргументах `x` и `y`. Итоговая линия начинается в текущей точке и заканчивается в точке с координатами (x;y). Линия имеет такие же характеристики, какие были установлены при последнем вызове метода `MovieClip.setStyle()`.

MovieClip.moveTo (6)

Синтаксис экземпляр_клипа.moveTo(x,y)

С помощью метода `MovieClip.moveTo` задается исходная точка для создания линий или кривых. Координаты этой точки `X` и `Y` (аргументы `x` и `y`) задаются относительно точки регистрации указанного экземпляра клипа. После того как была задана исходная точка, текущая точка всегда будет определяться координатами точки завершения, установленными при последнем вызове метода `lineTo()` или `curveTo()`.

Пример

В этом примере с помощью `ActionScript` динамически воспроизводится эффект сумерек. В первом блоке создается черный квадрат, названный `sky` (небо). Обратите внимание на то, что если форма рисуется только с помощью метода `MovieClip.beginFill()`, в ней нет обрамления.

```
_root.createEmptyMovieClip("sky",1);
_root.sky.beginFill(0x000000);
_root.sky.moveTo(0,0);
_root.sky.lineTo(0,200);
_root.sky.lineTo(200,200);
_root.sky.lineTo(200,0);
_root.sky.lineTo(0,0);
```

Во втором блоке рисуются звезды. Они произвольно размещаются в прямоугольнике размером 190x190 и разворачиваются на произвольное количество градусов в пределах от 0 до 45 градусов.

```
for(i=2; i<23; i++){
    var ptX=Math.floor(Math.random()*191);
    var ptY=Math.floor(Math.random()*191);
    var spin=Math.floor(Math.random()*46);
    _root.createEmptyMovieClip("star"+i,i);
    with(eval("_root.star"+i)){
        rotation=spin;
        lineStyle(2,0xcccc66,100);
        beginFill(0xcccc00);
        moveTo(ptX+7,ptY-7);
       .lineTo(ptX,ptY-8);
       .lineTo(ptX+5,ptY-13);
       .lineTo(ptX+4,ptY-20);
       .lineTo(ptX+10,ptY-20);
       .lineTo(ptX+15,ptY-13);
       .lineTo(ptX+20,ptY-8);
    }
}
```

```

    lineTo(ptX+13,ptY-7);
    lineTo(ptX+10,ptY);
    lineTo(ptX+7,ptY-7);
}
}

```

В последнем блоке сценария создается полумесяц. Для этого мы воспользовались методом `MovieClip.curveTo()`. Контур месяца пурпурный, а заливка светло-пурпурная.

```

_root.createEmptyMovieClip("moon");
_root.moon.linestyle(8,0x663366,100);
_root.moon.beginFill(0x666699);
_root.moon.moveTo(40,50);
_root.moon.curveTo(116,12,150,40);
_root.moon.curveTo(190,73,160,160);
_root.moon.curveTo(162,109,126,74);
_root.moon.curveTo(85,35,40,50);

```

Компоненты объекта Object (5+)

Синтаксис `new Object()`

С помощью объекта `Object` создаются объекты для временного хранения информации, то есть пользовательские объекты. Для создания пользовательского объекта применяется функция `new Object()`. Примеры создания таких объектов вы найдете в описаниях методов `Color.setTransform()`, `MovieClip.localToGlobal()` и `Sound.setTransform`.

Объект Sound (5+/6)

Синтаксис `new Sound()`
 `new Sound("target")`

Объект `Sound` предназначен для управления воспроизведением звука во Flash-фильмах. Для создания этого объекта воспользуйтесь функцией `new Sound()`. С ее помощью вы получите глобальный объект `Sound`, именуемый `globalSound`. Такой объект может контролировать все звуки фильма.

```
globalSound=new Sound();
```

В нижеприведенном выражении создается звуковой объект `bkgdSound` для воспроизведения звуков фильма, находящегося на уровне 1.

```
bkgdSound=new Sound("_level1");
```

В следующем выражении создается звуковой объект `clipSound` для воспроизведения звуков клипа `note`.

```
clipSound=new Sound("_root.note");
```

Создав объект `Sound`, вы можете воспользоваться его методами для регулировки громкости и позиции панорамирования звука. Кроме того, из библиотеки вашего фильма можно экспортировать отдельные звуки и связать их с определенными объектам `Sound` при помощи метода `attachSound()`. Эта техника обеспечивает управление всеми параметрами воспроизведения звука, в том числе определяет часть фильма, в которой начинается воспроизведение звука, и количество повторений последнего.

События

`Sound.onSoundComplete` возникает при окончании воспроизведения звука.

`Sound.onLoad` возникает при загрузке звука из внешнего источника во Flash-фильм.

Методы

`Sound.attachSound()` связывает звук, извлеченный из библиотеки, с объектом `Sound`.

`Sound.getPan()` возвращает значение позиции панорамирования для указанного объекта `Sound`.

`Sound.getTransform()` возвращает доли (в процентах) стереоканалов звука, которые на данный момент воспроизводятся в левом и правом динамиках.

`Sound.getVolume()` возвращает уровень громкости для указанного объекта `Sound`.

`Sound.setPan()` присваивает значение позиции панорамирования указанному объекту `Sound`.

`Sound.setTransform()` задает доли (в процентах) стереоканалов звука, воспроизводимых в левом и правом динамиках.

`Sound.setVolume()` задает уровень громкости для указанного объекта `Sound`.

`Sound.start()` воспроизводит звук, связанный с указанным объектом `Sound`.

`Sound.stop()` выключает звук, связанный с указанным объектом `Sound`.

Свойства:

`Sound.duration` возвращает полную возможную длительность воспроизведения указанного звука в миллисекундах.

`Sound.position` возвращает время в миллисекундах, в течение которого в действительности происходило воспроизведение звука.

attachSound (5+)

Синтаксис `объектSound.attachSound("идентификатор")`

Метод `Sound.attachSound()` служит для связывания звука, который вы экспортировали из библиотеки, с определенным объектом `Sound`. Для того чтобы экспортировать звук из библиотеки фильма, выберите его на панели Library, а затем выберите команду Linkage меню Options. Установите флажок Export for ActionScript и присвойте звуку имя в поле Identifier. При вызове метода `Sound.attachSound()` это имя указывается в аргументе идентификатор как строковое значение (то есть в кавычках). Таким образом, звук, который вы экспортировали из библиотеки, будет связан с объектом `Sound`. Теперь с помощью объекта `Sound` вы можете воспроизвести звук, остановить его воспроизведение, а также выполнить с ним другие действия. Если в фильме на разных уровнях используется много звуковых объектов, то, чтобы управлять звуками каждого уровня, необходимо указывать соответствующие аргументы.

Пример

В приведенном примере звук `brass` связывается с новым звуковым объектом `fanfare`, после чего этот звук воспроизводится.

```
fanfare=new Sound();
fanfare.attachSound("brass");
fanfare.start();
```

Sound.duration (6)

Синтаксис объект.Sound.duration

С помощью этого свойства извлекается общая продолжительность воспроизведения звука (в миллисекундах), ассоциированного с указанным объектом `Sound`.

Пример

В следующем примере два свойства, `position` и `duration`, используются для создания индикатора воспроизведения, отслеживающего, какая часть звука уже воспроизведена. По мере воспроизведения звука индикатор воспроизведения (первоначально имеющий свойство `_xscale` с нулевым значением) приближается к 100.

```
onClipEvent(load){
    bar_mc.xscale=0;
}
onClipEvent(enterFrame){
    played=_root.riff.position;
    total=_root.riff.duration;
    bar_mc._xscale=100*(played/total);
}
```

Sound.getPan (5+)

Синтаксис объект.Sound.getPan()

С помощью этого метода можно извлечь значение позиции панорамирования для указанного объекта `Sound`. Значение позиции панорамирования объекта измеряется в пределах от -100 (крайняя левая) до 100 (крайняя правая).

Sound.getTransform

Синтаксис объект.Sound.getTransform(объект1)

Используя этот метод, мы назначаем свойства `l1`, `lR`, `r1` и `rR`, которые содержатся в объекте1, звукам в указанном объекте `Sound` или звукам, ассоциированным с этим объектом. Эти свойства содержат доли (в процентах) левого и правого каналов стереозвука для соответствующих динамиков, заданные методом `Sound.setTransform()` для объекта1.

Sound.getVolume (5+)

Синтаксис объект.Sound.getVolume()

Этот метод возвращает текущий уровень громкости для указанного объекта `Sound`. Возвращенные значения находятся в пределах от 0 до 100.

Пример

Приведенная функция устанавливает уровень громкости звуков в фильме равным 10, если они воспроизводятся с уровнем громкости больше 10, и устанавливает максимальную громкость для звуков, громкость которых меньше 10.

```
function volToggle(obj){
  if(obj.getVolume()<=10){
    obj.setVolume(100);
  }else obj.setVolume(10);
}
```

Sound.onLoad (6)

Синтаксис объектSound.onLoad=функция

Событие `Sound.onLoad` возникает тогда, когда звук, ассоциированный с объектом `Sound`, полностью загрузится в фильм посредством метода `Sound.loadSound()`. Чтобы использовать это событие для запуска дополнительных сценариев, задайте функцию, которая будет выполнять нужные выражения.

Пример

Это событие можно использовать при создании анимации, отображающей процесс загрузки фильма.

```
track1.onLoad=function(){
  track1.start(0,2);
  status.nextFrame();
}
```

Звук `track1` воспроизводится с момента начала загрузки звукового файла два раза, а клип `status` переходит к следующему кадру.

Sound.onSoundComplete (6)

Синтаксис объектSound.onSoundComplete=функция

Это событие происходит тогда, когда заканчивается воспроизведение звука, связанного с указанным объектом `Sound`. Если объект воспроизводится циклически, событие `Sound.onSoundComplete` не будет возникать до тех пор, пока цикл не будет завершен. Чтобы использовать это событие для запуска дополнительных сценариев, задайте функцию, которая будет выполнять нужные выражения.

Пример

Возможности применения этого события очень разнообразны. Например, его можно использовать для воспроизведения следующей по списку песни.

```
track1.onSoundComplete=function(){
  track2.start();
}
```

По окончании воспроизведения звука `track1` начнет воспроизводиться звук `track2`.

Sound.position

Синтаксис объектSound.position

Данное свойство возвращает время (в миллисекундах), в течение которого происходило воспроизведение звука, связанного с указанным объектом `Sound`. Если объект `Sound` должен воспроизводиться циклически, его свойство `position` будет возвращать ноль в начале каждого повтора цикла. Пример использования этого свойства вы найдете в описании свойства `Sound.duration`.

Sound.setPan (5+)*Синтаксис* объектSound.setPan(позиция_панорамирования)

С помощью этого метода задается значение позиции панорамирования звука или звуков, связанных с указанным объектом Sound. Значение позиции панорамирования звука может находиться в пределах от -100 (крайняя левая) до 100 (крайняя правая). Если задать крайнюю левую позицию панорамирования, звук будет воспроизводиться только левым динамиком. И наоборот, крайняя правая позиция означает воспроизведение только в правом динамике.

Пример

При каждом нажатии кнопки, связанной с этим сценарием, воспроизведение звука объекта `moveSnd` постепенно переводится на левый динамик.

```
on(press){
    movSnd.setPan(moveSnd.getPan()-10);
}
```

В данном сценарии мы неявно задаем значение позиции панорамирования. Мы получаем ее текущее значение при помощи метода `getPan()`. После этого метод `setPan()` задает новое значение позиции панорамирования, вычитая из значения старой число 10. С такой же легкостью можно переводить воспроизведение звука на правый динамик, прибавляя 10 к текущему значению позиции панорамирования.

setTransform*Синтаксис* объектSound.setTransform(объект1)

Метод `Sound.setTransform()` относится к наиболее интересным методам ActionScript. Он позволяет динамически перераспределять компоненты левых и правых каналов стереозвуча в левом и правом динамике. Используя этот метод, можно полностью перестроить стереосостав звука. Например, можно воспроизводить все звуки левого канала в правом динамике (так, чтобы ни один из них не воспроизводился в левом), а большую часть звуков правого канала — в левом динамике и только 25 % из них — в правом.

Чтобы воспользоваться этим методом, необходимо сначала создать объект1 -- пользовательский объект со свойствами `ll`, `lr`, `rl` и `rr`. Эти свойства отвечают за распределение компонентов левых и правых каналов в левом и правом динамиках. Описание каждого свойства дано в табл. Б.3.

Таблица Б.3. Свойства объекта 1

Свойство	Диапазон	Описание
ll	0—100	Доля звуков левого канала (в процентах), предназначенная для воспроизведения в левом динамике
lr	0—100	Доля звуков правого канала (в процентах), предназначенная для воспроизведения в левом динамике
rr	0—100	Доля звуков правого канала (в процентах), предназначенная для воспроизведения в правом динамике
rl	0—100	Доля звуков левого канала (в процентах), предназначенная для воспроизведения в правом динамике

Создав объект1, присвойте значение каждому из четырех его свойств и передайте их значения указанному объекту Sound при помощи метода `Sound.setTransform()`. Пример использования этого метода вы найдете в главе 22.

Sound.setVolume (5+)

Синтаксис объектSound.setVolume(уровень_громкости) •

Этот метод устанавливает уровень громкости звука или звуков, связанных с указанным объектом Sound. Уровень громкости может устанавливаться в пределах от 0 до 100. Если уровень громкости установить равным 0, звук будет воспроизводиться, но не будет слышен. Можно подумать, что воспроизведение остановилось. Хотя Macromedia и не документировала этот факт, но фактически максимального уровня громкости для звуков не существует. Однако, если до импортирования во Flash-фильм звуки были должным образом нормализованы, установка уровня громкости больше 100 единиц может вызвать их искажение.

Пример

В приведенном сценарии громкость звука объекта `movSnd` постепенно уменьшается при каждом нажатии кнопки.

```
on(press){
    movSnd.setVolume(movSnd.getVolume()-10);
```

Вместо того чтобы явно задать уровень громкости, мы извлекли его текущее значение при помощи метода `getVolume()`, а после этого посредством метода `Sound.setVolume()` установили новое значение уровня громкости, уменьшив текущее на 10. В этом сценарии можно с такой же легкостью усиливать громкость звука, прибавляя 10 к текущему значению уровня громкости. См. также пример в описании метода `Sound.getVolume()`.

Sound.start (5+)

Синтаксис объектSound.start(количество_секунд,цикл)

Этот метод применяется для воспроизведения звука, связанного с объектом Sound при помощи метода `Sound.attachSound()`. Метод `Sound.start()` принимает два необязательных аргумента: `количество_секунд` и `цикл`. В первом аргументе указывается количество секунд, через которое следует начать воспроизведение звука. Если в этом аргументе указать, например, число 1,5, то воспроизведение звука начнется через полторы секунды после начала воспроизведения звукового файла. По умолчанию данный аргумент имеет значение 0, при котором звук проигрывается с самого начала. В аргументе `цикл` вы задаете, сколько раз звук будет повторяться. Он может иметь значение 1 (значение по умолчанию), 2 и т. д., звук соответственно будет воспроизводиться один раз, два раза и т. д.

Пример

В следующем примере демонстрируется способ применения метода `Sound.start()`. Во-первых, необходимо создать объект Sound и связать с ним звук из библиотеки.

```
groove=new Sound();
groove.attachSound("drums");
```

Затем мы воспроизводим звук один раз с самого начала.

```
groove.start();
```

Вы также можете воспроизвести звук, начиная с 1,8 секунды.

```
groove.start(1.8);
```

Sound.stop(5+)*Синтаксис* `объектSound.stop("идентификатор")`

Этот метод предназначен для отключения звуков в фильме. С его помощью можно задать, воспроизведение каких звуков должно быть остановлено. Если объект `Sound` является глобальным звуковым объектом, в фильме будут отключены все звуки. Если вы хотите отключить звук, связанный с объектом `Sound` при помощи метода `Sound.attachSound()`, укажите его имя (как строку) в аргументе идентификатор.

Пример

Поскольку объект `bkgd` является глобальным звуковым объектом, это выражение остановит воспроизведение всех звуков в фильме.

```
bkgd.stop();
```

В нижеприведенном выражении выключается только звук `drums`, связанный с объектом `groove`.

```
groove.stop("drums")
```

Объект String (5+)*Синтаксис* `new String(выражение)`

Объект `String` преобразовывает указанное выражение в строку и возвращает результат как свойство объекта `String`. Этот объект также может применяться для преобразования различных типов данных в значения типа `string`. Однако такое применение объекта, как правило, не является целесообразным. В документации по ActionScript рекомендуется присваивать строкам значения с помощью литерала (`stringName="text string"`), а для преобразования значений пользоваться функцией `String()`. В первую очередь в объекте `String` важны его методы. Они могут применяться к любым строкам и предоставлять различные возможности манипулирования данными типа `string`.

Методы

`String.charAt()` возвращает символ, занимающий указанную позицию в строке.

`String.concat()` объединяет одну или несколько строк, чтобы создать единую строку.

`String.indexOf()` определяет позицию, в которой искомый элемент встречается первый раз.

`String.lastIndexOf()` определяет позицию, в которой искомый элемент встречается последний раз.

`String.slice()` возвращает указанную часть строки как подстроку, используя положительные и отрицательные целые числа.

`String.substring()` возвращает указанную часть строки как подстроку, используя положительные целые числа.

`String.toLowerCase()` возвращает строку, все символы которой преобразованы в символы нижнего регистра.

`String.toUpperCase()` возвращает строку, все символы которой преобразованы в символы верхнего регистра.

Свойства

`String.length` определяет количество символов в строке.

String.charAt (5+)

Синтаксис строка.charAt(номер_позиции)

Этот метод возвращает символ, номер позиции которого в строке соответствует номеру, указанному в аргументе `номер_позиции`. Нумерация позиций в строке начинается с 0 (0 соответствует первому знаку строки, 1 — второму, 2 — третьему и т. д.).

Пример

В этом сценарии заданную позицию в строке `greeting` занимает символ `G`.

```
greeting="Good afternoon!";
trace(greeting.charAt(0));
```

В следующем сценарии возвращается символ `!`, поскольку строка состоит из 15 символов и именно символ `!` занимает позицию под номером 14.

```
trace(greeting.charAt(greeting.length-1));
```

String.concat (5+)

Синтаксис строка.concat(элемент1,элемент2...элементх)

Метод `String.concat()` можно использовать для добавления в строку символов `элемент1`, `элемент2` и т. д.

Пример

Из строки `mile` в этом сценарии создается строка `milestone`.

```
one="mile";
pair=one.concat("stone");
```

В данном сценарии создается строка `milestones`.

```
trio=one.concat("stone","s");
```

String.indexOf (5+)

Синтаксис строка.indexOf("подстрока",номер_позиции)

Этот метод возвращает номер первой позиции подстроки в строке. В необязательном аргументе `номер_позиции` можно указать целое число. Это число определит номер позиции, с которой начнется поиск подстроки. Аргумент `номер_позиции` может содержать число от 0 (первый символ строки) до значения, равного `string.length-1` (последний символ строки). Если подстрока не найдена, метод возвращает `-1`.

Пример

Приведенная функция извлекает имя пользователя из почтового адреса (`id`) — имя, значащееся перед символом `@`. Возвращенное имя может использоваться в фильме для приветствия или регистрации посетителей Web-узла.

```
function netID(id){
    var at=id.indexOf("@");
    var user=id.substring(0;at);
    return user;
}
```

String.lastIndexOf (5+)

Синтаксис строка.lastIndexOf("подстрока",номер_позиции)

Этот метод возвращает номер последней позиции подстроки в строке. В необязательном аргументе номер_позиции можно указать целое число. Это число определит номер позиции, с которой начнется поиск подстроки. Аргумент номер_позиции может содержать число от 0 (первый символ строки) до значения, равного string.length-1 (последний символ строки). Если подстрока не найдена, метод возвращает -1.

String.length (5+)

Синтаксис строка.length

Это свойство возвращает число символов в строке. Поскольку нумерация символов начинается с 0, длина строки всегда на одно значение больше, чем номер позиции последнего символа. Например, в строке *abc*, длина которой равна 3, номер позиции последнего символа равен 2 (a = 0, b = 1, c = 2). Если в строке есть пробелы, каждый из них считается как один символ.

Пример

В следующем сценарии будет возвращено значение 8.

```
name="Vladimir";
trace(name.length);
```

В следующем сценарии будет возвращено значение 20.

```
name="Berezovskiy Vladimir";
trace(name.length);
```

String.slice (5+)

Синтаксис строка.slice(начальная_позиция,конечная_позиция)

Этот метод возвращает подстроку из строки. Чтобы указать, какая часть строки должна извлекаться, задайте два аргумента: начальная_позиция и конечная_позиция. Аргумент начальная_позиция задает номер позиции первого символа возвращаемой подстроки, а конечная_позиция — номер символа строки, находящийся сразу за последним символом подстроки. Если вы не укажете аргумент конечная_позиция, по умолчанию в нем будет использовано значение длины строки (количество символов в строке). Оба аргумента содержат целые числа. При этом учитывается, что 0 обозначает первый символ строки. Вы можете указывать и отрицательные числа, тогда -1 будет означать последний символ, -2 — предпоследний и т. д.

Пример

Из строки name мы извлекаем имя, фамилию и инициал.

```
name="Sirko D Dog"
firstName=name.slice(0,5);
initial=name.slice(6,7);
lastName=name.slice(8,11);
```

Того же результата можно достичь, используя метод substring(). Разница лишь в том, что этот метод не поддерживает отрицательные числа.

```
name="Sirko D Dog"
firstName=name.substring(0,5);
```

```
initial=name.substring(6,7);
lastname=name.substring(9);
```

Обратите внимание на то, что в примере со строкой `lastName` не указан аргумент `endRef` и по умолчанию используется значение длины строки (`string.length`). В сценарии также можно использовать отрицательные числа.

```
firstName=name.slice(-11,-6);
initial=name.slice(-5,-4);
lastname=name.slice(-3,11);
```

String.toLowerCase (5+)

Синтаксис строка.toLowerCase()

Этот метод преобразовывает все символы строки в символы нижнего регистра. Учтите, что сама строка при этом не меняется, а создается копия строки, где все символы представлены символами нижнего регистра. Пример использования этого метода вы найдете в описании метода `String.toUpperCase()`.

String.toUpperCase (5+)

Синтаксис: строка.toUpperCase()

Данный метод преобразовывает все символы строки в символы верхнего регистра. Учтите, что сама строка при этом не меняется, а создается копия строки, где все символы представлены символами верхнего регистра.

Пример

В этом сценарии все символы строкового значения переменной `text` преобразуются сначала в символы верхнего, а затем — нижнего регистра. Символ `&` остается без изменений.

```
text="UPPER & lower";
format1=text.toUpperCase();
format2=text.toLowerCase();
```

Переменная `format1` получает строковое значение `UPPER & LOWER`, а переменная `format2` — `upper & lower`.

Объект TextField (6)

Объект `TextField` позволяет манипулировать характеристиками полей, предназначенных для размещения динамического текста и текста ввода с помощью `ActionScript`. Для этого текстовому полю следует присвоить уникальное имя экземпляра. Щелкните на текстовом поле и откройте панель `Properties (Window ▸ Properties)` или комбинация клавиш `Cmd/Ctrl+F3`. Введите имя в поле `Instance Name` панели `Properties`. Это имя и будет именем экземпляра поля.

Свойства

`TextField.hscroll` задает и возвращает текущее положение бегунка прокрутки горизонтали.

`TextField.maxhscroll` возвращает максимальное положение бегунка прокрутки по горизонтали.

`TextField.maxvscroll` возвращает максимальное положение бегунка прокрутки по вертикали.

`TextField.scroll` задает и возвращает текущее положение бегунка прокрутки по вертикали.

TextField.hscroll (6)

Синтаксис `текстовое_поле.hscroll=выражение`

С помощью этого свойства можно как задавать, так и получать значение, соответствующее положению бегунка прокрутки по горизонтали для объекта `TextField`. Если значение свойства `hscroll` равно 0, текст в поле выравнивается по левому краю. Пример использования данного свойства вы найдете в описании свойства `scroll`.

TextField.maxhscroll (6)

Синтаксис `текстовое_поле.maxhscroll=выражение`

Это свойство возвращает значение максимально возможной правой позиции символов в текстовом поле. Если символ текста в этой позиции поля отсутствует, свойство возвращает значение 0. Вы не можете устанавливать значение этого свойства.

Пример

В этом сценарии текст в поле `line` переместится в его (поля) конец, то есть последний символ последнего слова будет занимать крайнюю правую позицию поля.

```
on(press){
    edge=line.maxhscroll;
    line.hscroll=edge;
}
```

TextField.maxscroll (6)

Синтаксис `текстовое_поле=выражение`

Это свойство объекта `TextField` возвращает номер верхней видимой строки, при котором в поле видна самая нижняя строка. Если в текстовом поле видно 5 строк, а всего их в нем 20, свойство `maxscroll` будет иметь значение 16, потому что если внизу видна строка 20, в самом верху поля будет видна строка 16. Вы не можете устанавливать значение этого свойства.

TextField.scroll (6)

Синтаксис `текстовое_поле.scroll="выражение"`

С помощью этого свойства объекта `TextField` можно как устанавливать, так и получать номер самой верхней строки, видимой в текстовом поле. Свойство применяется для создания интерактивных и анимированных полей с прокруткой.

Пример

Любой из приведенных примеров можно поместить в обработчик события `on(MouseEvent)` или `onClipEvent()`. Кроме того, вы можете воспользоваться объектом `MovieClip` или `Button` для прокрутки содержимого поля. Чтобы прокрутить поле вверх, воспользуйтесь следующим сценарием.

```
verField.scroll+=1;
```

Чтобы прокрутить поле вниз — другим сценарием.

```
verField.scroll-=1;
```

Для прокрутки поля в горизонтальном направлении (вправо) воспользуйтесь сценарием

```
horizField.hscroll+=10;
```

Чтобы восстановить исходное положение текста, воспользуйтесь сценарием

```
horizField.hscroll=0;
```

Значения кодов клавиш для объекта Key

В табл. Б.4 содержится полный список значений кодов клавиш для объекта Key.

Таблица Б.4. Значения кодов клавиш

Буквенные клавиши	Код	Цифровые клавиши	Код
A	65	0	48
B	66	1	49
C	67	2	50
D	68	3	51
E	69	4	52
F	70	5	53
G	71	6	54
H	72	7	55
I	73	8	56
J	74	9	57
K	75	Клавиши цифровой клавиатуры	Код
L	76	клавиша 0	96
M	77	клавиша 1	97
N	78	клавиша 2	98
O	79	клавиша 3	99
P	80	клавиша 4	100
Q	81	клавиша 5	101
R	82	клавиша 6	102
S	83	клавиша 7	103
T	84	клавиша 8	104
U	85	клавиша 9	105
V	86	знак умножения (*)	106
W	87	знак сложения (+)	107
X	88	Enter	13
Y	89	знак вычитания (-)	109
Z	90	знак десятичной дроби (.)	110
		знак деления (/)	111

Функциональные клавиши	Код	Прочие клавиши	Код
F1	112	Пробел	32
F2	113	Page Up	33
F3	114	Page Down	34
F4	115	End	35
F5	116	Home	36
F6	117	Стрелка влево	37
F7	118	Стрелка вверх	38
F8	119	Стрелка вправо	39
F9	120	Стрелка вниз	40
F10	121	Insert	45
F11	122	Delete	46
F12	123	Help	47
Прочие клавиши	Код	Num Lock	144
Backspace	8	;;	186
Tab	9	=+	187
Clear (Macintosh)	12	-	189
Enter	13	/?	191
Shift	16	'-	192
Control	17	{	219
Alt	18	\	220
Caps Lock	20	}	221
Esc	27	"	222

Устаревшие термины

С выходом Flash MX элементами ActionScript, список которых приведен в левом столбце табл. БД перестали пользоваться, отдав предпочтение более эффективной новой терминологии. Эти элементы все еще функционируют, по лучше по возможности избегать их, чтобы в будущем не возникало проблем с совместимостью фильмов.

Таблица Б.5. Устаревшие термины и их новые эквиваленты

Устаревший термин	Новый термин	Устаревший термин	Новый термин
<> (неравенство)	Оператор !=	_highquality	Свойство _quality
add	Оператор +	ifframeLoaded	Свойство _framesloaded
and	&&	int	Math.floor
call	function	le	<=
chr	String.fromCharCode	length	String.length
eq	==	lt	<
ge	>=	maxscroll	textField.maxscroll

Устаревший термин	Новый термин
gt	>
mbchr	String.fromCharCode
mblength	String.length
mbord	String.fromCharCode
mbsubstring	String.substr
ne	!= (неравенство)
not	! (логическое НЕ)

Устаревший термин	Новый термин
or	
ord	Объект String
random	Math.random
scroll	textField.scroll
substring	String.substring
tellTarget	with
toggleHighQuality	Свойство _quality

Алфавитный указатель

А

- Автономный Flash-файл, 600
- Аналоговая звукозапись, 387
- Анимация, 168
 - кадровый метод, 168
 - построение промежуточных кадров, 168
 - тип motion path tweening, 183
 - тип motion tweening, 178
 - тип shape tweening, 172, 181, 483
- Аппроксимация кривой, 104
- Аудиоэффект, 405

Б

- База данных, 350
- Безопасная зона действий, 587
- Библиотека, 126
 - внешняя, 133
 - звуковая встроенная, 404
 - кнопок, 248
 - команд языка ActionScript, 222
 - совместно используемая, 134
- Библиотечный фонд, 134

В

- Видео
 - вложение во Flash, 577
 - импорт, 576
 - связанное, 579
- Видеоклип, 423
- Вращение объекта, 181
- Временная шкала, 46
 - управление, 332
 - управление с помощью ActionScript, 335
- Временное сжатие, 575
- Встроенная звуковая библиотека, 404
- Выборка в звукозаписи, 387
- Выпрямление, 65

Г

- Гибридная технология, 531
- Глубина выборки в звукозаписи, 388
- Головка воспроизведения, 46
- Градиент,
 - линейный, 84
 - радиальный, 85

- Графический элемент
 - анимирование, 341
 - масштабирование, 343
- Группирование, 107

Д

- Действие, 222, 313, 640
 - continue, 640
 - do...while(), 315
 - do...while, 641
 - duplicateMovieClip(), 641
 - for(), 315
 - fscommand(), 234, 239, 643
 - function(), 643
 - getURL(), 234, 237, 352, 644
 - goto(), 235
 - gotoAndPlay(), 644
 - gotoAndStop(), 644
 - if(), 645
 - ifFrameLoaded(), 235
 - loadMovie(), 234, 335, 354, 645
 - loadMovieNum(), 645
 - loadVariables(), 354, 646
 - loadVariablesNum(), 354, 646
 - nextFrame(), 646
 - nextScene(), 646
 - onClipEvent(), 234
 - play(), 236, 647
 - prevFrame(), 647
 - prevScene(), 647
 - printAsBitmap(), 648
 - printAsBitmapNum(), 648
 - printNum(), 648
 - removeMovieClip(), 648
 - setProperty(), 234, 244, 516, 649
 - startDrag(), 649
 - stop(), 236, 649
 - stopAllSounds(), 234, 237
 - stopDrag(), 325, 650
 - switch(), 314
 - tellTarget(), 235, 243
 - toggleHighQuality(), 235
 - unloadMovie(), 234, 651
 - unloadMovieNum(), 651
 - updateAfterEvent(), 651
 - var(), 310

- while(), 315
- with(), 235
- Динамический внешний интерфейс, 350
- Динамический текст
 - имя текстового поля, 96
 - рамка, 96
 - тип строки, 95
 - форматирование HTML, 96

З

- Заказной контроллер, 325
- Задание
 - направления текста, 93
 - положения символа, 92
 - размера шрифта, 91
 - свойств растрового изображения, 104
 - стиля текста, 91
 - тип загрузки URL, 93
 - цвета текста, 92
 - шрифта, 91
 - шрифтов устройства, 94
- Звук
 - аудиоэффект, 405
 - иммерсия, 397
 - импорт, 400
 - многоуровневый эффект, 422
 - нормализация, 396
 - панорамирование, 405
 - параметры, 461
 - эффект Fade In, 439
 - эффект Fade Out, 439
- Звук
 - кнопки, 416
 - потоковый, 402
- Звуковой файл
 - синхронизации, 453
 - тип, 390
 - Sync Track, 453, 462
- Звуковой эффект, 405, 453
 - Fade In, 406
 - Fade Left to Right, 406
 - Fade Out, 406
 - Fade Right to Left, 406
 - Left/Right Channel, 406
- Звукозапись
 - выборка, 387
 - глубина выборки, 388
 - частота выборки, 387

И

- Изображение растровое, 31, 101
- Импорт
 - звука, 400
 - растрового изображения, 102
- Имя связывающее, 432
- Инструмент
 - Arrow, 62
 - Brush, 69
 - Eraser, 78
 - Eyedropper, 77
 - Fill Transform, 75
 - Free Transform, 36
 - Hand, 43
 - Ink Bottle, 73
 - Lasso, 65
 - Line, 65
 - Oval, 67
 - Paint Bucket, 74
 - Pen, 70
 - Pencil, 68
 - Rectangle, 68
 - Subselection, 72, 73
 - Text, 37, 88
 - Zoom, 43
- Интерактивность, 221

К

- Кадр, 46, 169
 - вид на временной шкале, 169
 - действие, 228
 - ключевой, 169
 - пустой ключевой, 169
 - частота, 453, 455
- Калькулятор частоты кадров, 455
- Каталог слоев, 153
- Киоск, 34
- Клип, 122, 190, 423
 - редактирование, 194
 - событие, 232
- Ключевое слово, 305
- Кнопка, 122
 - анимированная, 258
 - библиотека, 248
 - пользовательская, 249
 - предварительный просмотр, 254
 - редактирование, 254
 - сложная, 256
 - тестирование, 254

Кодек Soreson Spark, 575
 Кодирование межкадровое, 575
 Команда

- ActionScript Dictionary, 57
- Antialias, 89
- Bandwidth Profiler, 569
- Break Apart, 66
- Common Libraries, 126
- Convert to Symbol, 125
- Copy Frames, 154
- Deselect All, 63
- Edit Selected, 108
- Edit Symbol, 255
- Enable Simple Buttons, 254
- Flash Support Center, 57
- Flip Horizontal, 116
- Flip Vertical, 116
- Group, 63, 108
- Import, 102
- Keyboard Shortcuts, 56
- Layer, 150
- Lessons, 57
- Paste Frames, 154
- Publish, 569
- Publish Preview, 569
- Publish Settings, 137
- Samples, 57
- Save Panel Layout, 56
- Smooth, 65
- Straighten, 65
- Test Movie, 254
- Test Scene, 254
- Trace Bitmap, 103
- Transform, 113
- Tutorials, 57
- Work Area, 44

 Комментарий, 303

- ActionScript, 366

 Конверт, 118
 Консорциум WWW, 556
 Константа, 304
 Контрольная точка, 369

M

Межзнаковый интервал, 92
 Межкадровое кодирование, 575
 Межстрочное расстояние, 95
 Меню

- Control, 41
- Edit, 40
- File, 40

- Insert, 41
- Modify, 41
- Text, 41
- View, 41
- Window, 41

Метод, 335

- сжатия Variable Bit Rate (VBR), 575
- Array.concat(), 668
- Array.pop(), 669
- Array.push(), 669
- Array.shift(), 670
- Array.slice(), 670
- Array.unshift(), 671
- Color.getRGB, 672
- Color.getTransform(), 672
- Color.setRGB, 672
- Color.setTransform(), 672
- Date.getDate(), 675
- Date.getDay(), 675
- Date.getFullYear(), 675
- Date.getHours(), 675
- Date.getMilliseconds(), 675
- Date.getMinutes(), 676
- Date.getMonth(), 676
- Date.getTime(), 676
- Date.getYear(), 676
- Date.setDate(), 676
- Date.setFullYear(), 676
- Date.setHours(), 676
- Date.setMilliseconds(), 677
- Date.setMinutes(), 677
- Date.setMonth(), 677
- Date.setSeconds(), 677
- Date.setTime(), 677
- Date.setYear(), 677
- Date.toString(), 677
- globalToLocal(), 685
- Key.getAscii(), 678
- Key.getCode(), 679
- Key.isDown(), 679
- Math.abs(), 680
- Math.ceil(), 680
- Math.floor(), 680
- Math.max(), 680
- Math.min(), 681
- Math.pow(), 681
- Math.random(), 681
- Math.round(), 681
- Mouse.hide(), 682
- Mouse.show(), 682
- MovieClip.attachMovie(), 684
- MovieClip.beginFill(), 690

- MovieClip.createEmptyMovieClip(), 690
- MovieClip.curveTo(), 690
- MovieClip.duplicateMovieClip(), 684
- MovieClip.endFill(), 690
- MovieClip.getBytesLoaded(), 684
- MovieClip.getBytesTotal(), 685
- MovieClip.getURL(), 685
- MovieClip.gotoAndPlay(), 686
- MovieClip.gotoAndStop(), 686
- MovieClip.hitTest(), 686
- MovieClip.lineStyle(), 690
- MovieClip.loadMovie(), 687
- MovieClip.loadVariables(), 687
- MovieClip.localToGlobal(), 687
- MovieClip.moveTo(), 691
- MovieClip.nextFrame(), 688
- MovieClip.play(), 688
- MovieClip.prevFrame(), 688
- MovieClip.removeMovieClip(), 688
- MovieClip.startDrag(), 688
- MovieClip.stop(), 689
- MovieClip.stopDrag(), 689
- MovieClip.unloadMovie(), 689
- MovieClip.valueOf(), 689
- setVolume(), 697
- Sound.attachSound(), 693
- Sound.getPan(), 694
- Sound.getTransform(), 694
- Sound.getVolume(), 694
- Sound.setPan(), 696
- Sound.setTransform(), 696
- Sound.start(), 697
- Sound.stop(), 698
- String.charAt(), 699
- String.concat(), 699
- String.indexOf(), 699
- String.lastIndexOf(), 700
- String.slice(), 700
- String.toLowerCase(), 701
- String.toUpperCase(), 701
- Микшер цветов, 345
- Микширование, 587
- Многокнопочный интерфейс, 261
- Многоканальная запись, 615
- Моделирование органических объектов, 527
- Модель библиотеки
 - создание, 135
 - author-time, 134, 138
 - runtime, 134
- Модель цвета
 - RGB, 81
 - CMYK, 81
- Монозвук, 390
- Н
- Найквиста теорема, 388
- Насыщенность цветового оттенка, 141
- Нормализация звука, 396
- О
- Область действия, 44
 - размеры, 45
 - цвет, 45
- Обработчик события, 227
- Объект, 317
 - абсолютный путь, 334
 - вращение, 115
 - выбор, 62
 - выравнивание, 110
 - группирование, 107
 - действие, 226
 - замедление движения, 180
 - зеркальное отражение, 116
 - изменение формы, 64
 - изменение яркости, 187
 - интервал, 112
 - конструктор, 428
 - манипулирование конвертом, 118
 - масштабирование, 114
 - метод
 - getBytesLoaded, 447
 - getBytesTotal, 447
 - getPan, 431
 - getVolume, 430
 - loadSound, 447
 - setPan, 430
 - setVolume, 430
 - start, 429
 - stop, 429
 - относительный путь, 335
 - перемещение, 63, 109
 - разгруппирование, 107
 - размещение, 113
 - распределение, 111
 - согласование размеров, 112
 - трансформация, 113
 - трехмерный, 510
 - Arguments, 317
 - Array, 317, 667
 - Boolean, 317
 - Button, 317, 321
 - Capabilities, 317

- присвоения результата сложения (+=), 663
- присвоения результата умножения (*=), 662
- равенства (==), 664
- сложения (+), 663
- строгого равенства (===), 664
- умножения (*), 662
- условный (?:), 662
- Опорная точка
 - гладкая, 70
 - добавление, 72
 - угловая, 70
 - удаление, 72
- Отладчик, 366
 - доступ к переменным, 368
 - контрольная точка, 369
 - окно, 367
 - окно Output, 371
 - список контроля, 368
 - удаленная отладка, 370

П

Панель

- развертывание, 55
- свертывание, 55
- стыковка, 53
- Accessibility, 38, 51, 570
- Actions, 37, 50, 225
 - режим Expert, 223
 - режим Normal, 223
- Align, 49
- Color Mixer, 49, 82, 83
- Color Swatches, 49, 82
- Component Parameters, 52
- Components, 52
- Controller, 42
- Debugger, 50
- Info, 49
- Main, 41
- Movie Explorer, 51, 143
- Properties
 - манипулирование текстом, 90
 - экземпляр символа, 140
- Property
 - изменение заливки, 80
 - изменение обводки, 78
- Reference, 51
- Scene, 50
- Sitespring, 52
- Tools, 43

- Color, 317, 344
- Date, 317, 673
- Key, 317, 329
- LoadVars, 317
- Math, 317, 446
- Mouse, 317
- MovieClip, 317
- Number, 317
- Object, 317, 692
- Selection, 317
- Sound, 317, 427, 692
- Stage, 317
- String, 317, 698
- TextField, 317
- TextFormat, 318
- Transformer, 431
 - метод getTransform, 432
 - метод setTransform, 431, 432
- Video, 318
- XML, 318
- XML Socket, 318

Объектная ссылка

- global, 665
- level, 666
- parent, 666
- this, 667

Окно

- Debugger, 366
- Output, 51, 371

Окружающий свет, 520

Операнд, 311

Оператор, 311

- больше (>), 665
- больше или равно (>=), 665
- деления по модулю (%), 661
- доступа к элементу массива (:), 668
- инкрементирования (++), 660
- логического ИЛИ (||), 663
- логического НЕ (!), 660
- логическое И (&&), 661
- меньше (<), 664
- меньше или равно (<=), 664
- многострочный комментарий (/*...*/), 303, 663
- не равно (!=), 660
- однострочный комментарий (//), 303, 662
- присваивания (=), 664
- присвоения по модулю (%=), 661
- присвоения результата вычитания (-=), 664
- присвоения результата деления (/=-), 663

- Transform, 50
 - Панорамирование, 390, 430, 436, 441
 - Параметры публикации, 557
 - GIF, 563
 - HTML, 560
 - JPEG, 565
 - PNG, 566
 - Quick Time, 567
 - Пароль, 97
 - Передача информации из Flash в Director, 474
 - Переменная, 308
 - глобальная, 309
 - локальная, 310
 - область действия, 309
 - Поиск элементов, 145
 - Потоковый звук, 402
 - Привязка
 - к объектам, 64
 - к сетке пикселей, 65
 - Приложение
 - Adobe Premiere, 594
 - Adobe Type Manager, 88
 - Amari 3D, 532
 - Amorphium Pro
 - интерфейс, 538
 - источник света, 547
 - материалы, 539
 - экспорт во Flash, 539
 - Carrara Studio, 543
 - система моделирования, 544
 - Dreamweaver, 546, 550
 - Director, 465
 - Flash Player, 601
 - FreeHand
 - библиотека, 483
 - единицы измерения, 483
 - инструмент эллипс, 482
 - инспектор объектов, 484
 - инструмент Rectangle, 483
 - инструмент Pointer, 484
 - линейка, 483
 - окно Export Movie, 488
 - окно Release to Layers, 484
 - панель Controller, 485
 - панель Navigation, 486
 - переход, 484
 - символ, 483
 - слой, 482
 - LightWave 3D, 543
 - Poser
 - экспорт в SWF-файл, 529
 - Roxio Toast 5 Titanium, 610
 - Swift 3D, 517
 - анимация, 524
 - импорт файла, 518
 - новый проект, 518
 - открытие документа, 518
 - панель Property, 519
 - свет, 523
 - текст, 521, 523
 - экспорт в SWF-файл, 525
 - Vesta3D
 - импорт модели, 533
 - инструменты, 534
 - создание анимации, 536
 - Прозрачность, 83, 141
 - Псевдотрехмерная сфера, 511
 - Публикация, 554
 - на видеоленту, 597
 - Путь к объекту
 - абсолютный, 334
 - относительный, 335
- ## Р
- Рабочая область, 44
 - Разбивка текста, 97
 - Растровая заливка, 85
 - Расширенный компакт-диск, 615
 - Регулятор, 325
 - Редактирование сегмента кривой, 83
 - Редактор звука, 405
 - Режим
 - редактирования символов, 404
 - onion skinning, 175
 - onion skinning outlines, 176
 - Allow Smoothing, 105
 - Рисование
 - кривых, 71
 - одиночной прямой линии, 67
 - прямоугольников, 68
 - прямых линий, 71
 - эллипсов и окружностей, 67
- ## С
- Свет окружающий, 520
 - Свойство, 335
 - _alpha, 653
 - _dropTarget, 653
 - _framesloaded, 654
 - _height, 340
 - _name, 654

Слой, 46, 47, 148
 -- блокировка, 152
 -- контурное представление объектов, 151
 -- маска, 157
 -- отображение и сокрытие, 150
 -- перемещение, 154
 -- разблокировка, 152
 -- удаление, 155
 Смещение от базовой линии, 92
 Событие
 -- клипа, 231
 -- задание, 242
 -- data, 232
 -- load, 232
 -- мыши, 229
 -- задание, 236
 -- dragOver, 231
 -- keyPress, 231
 -- press, 231
 -- release, 231
 -- rollout, 231
 -- обработчик, 227
 -- Sound.onLoad, 695
 -- Sound.onSoundComplete, 695
 Состояние, 122
 -- кнопки
 -- Hit, 250
 -- Over, 250
 Спрайт, 465
 Ссылка `global`, 309
 Стереoeffект, 444
 Сфера псевдотрехмерная, 511
 Цена, 201

Т

Текст, 87
 Текстовое поле, 89, 350
 -- ввода, 89, 350
 -- расширяемое, 89
 -- фиксированное, 89
 Теорема Найквиста, 388
 Технология
 -- гибридная, 531
 -- ASP (Active Server Pages), 356
 -- NURBS, 531
 Тип данных, 306
 -- булевый, 307
 -- клип, 307
 -- неопределенный, 307
 -- объект, 307
 -- строковый, 306

-- `_quality`, 654
 -- `_rotation`, 340
 -- `_target`, 655
 -- `_visible`, 340, 656
 -- `_width`, 341, 656
 -- `_xmouse`, 341, 656
 -- `_xscale`, 341, 657
 -- `_ymouse`, 341, 657
 -- `_yscale`, 341
 -- `Array.length`, 669
 -- `currentFrame`, 653
 -- `frame`, 473
 -- `FrameCount`, 473
 -- `NaN`, 654
 -- `Sound.duration`, 694
 -- `Sound.position`, 695
 -- `String.length`, 700
 -- `TextField.hscroll`, 702
 -- `TextField.maxhscroll`, 702
 -- `TextField.maxscroll`, 702
 -- `TextField.vscroll`, 702
 Связывание действий
 -- с кадрами, 233
 -- с объектами, 229
 Связывающее имя, 432
 Сглаживание, 65
 Сжатие
 -- временное, 575
 -- метод Variable Bit Rate (VBR), 575
 Сжатие звукового файла
 -- речевое сжатие, 412
 -- ADPCM, 410
 -- MP3, 411
 Символ, 121, 226
 -- графический, 122, 123
 -- дублирование, 131
 -- импортированный, 122
 -- собственный Flash-символ, 122
 -- экземпляр, 121, 226
 -- присвоение имени, 193
 Символьная характеристика, 122
 Синтаксис
 -- с использованием символа `/`, 302
 -- точечный, 302
 Синхронизация, 588
 -- многоканальный и многослойный звук, 417
 -- речи, 417
 -- `Event`, 416, 422
 -- `Start`, 418
 -- `Stop`, 418
 -- `Stream`, 417

- числовой, 307

Тип звукового файла, 390

- AIFF, 390

- AU, 391

- MP3, 391

- Sound Designer II, 391

- System 7 Sound, 391

- WAV, 391

Точечный синтаксис, 302

Точка

- контрольная отладчика, 369

- регистрации, 119

- трансформации, 119

- управляющая, 71

Трассировка, 103

У

Управляющая точка, 71

Установка

- направления текста, 93

- положения символа, 92

- размера шрифта, 91

- свойств растрового изображения, 104

- стиля текста, 91

- тип загрузки URL, 93

- цвета текста, 92

- шрифта, 91

- шрифтов устройства, 94

Ф

Файл

- звуковой Sync Track, 462

- синхронизации звуковой, 453, 455

Формат

- AIFF, 592

- Flash, 555

- GIF, 555

- HTML, 555

- JPEG, 556

- Macintosh Projector, 557

- PNG, 556

- QuickTime, 412, 557

- Windows Projector, 557

Функция, 315, 658

- eval(), 658

- getProperty(), 658

- getTimer(), 658

- String(), 659

- targetPath(), 659

Х

Характеристическое отношение, 586

Ц

Цветовая модель

- Hexadecimal notation, 81

- HSB создание цвета, 83

- RGB создание цвета, 83

Цветовая палитра безопасная, 81

Цветовой порог, 103

Цветовой тон, 141

Цикл, 315

Ч

Частота

- выборки, 387

- кадров, 453

- калькулятор 455

Ш

Шрифт, 87

Э

Экспорт Flash-фильма

- QuickTime Video, 589

- Windows (AVI), 592

Элемент управления

- CheckBox, 337

- ComboBox, 337

- ListBox, 337

- PushButton, 337

- RadioButton, 337

- ScrollBar, 337

- ScrollPane, 337

Эффект letterbox, 605

Эффект звуковой, 405, 453

- Fade In, 406

- Fade Left to Right, 406

- Fade Out, 406

- Fade Right to Left, 406

- Left/Right Channel, 406

Я

Ядро, 465

- компонент, 465

Язык программирования Lingo, 465

Язык сценариев

- ActionScript, 30, 37
 - управление воспроизведением видео, 580
- CFML (ColdFusion Markup Language), 356
- CGI (Common Gateway Interface), 358
- PHP, 357

Яркость, 140

**A**

ActionScript, 37, 280

- аргумент, 286
- действие, 286
- класс, 286
- комментарий, 366
- константа, 286
- окно Output, 371
- оператор if, 283
- отладчик, 366
- цикл, 284

Adobe Type Manager, приложение, 88

Amorphium Pro, приложение, 546

ASP (Active Server Pages), технология, 356

author-time, модель библиотеки, 134

C

CFML (ColdFusion Markup Language), язык сценариев, 356

CGI (Common Gateway Interface), язык сценариев, 358

CheckBox, элемент управления, 337

ComboBox, элемент управления, 337

D

DirectX 7, программа, 123

DoReMedia, звуки, 453

Dreamweaver, приложение, 546, 550

F

Flash projector, тип файла, 600

Flash-фильм

- выбор уровней, 336
- загрузка, 335
- импорт в среде Director, 465
- использование в качестве спрайта, 470

-- предварительный просмотр, 569

— публикация, 569

-- свойства в среде Director, 467

-- связывание в среде Director, 467

— управление воспроизведением при помощи Lingo, 472

Frame by frame, метод анимации, 168

FreeHand, приложение

— библиотека, 483

— единицы измерения, 483

— инструмент эллипс, 482

-- инспектор объектов, 484

-- инструмент Rectangle, 483

— инструмента Pointer, 484

— линейка, 483

— окно Export Movie, 488

— окно Release to Layers, 484

— панель Controller, 485

— панель Navigation, 486

-- переход, 484

— символ, 483

— слой, 482

FutureSplash Animator, приложение, 29

Future Wave, компания, 28

K

Keyboard Shortcuts, редактор, 57

L

Letterbox, эффект, 605

Lingo, язык программирования, 465

ListBox, элемент управления, 337

M

Micromedia, компания, 28

P

PHP, язык сценариев, 357

PushButton, элемент управления, 337

R

RadioButton, элемент управления, 337

runtime, модель библиотеки, .134

S

ScrollBar, элемент управления, 337

ScrollPane, элемент управления, 337

Shape tweening, тип анимации, 172

SmallSketch, графическое приложение, 29

Soreson Spark, кодек, 575

SQL, язык структурированных
запросов, 350

Sync Track, звуковой файл, 462

T

TrueType, шрифт, 88

Tweening, метод анимации, 168

Type 1 PostScript, шрифт, 88

U

URL-указатель, 93

V

Variable Bit Rate (VBR), метод сжатия, 575

W

Web-сервер, 349

World Wide Web Consortium, 556

КЛУБ ПРОФЕССИОНАЛ

В 1997 году по инициативе генерального директора **Издательского дома «Питер»** Валерия Степанова и при поддержке деловых кругов города в Санкт-Петербурге был основан **«Книжный клуб Профессионал»**. Он собрал под флагом клуба профессионалов своего дела, которых объединяет постоянная тяга к знаниям и любовь к книгам. Членами клуба являются лучшие студенты и известные практики из разных сфер деятельности, которые хотят стать или уже стали профессионалами в той или иной области.

Как и все развивающиеся проекты, с течением времени книжный клуб вырос в **«Клуб Профессионал»**. Идею клуба сегодня формируют три основные «клубные» функции:

- неформальное общение и совместный досуг интересных людей;
- участие в подготовке специалистов высокого класса (семинары, пакеты книг по специальной литературе);
- формирование и высказывание мнений современного профессионала (при встречах и на страницах журнала).

КАК ВСТУПИТЬ В КЛУБ?

Для вступления в **«Клуб Профессионал»** вам необходимо:

- ознакомиться с правилами вступления в **«Клуб Профессионал»** на страницах журнала или на сайте www.piter.com;
- выразить свое желание вступить в **«Клуб Профессионал»** по электронной почте postbook@piter.com или по тел. **(812) 387-01 -04**;
- заказать книги на сумму не менее 500 рублей в течение любого времени или приобрести комплект **«Библиотека профессионала»**.

«БИБЛИОТЕКА ПРОФЕССИОНАЛА»

Мы предлагаем вам получить все необходимые знания, подписавшись на **«Библиотеку профессионала»**. Она для тех, кто экономит не только время, но и деньги. Покупая комплект - книжную полку **«Библиотека профессионала»**, вы получаете:

- **скидку 15%** от розничной цены издания, без учета почтовых расходов;
- при покупке двух или более комплектов - дополнительную **скидку 3%**;
- членство в **«Клубе Профессионал»**;
- подарок - журнал **«Клуб Профессионал»**.

**Закажите бесплатный журнал
«Клуб Профессионал».**

ИЗДАТЕЛЬСКИЙ ДОМ
ПИТЕР[®]
WWW.PITER.OOM



КНИГА-ПОЧТОЙ



**ЗАКАЗАТЬ КНИГИ ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР»
МОЖНО ЛЮБЫМ УДОБНЫМ ДЛЯ ВАС СПОСОБОМ:**

- по телефону: **(812) 387-01 -04;**
- по электронному адресу: **postbook@piter.com;**
- на нашем сервере: **www.piter.com;**
- по почте: **197198, Санкт-Петербург, а/я 619
ЗАО «Питер Пост».**

**ВЫ МОЖЕТЕ ВЫБРАТЬ ОДИН ИЗ ДВУХ СПОСОБОВ ДОСТАВКИ
И ОПЛАТЫ ИЗДАНИЙ:**

-  Наложенным платежом с оплатой заказа при получении посылки на ближайшем почтовом отделении. Цены на издания приведены ориентировочно и включают в себя стоимость пересылки по почте **(но без учета авиатарифа)**. Книги будут высланы нашей службой «Книга-почтой» в течение двух недель после получения заказа или выхода книги из печати.
-  Оплата наличными при курьерской доставке **(для жителей Москвы и Санкт-Петербурга)**. Курьер бесплатно доставит заказ по указанному адресу в удобное для вас время в течение трех дней. Такой заказ лучше оформлять по телефону.

ПРИ ОФОРМЛЕНИИ ЗАКАЗА УКАЖИТЕ:

- фамилию, имя, отчество, телефон, факс, e-mail;
- почтовый индекс, регион, район, населенный пункт, улицу, дом, корпус, квартиру;
- название книги, автора, код, количество заказываемых экземпляров.

**Вы можете заказать бесплатный
журнал «Клуб Профессионал».**

ИЗДАТЕЛЬСКИЙ ДОМ
 **ПИТЕР**[®]
WWW.PITER.COM

ПРЕДСТАВИТЕЛЬСТВА ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР»
предлагают эксклюзивный ассортимент компьютерной, медицинской,
психологической, экономической и популярной литературы

РОССИЯ

Москва

Представительство издательства «Питер»
М. «Калужская», ул. Бутлерова, д. 176, офис 207, 240
Тел./факс (095) 777-54-67
E-mail: sales@piter.msk.ru

Санкт-Петербург

Представительство издательства «Питер»
М. «Выборгская», Б. Сампсониевский пр., д. 29а
Тел. (812) 103-73-73, факс (812) 103-73-82
E-mail: sales@piter.com

Воронеж

Представительство издательства «Питер»
Ул. Ленинградская, д. 138
Тел. (0732) 49 68 86
E-mail: piter-vrn@vmail.ru

Екатеринбург

Представительство издательства «Питер»
Ул. Завокзальная, д. 4а, склад 10а
Тел.(3432) 33-96-40
E-mail: piter-ural@r66.ru

Нижний Новгород

Представительство издательства «Питер»
Ул. Премудрова, д. 31а
Тел. (8312) 58-50-15, 58-50-25
E-mail: piter@infonet.nnov.ru

Ростов-на-Дону

Представительство издательства «Питер»
Ул. Калитвинская, д. 17в
Тел. (8632) 95-36-31, (8632) 95-36-32
E-mail: jupiter@rost.ru

УКРАИНА

Харьков

Представительство издательства «Питер»
Ул. Энгельса, д. 29а, офис 610
Тел. (0572) 23-75-63, (0572) 28-20-04,
(0572) 28-20-05, факс (0572) 14-96-09
E-mail: piter@tender.kharkov.ua

Киев

Представительство издательства «Питер»
Пр. Красных казаков, д. 6, корп. 1
Тел./факс (044) 490-35-68, 490-35-69
E-mail: office@piter-press.kiev.ua

БЕЛАРУСЬ

Минск

Представительство издательства «Питер»
Ул. Бобруйская д., 21, оф. 3
Тел/факс (37517) 226-19-53
E-mail: piterbel@tut.by

МОЛДОВА

Кишинев

«Ауратип-Питер»
Ул. Митрополит Варлаам, 65, офис 345
Тел. (3732) 22-69-52, факс (3732) 27-24-82
E-mail: lili@auratip.mldnet.com



Ищем зарубежных партнеров или посредников, имеющих выход на зарубежный рынок.
Телефон для связи: **(812) 103-73-73.**
E-mail: grigorjan@piter.com



Редакции компьютерной, психологической, экономической, юридической, медицинской, учебной и популярной (оздоровительной и психологической) литературы **Издательского дома «Питер»** приглашают к сотрудничеству авторов.
Обращайтесь по телефонам: **Санкт-Петербург - тел.: (812) 103-73-72,**
Москва - тел.: (095) 234-38-15, 777-54-67.

Башкортостан

Уфа, «Азия», ул. Зенцова, д. 70 (оптовая продажа),
маг. «Оазис», ул. Чернышевского, д. 88,
тел./факс (3472) 50-39-00.
E-mail: asiaufa@ufanet.ru

Дальний Восток

Владивосток, «Приморский торговый дом книги»,
тел./факс (4232) 23-82-12.
E-mail: bookbase@mail.primorye.ru

Хабаровск, «Мирс»,
тел. (4212) 30-54-47, факс 22-73-30.
E-mail: sale_book@bookmirs.khv.ru

Хабаровск, «Книжный мир»,
тел. (4212) 32-85-51, факс 32-82-50.
E-mail: postmaster@worldbooks.kht.ru

Европейские регионы России

Архангельск, «Дом книги»,
тел. (8182) 65-41-34, факс 65-41-34.
E-mail: book@atnet.ru

Калининград, «Вестер»,
тел./факс (0112) 21-56-28, 21-62-07.
E-mail: nshibkova@vester.ru
<http://www.vester.ru>

Ростов-на-Дону, ПБОЮЛ Остроменский,
пр. Соколова, д. 73,
тел./факс (8632) 32-18-20.
E-mail: ostrom@don.sitek.net

Северный Кавказ

Ессентуки, «Россы», ул. Октябрьская, 424,
тел./факс (87934) 6-93-09.
E-mail: rossy@kmw.ru

Сибирь

Иркутск, «ПродаЛитЪ»,
тел. (3952) 59-13-70, факс 51-30-70.
E-mail: prodalit@irk.ru
<http://www.prodalit.irk.ru>

Иркутск, «Антей-книга»,
тел./факс (3952) 33-42-47.
E-mail: antey@irk.ru

Красноярск, «Книжный мир»,
тел./факс (3912) 27-39-71.
E-mail: book-world@public.krasnet.ru

Нижневартовск, «Дом книги»,
тел. (3466) 23-27-14, факс 23-59-50.
E-mail: book@nvarovsk.wsnet.ru

Новосибирск, «Топ-книга»,
тел. (3832) 36-10-26, факс 36-10-27.
E-mail: office@top-kniga.ru
<http://www.top-kniga.ru>

Тюмень, «Друг»,
тел./факс (3452) 21-34-82.
E-mail: drug@tyumen.ru

Тюмень, «Фолиант»,
тел. (3452) 27-36-06, факс 27-36-11.
E-mail: foliant@tyumen.ru

Челябинск, ТД «Эврика», ул. Барбюса, д. 61,
тел./факс (3512) 52-49-23.
E-mail: evrika@chel.surnet.ru

Татарстан

Казань, «Таис»,
тел. (8432) 72-34-55, факс 72-27-82.
E-mail: tais@bancorp.ru

Урал

Екатеринбург, магазин № 14,
ул. Челюскинцев, д. 23,
тел./факс (3432) 53-24-90.
E-mail: gvardia@mail.ur.ru

Екатеринбург, «Валео-книга»,
ул. Ключевская, д. 5,
тел./факс (3432) 42-56-00.
E-mail: valeo@etel.ru

Этан Уотролл, Норберт Гербер

*Эффективная
Работа™*

Flash MX

На страницах книги вы найдете все необходимые сведения для продуктивной работы с Macromedia Flash MX:

- рисование фигур, обработка текста;
- библиотеки символов;
- создание анимации;
- * использование языка ActionScript;
- разработка интерфейса Flash-фильма;
- синхронизация, сжатие и программное управление звуком;
- интеграция со звуковыми редакторами, Macromedia Director и другими приложениями;
- публикация фильмов на различных носителях.



CD-ROM

На компакт-диске вы найдете:

- необходимые файлы для выполнения примеров и упражнений;
- готовые примеры анимаций и интерактивных фильмов;
- пробные версии Flash MX, а также других приложений для Windows и MacOS.

УРОВЕНЬ ПОЛЬЗОВАТЕЛЯ: **НАЧИНАЮЩИЙ/ОПЫТНЫЙ**

КАТЕГОРИЯ КНИГИ: **УЧЕБНОЕ РУКОВОДСТВО**

Посетите наш web-магазин: www.piter.com

Посетите web-сайт издательства BHV: www.bhv.kiev.ua



Авторы ставили перед собой цель создать нечто большее, чем стандартное руководство по использованию Flash MX. В этой книге подробно описаны возможности программы, а также на профессионально выполненных примерах показано, как можно реализовать творческие идеи в области анимации. Данная книга адресована web-дизайнерам, художникам и аниматорам, работающим в операционных системах Windows и MacOS.

ISBN 5-8046-0009-5



9 785804 600090