

VIDEO

MACROMEDIA

FLASH MX

СПРАВОЧНИК ПРОФЕССИОНАЛА

Кристиан Бесли
Хосс Джиффорд
Брайан Моннон
Тодд Маркс



friends of ED.gotoAndPlay

Самые замечательные книги. Быстрая, удобная и бесплатная техническая поддержка

Практическое руководство, учебное пособие, справочник, собрание техник и методов, источник вдохновения. Все ответы на ваши вопросы о Flash MX и ActionScript - здесь: <http://www.flashmxlibrary.com>

Flash **MX** Library.com

ЭКОМ

Flash MX Video

Flash MX Video

Kristian Besley
Hoss Gifford
Brian Monnone
Todd Marks

friends of ED

Flash MX Video

Кристиан Бесли
Хосс Джиффорд
Брайан Моннон
Тодд Маркс

Издательство

СП ЭКОМ

Москва, 2003

ББК 32.97

УДК 681.3

К. Бесли, Х. Джиффорд, Б. Моннон, Т. Маркс

Flash MX Video. Справочник профессионала. **Практ. пособ.** /Пер. с англ.—М.: Издательство «СП ЭКОМ», 2003.—320 с.: илл.

ISBN 5-9570-0011-6

Macromedia Flash MX— это возможность импортировать цифровое видео и звук и управлять ими, как и любыми другими мультимедийными объектами. Эта книга - ваш проводник в волшебный мир видео-дизайна. В первом разделе вы найдете исчерпывающие сведения о том, как импортировать видео и звук во Flash и о способах последующего манипулирования этими данными. Второй раздел посвящен тому, как создавать настоящие видеофильмы - от подготовки к производству до выпуска конечного продукта, размещения его в сети или на CD-ROM. В последнем разделе мы погрузимся в ActionScript и создадим полностью объектно-ориентированный компонент Flash для воспроизведения звука и видео. Давайте делать фильмы!

ББК 32.97

Copyright © 2002 friends of ED

© «Издательство ЭКОМ», Москва, 2003, перевод и оформление

ISBN 1-903450-85-3 (англ.)

ISBN 5-9570-0011-6

Оглавление

Добро пожаловать.....	11
Где взять тексты программ и видео.....	11
Оформление и условные обозначения.....	12
Мы всегда готовы помочь.....	13
Раздел 1	
Основные концепции.....	15
Глава 1. Введение в видео.....	16
Как используется сегодня видео во всемирной сети?.....	16
Музыкальные видео-клипы.....	17
Кино-трейлеры.....	18
Демонстрация товаров.....	19
Обучающие видеоролики.....	19
Телевизионное и веб-вещание.....	20
Плохая новость: почему WWW не является новым TV.....	21
Слишком мало пользователей имеет широкополосное подключение.....	22
У средней «персоналки» не хватает мощности.....	22
Нет единого стандарта представления видео.....	23
Хорошие новости: Flash MX дает ответ.....	23
Наконец-то действительно широко распространенный видео-модуль.....	24
Наилучший компромисс между размером файла и загрузкой процессора.....	25
Анимация (мультипликация).....	26
Образовательные материалы.....	26
Фильмы.....	27
Закключение.....	27
Глава 2. Подготовка видео материала для Flash.....	29
Нелинейный монтаж.....	30
FireWire и DV.....	31
Переходы.....	31
Movie Maker для Windows XP.....	34
Начинаем.....	35
Захват видео.....	36
Импортирование файла.....	39
Работа со сборником клипов.....	41
Деление клипа на две части.....	41
Объединение нескольких клипов.....	42

Режимы монтажа в Movie Maker.....	42
Режим «раскадровка» (storyboard).....	42
Режим «временная диаграмма» (timeline).....	43
Монтаж клипов.....	44
Экспорт.....	47
iMovie для Mac OS.....	48
Начинаем.....	49
Захват видео.....	50
Импортирование файлов в iMovie.....	52
Работа с клипами.....	53
Режимы монтажа в iMovie.....	56
Режим «раскадровка» (storyboard).....	57
Режим «временная диаграмма» (timeline).....	57
Монтаж клипов.....	58
Экспорт фильма.....	62
Прочие возможности и трюки iMovie.....	63
Направление клипов.....	64
Скорость клипов.....	64
Специальные эффекты.....	65
Звук: громкость и фединг.....	66
Импорт звука с CD.....	66
Запись с микрофона.....	67
Титры.....	67
Заключение.....	70
Глава 3. Перенос видео материала во Flash.....	71
Прежде всего.....	72
Импортирование видео: подготовка.....	73
Параметры видео.....	74
Частота кадров.....	75
Качество.....	75
Интервал ключевого кадра.....	75
Масштаб.....	78
Синхронизация.....	78
Звук.....	78
Устанавливаем параметры импорта видео.....	79
Sorenson Squeeze.....	81
Сжатие видео.....	84
Параметры сжатия.....	84
Параметры видео фильтра.....	89
Параметры сжатия.....	91
Перенос видео во Flash.....	93
Заключение.....	94

Глава 4. Обработка видео во Flash	96
Потоковая обработка видео во Flash.....	96
Потоковая передача без буферизации.....	97
Профиль полосы.....	101
Буферизация потока.....	102
Автономные видеоклипы.....	104
Видео-стена.....	105
Манипуляции с монтажными столами.....	ПО
Кнопка «пауза».....	111
Кнопка воспроизведения.....	115
Кнопки перемотки вперед и назад.....	118
Управление несколькими видеоклипами.....	121
Хочу Flash-телевизор!.....	122
Кнопки управления.....	126
Добавляем звук.....	129
Маскирование.....	131
Заключение.....	134
Глава 5. Применение эффектов для Flash видео	135
Нестандартные приемы маскирования.....	135
Текст в качестве маски.....	137
Текстуры ...	138
Множественные маски.....	141
Параметры видеоклипа.....	142
Различные вращения.....	142
Скос.....	145
Кубик.....	145
Зеркальное отражение.....	149
С обратной стороны.....	149
Яркость.....	151
Тонирование.....	151
Убийственный танцпол.....	152
Альфа-фактор.....	154
Проявление и растворение.....	154
Расширенные эффекты.....	158
Повышение резкости видео изображения.....	158
Инверсия цвета.....	159
Заключение.....	161
Глава 6. Звук	162
Звуковые редакторы.....	162
Фильтрация звука.....	163
Sound Forge для Windows.....	164
Работаем в «Кузнице звука».....	164

Sound Studio 2 для Macintosh.....	170
Работаем в «Студии».....	170
Звук во Flash.....	176
Музыка.....	176
Работа без скриптов.....	178
Потоковая обработка и события.....	182
Поток.....	182
Событие.....	183
Start и Stop.....	184
Звуковые эффекты Flash.....	185
Подправляем звукозапись.....	187
Заключение.....	190
Глава 7. Объект типа Sound.....	191
Знакомство с объектом Sound.....	191
start.....	192
stop.....	192
setPan.....	192
setVolume.....	192
loadSound.....	193
Дополнительные методы.....	193
Динамический звук.....	193
Добавляем графический интерфейс.....	197
Звук и видео в «свободном плавании».....	199
Использование setVolume.....	200
Видео «плавает» вместе со звуком.....	202
Параметр position.....	203
Я и мой звук.....	204
Не прекращать до onSoundComplete.....	208
Просто, как MP3.....	208
Заключение.....	212
Раздел 2	
Конкретные примеры.....	213
Глава 8. Конкретные примеры: планирование, изготовление и компоновка.....	214
Неудачный план или планирование неудач.....	216
Просмотр отснятого материала.....	217
Скачивание видео.....	218
Настройка.....	222
Заключение.....	223

Глава 9. Внутри Flash	224
Абстракция.....	225
Структура Flash.....	226
Пора начинать.....	227
Подготовка к видео.....	227
Сборка.....	229
Бразды правления.....	233
Начало и остановка воспроизведения.....	234
Перемотка вперед, назад и замедленный показ.....	235
Заключение.....	239
Глава 10. Подключение звука и распространение проекта	241
Звук.....	241
Подключение звука.....	242
Приглушение звука.....	245
Уже загрузили?.....	247
Создание индикатора выполнения.....	248
В каком мы месте фильма?.....	250
Создание индикатора просмотра.....	250
Ну а теперь можно распространять проект?.....	251
Подготовка.....	251
Заключение.....	258
Раздел 3	
Передовые идеи	259
Глава 11. Программирование с использованием видео и звука	260
Объекты и объектно-ориентированное программирование.....	261
Простейший объект.....	262
Динамический Фред.....	262
Методы и свойства.....	264
Вращение Фреда.....	264
Классы и наследование.....	265
Знакомьтесь, Фред Младший.....	265
Прототипы.....	266
Переделки.....	266
Область действия.....	267
Определение области действия объектов.....	268
Компоненты.....	269
Компонент ListBox.....	270
Компонентпроигрыватель.....	272
Фильм для кнопки stop.....	273
Фильм для кнопки play.....	275
Фильм pan.....	277

Фильм Volume.....	280
Индикатор воспроизведения.....	283
Создание клипа controls.....	284
Эквалайзер.....	287
Объекты для эквалайзера.....	288
Изображение для bar.....	291
Постепенное появление.....	294
Фильм equalizer.....	295
Файлы видео.....	297
Подключение видео.....	297
Окончательная сборка.....	301
Заключение.....	311
Послесловие.....	313
Приложение	
Прочие видео надстройки.....	314
QuickTime.....	314
Распространение.....	315
Поддерживаемые форматы файлов.....	315
Потоковая передача.....	315
Интерфейс.....	315
Прочая информация.....	316
RealOne Player.....	316
Распространение.....	317
Поддерживаемые форматы файлов.....	317
Потоковая передача.....	317
Интерфейс.....	317
Windows Media Player.....	318
Распространение.....	318
Поддерживаемые форматы файлов.....	318
Потоковая передача.....	319
Интерфейс.....	319

Добро пожаловать

Давайте делать фильмы! Интерактивные фильмы, полностью интегрированные в интерфейс Flash! С Macromedia Flash MX вы получаете возможность импортировать цифровое видео и звук, манипулировать ими, как и любыми другими мультимедийными объектами. Это целый мир возможностей дизайна, а конечный продукт может быть размещен в интернете, записан на CD-ROM или стать просто некой эксцентричной видео-игрушкой, которой вы будете забавляться на своей собственной машине.

Мы надеемся, что с помощью этой книги вы узнаете все, что нужно знать о производстве таких удивительных игрушек. В первом разделе вы найдете исчерпывающие сведения о том, как импортировать видео и звук во Flash и о способах последующего манипулирования этими данными. Во втором разделе мы будем учиться создавать настоящие видеофильмы — от подготовки к производству до выпуска конечного продукта, размещения его в сети или на CD-ROM. Наконец, в последнем разделе мы погрузимся в сумрачный мир ActionScript и создадим полностью объектно-ориентированный компонент Flash для воспроизведения звука и видео.

Однако мы, кажется, забегаем вперед — следует дать некоторые пояснения, чтобы вы могли использовать книгу с максимальной эффективностью.

Где взять тексты программ и видео

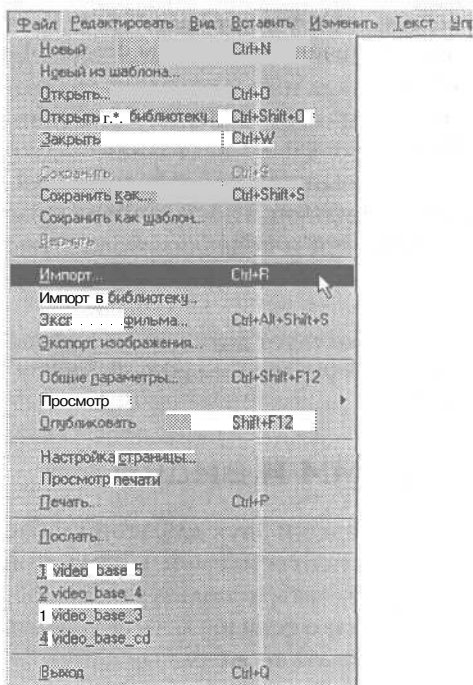
Файлы, содержащие примеры программ, видео и звук для этой книги, имеются в изобилии. Вообще, они являются неотъемлемой частью этой книги — все, что в ней описано, вы можете сделать, используя ваши собственные файлы MPG, MOV и скрипты. Однако если вы хотите увидеть именно то, что сделали авторы или использовать те же самые клипы — пожалуйста. Они систематизированы по главам и находятся по адресу www.friendsofED.com/flashmxvideo.

Кстати, если вы еще не добрались до Macromedia Flash MX, вы можете скачать полнофункциональную 30-дневную пробную версию с www.macromedia.com.

Оформление и условные обозначения

Мы старались сделать книгу максимально понятной и простой, и поэтому использовали специальное оформление.

- Различные стили текста призваны обратить ваше внимание на то, что появляется на экране, пояснить, что речь идет о НАЖАТИИ КЛАВИШИ или ЯРЛЫКЕ, либо о гиперссылке.
- Если мы вводим какой-либо новый важный термин, он выделяется полужирным шрифтом.
- *Если речь идет о чем-то важном, текст будет выделен так, как здесь!*
- Когда мы хотим, чтобы вы щелкнули на меню и затем выбрали команду или подменю, это будет выглядеть так: **Файл > Импорт (File > Import)**.



- Практический пример, который вам следует выполнить, будет предваряться заголовком.

1. Затем будут перечислены шаги, которые следует выполнить.
2. Следуйте указаниям; при этом вам могут встретиться снимки экрана или поясняющие диаграммы.

Дальнейшие пояснения могут быть напечатаны с отступом, как здесь

3. Выполняйте шаги, пока не достигнете конца упражнения.

Мы всегда готовы помочь

Мы старались, чтобы наши примеры были доступными и не содержали ошибок. Однако если вы все-таки столкнулись с проблемами, без колебаний связывайтесь с нами — наша служба поддержки дружелюбная и бесплатная.

Вы можете связаться с нами по адресу support@friendsofED.com, указав в поле темы сообщения последние четыре цифры номера ISBN (то есть, 0853), и даже если наша команда поддержки не сможет решить вашу проблему немедленно, ваш вопрос будет передан создателям книги, редакторам и авторам. Наши авторы либо отвечают читателям сами, либо (чаще всего) связываются с редактором.

Мы будем рады получить от вас весточку, даже если это просто запрос о готовящихся изданиях или рассказ о том, как вы любите *Macromedia Flash MX Video!*

Если у вас возникли вопросы по этому изданию, не связанные напрямую с содержанием книги, их лучше поместить на нашей доске объявлений на <http://friendsofed.inforop.net/2/OpenTopic>. Здесь вы встретите многих дизайнеров с рассказами об их деятельности, и, вероятно, найдете какие-то идеи и решения.

Новости, другие книги, примеры, файлы для скачивания, интервью с авторами и многое другое — по адресу www.friendsofED.com. Узнать больше о серии книг friends of ED о Flash MX можно по адресу www.flashmxlibrary.com.

Раздел 1

ОСНОВНЫЕ КОНЦЕПЦИИ

Введение в видео

Краткое содержание главы:

- *многообразие применений видео во всемирной сети;*
- *технологические **ограничения** для видео-содержания в сети;*
- *как, используя Macromedia Flash MX, посредством **внедрения видео** преодолеть эти ограничения;*
- *примеры того, как с помощью Macromedia Flash MX, творчески сочетая **видео, интерактивность и векторную графику**, можно достигнуть ошеломляющих результатов.*

Эта книга написана не для того, чтобы просто перечислить сухие технические подробности использования видео с Macromedia Flash MX; она призвана проиллюстрировать творческие **возможности**, открывающиеся благодаря обработке видео с помощью Flash MX. Короче говоря, мы хотим в равной мере вдохновлять и объяснять.

А потому, прежде чем перейти к практической стороне применения видео с Flash MX, мы хотим отступить немного назад и задуматься о том, как используется сегодня видео во всемирной сети. Кое-кто провозглашал появление видео в сети, так давайте посмотрим, насколько это соответствует действительности, насколько удачно это получается, что мешает и сдерживает и как Flash MX предлагает решать некоторые из этих проблем.

Как используется сегодня видео во всемирной сети?

Видео в сети нашло довольно разнообразное применение. Мы сделаем краткий обзор основных направлений: музыкальные видеоклипы, отрывки из фильмов, представление товаров, ролики образовательного содержания, а также телевизионное вещание для веб.

Музыкальные видео-клипы

С наступлением бума MTV музыкальные видеоклипы, предназначенные, прежде всего, для продвижения на рынке новых альбомов, приобрели огромное значение. Миллионы долларов, затраченные на создание клипов, гарантируют большие объемы продаж альбомов благодаря эфирной трансляции.

Чтобы сделать клипы выделяющимися из общей массы, директора вынуждены вкладывать в них все больше и больше средств. Недавняя тенденция — дополнять клипы «повествовательной» и танцевальной частями в начале, середине и/или в конце. Директорам становится все труднее удерживаться на высшем уровне, и они заимствуют технологии из других областей в попытке создать что-нибудь свеженькое.

Поп-клипы успешно завоевали телевидение, но как же перенести их во всемирную сеть? Конечно же, всякий уважающий себя артист имеет веб-сайт, где акцент сделан, естественно, на том, чтобы привлечь потенциального потребителя, и видеоклипы могли бы сыграть в этом важную роль. Онлайн-видеоклипы способны не только поощрить преданных фанов, но и помочь завоевать новых поклонников, показав нечто более интересное, чем статичные изображения кумиров в сопровождении аккордов (конечно, последнее тоже можно сделать очень неплохо, однако...). Отличный пример использования видеоклипов — www.mtv.com/music/video.



Кино-трейлеры

Этим термином (movie trailers) называют видео-ролики, создаваемые для рекламы новых фильмов. К моменту выхода фильма на экраны обычно подготавливают несколько таких роликов. В качестве примера можно привести рекламный материал к фильму «Звездные войны Эпизод II: Атака клонов», который включал четыре трейлера совершенно различного содержания! Трейлеры, предназначенные для демонстрации в кинотеатрах, как правило, бывают довольно продолжительными, тогда как для телевидения создаются сокращенные версии. Ролики для сети обычно более содержательны, чем теле-версии, так что пользователи всемирной паутины имеют возможность получить представление о фильме, прежде чем отправиться в кинотеатр.

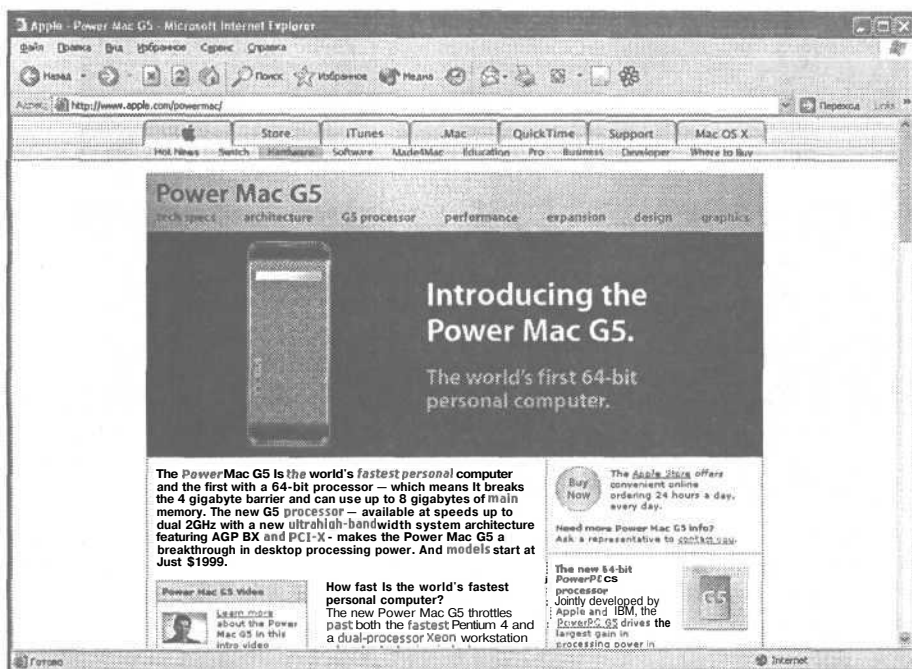
Сайт QuickTime (www.apple.com/quicktime) получил признание благодаря тому, что на нем размещено большое количество трейлеров, а также удобству их просмотра, поскольку учитывается пропускная способность линии пользователя. Ролики различаются по размеру и качеству видео. Например, трейлер к фильму *Amelie*, как можно видеть на снимке, доступен в трех различных категориях пропускной способности (низкая — до 56к; средняя — цифровые абонентские линии (DSL), ISDN и кабельные сети; и высокая — широкополосные сети).



Демонстрация товаров

Навязчивая телевизионная реклама во время фильмов и других передач стала уже притчей во языцех. Потокное вещание во всемирной сети (как, например, Real Networks) в последнее время тоже не обходится без 10-15 секундных рекламных вставок.

Однако демонстрация товаров не всегда является прямой рекламой. В некоторых случаях видео помогает лучше рассмотреть продукт, полнее раскрыть его возможности — задачи, с которыми трудно справиться неподвижному изображению. Галерея компьютерного оборудования на www.apple.com — еще один подход к показу изделий: пользователь может приблизить предмет, как угодно повернуть, рассмотреть под любым углом. Такая возможность позволяет получить потрясающе верное представление об изделии, словно вы держите вещь в руках. Вот, например, Apple PowerBook G4:

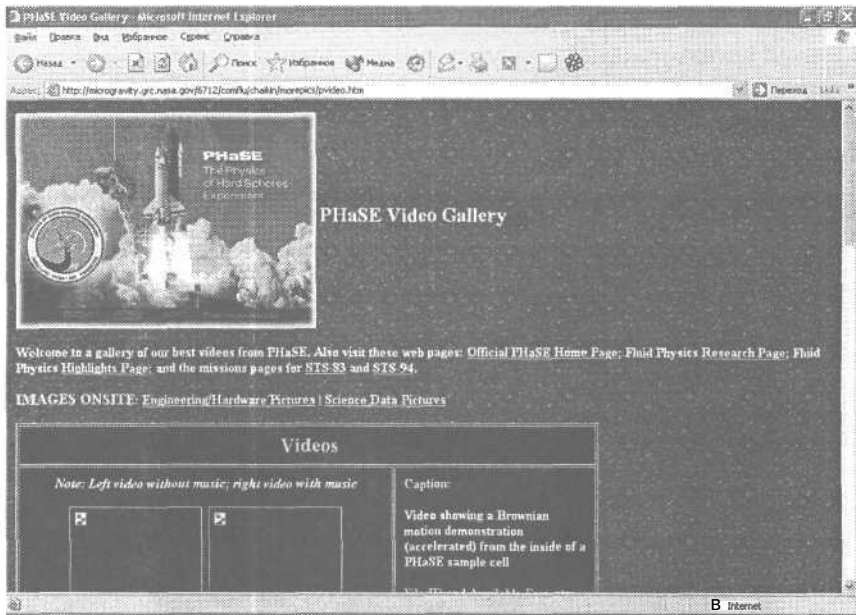


Обучающие видеоролики

Веб-видео широко используется как средство обучения в различных формах — от видеозаписей до 3D анимации. Видео — фантастическое средство внеклассного обучения, позволяющее заинтересовать ребен-

ка, усвоение информации происходит при этом практически на уровне подсознания.

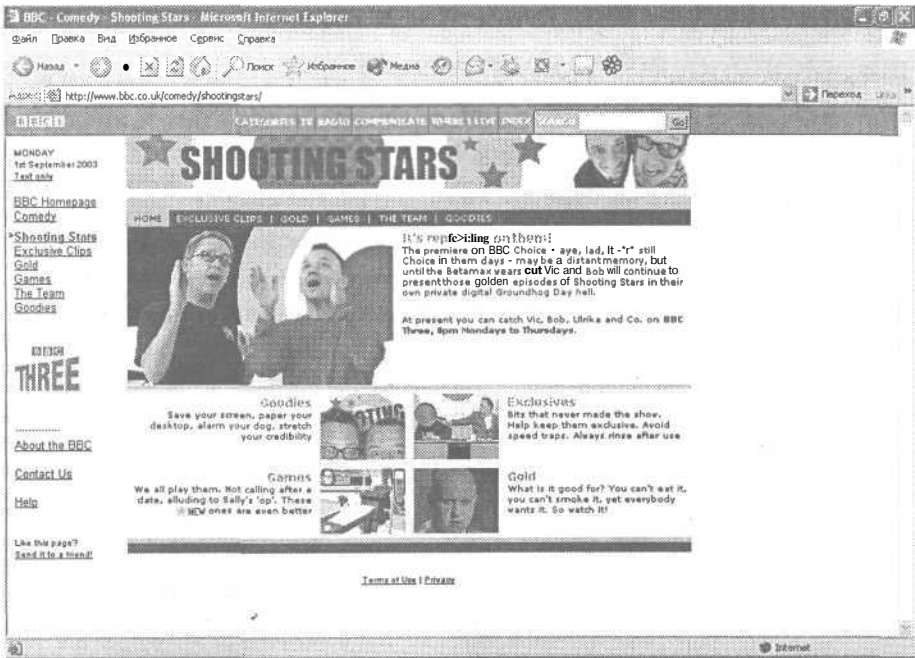
Образовательная информация в сети не ограничена только школьным уровнем. Видео давно уже используется как средство визуального представления научных идей, является важным компонентом содержания таких сайтов, как, например, NASA Glenn Research Center (<http://microgravity.grc.nasa.gov>), где видеоролики служат великолепными иллюстрациями для объяснения сложных теорий. Согласитесь, при помощи видео куда легче представить себе броуновское движение частиц!



Телевизионное и веб-вещание

Думается, многие из нас сталкивались с проблемами программирования видеомэгагитофонов, и кое-кому при этом приходила в голову мысль, что было бы неплохо иметь возможность посмотреть пропущенные телепередачи в интернете. Что ж, сегодня для вас, возможно, уже не будет откровением то, что просмотр телепрограмм во всемирной паутине стал реальностью. Многие телевизионные каналы специально готовят передачи для сети и ведут вещание, используя потоковые приложения, такие, как RealOne Player и QuickTime.

Хороший пример сайта, предоставляющего пользователям широкий выбор телепередач — www.bbc.co.uk:



Передача ведется потоковым способом, при этом скорость зависит от пропускной способности линии пользователя. Хотя звучит все это привлекательно, транслируемое изображение зачастую отображается на экране в маленьком окошке, к тому же качество его при низкой скорости соединения оставляет желать много лучшего.

Сказанное относится ко всему веб-вещанию. Музыкальные концерты, мировые новости, отчеты о конференциях, демонстрационные показы транслируются в сети напрямую или в записи. Эти трансляции предназначены, прежде всего, для тех, кто не в состоянии посмотреть соответствующие передачи по телевидению (и призваны, скорее, заинтересовать потенциального зрителя).

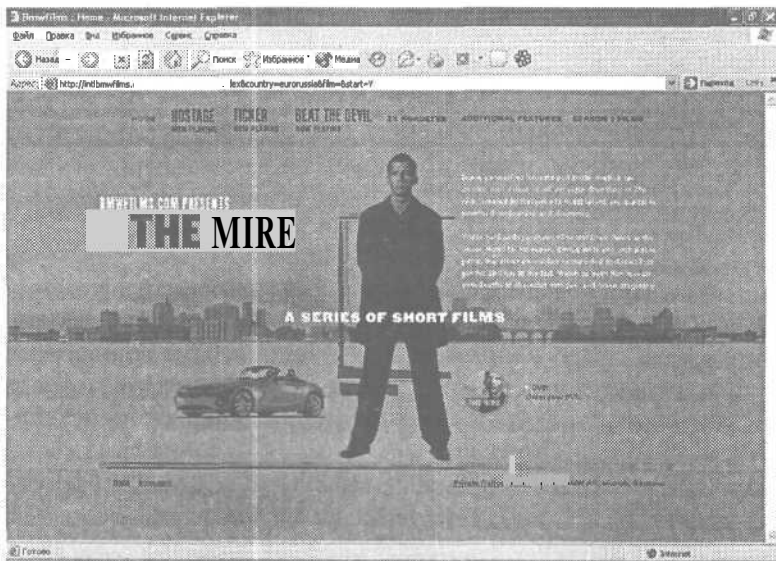
Плохая новость: почему WWW не является новым TV

На протяжении ряда лет шло много разговоров о том, что всемирная паутина будет все ближе и ближе к телевидению. Так отчего же видео и мультимедиа до сих пор не перекочевали в сеть?

Слишком мало пользователей имеет широкополосное подключение

Широкополосное подключение дает возможность передавать информацию на скоростях свыше 512 kbps (килобайт в секунду), тогда как скорость распространенных модемов не превышает 56 kbps. Хотя широкополосная технология появилась еще в 1995 году, в большинстве регионов мира она распространяется отнюдь не так быстро, как ожидалось. Яркий пример — Великобритания, где широкополосным подключением располагают всего лишь 5% пользователей всемирной паутины, тогда как в США — 22%.

Нельзя оставить без внимания и тот факт, что широкополосное подключение доступно, как правило, лишь определенным слоям населения (компаниям, коммерческим учреждениям и т.п.), а потому некоторые сайты заранее ориентируются именно на такую «широкополосную аудиторию». Вот, например, сайт www.bmwfilms.com, безусловно, выполнил свое домашнее задание...



У средней «персоналки» не хватает мощности

Многие настольные машины не обладают достаточной производительностью для воспроизведения высококачественного видео с высокой частотой кадров или сжатых видео-файлов. Сжатие — процесс «упаковки», в результате которого файл уменьшается в размере, тем самым ска-

чивание его из сети происходит быстрее. Некоторые способы сжатия требуют весьма напряженной работы процессора при «распаковке» и последующем воспроизведении видео. Если оборудование пользователя не обладает достаточной производительностью для работы со сжатым видео, то при воспроизведении будут пропускаться кадры, фильм будет прерывистым, «дерганым».

Итак, вы можете себе представить, насколько существенна эта проблема. Если мы предположим, что пользователь имеет модемное подключение на 56kbps, мы имеем основания предположить, что компьютер у него не самый быстрый. Сжатые клипы, имеют достаточно небольшой размер для загрузки, но в то же время их гораздо труднее воспроизвести на слабой машине.

Однозначное решение этой проблемы найти трудно, однако мощность машин постепенно увеличивается — минимальные требования для Flash Player 6 — это Win 95/NP или Mac OS 8.6. В значительной степени помогают найти решение правильное использование параметров сжатия. Поддержка разных параметров для разных скоростей подключения — тоже неплохая идея, как в трейлере *Amelie*, помните?

Нет единого стандарта представления видео

Веб-видео обычно базируется на одном из основных модулей: QuickTime, RealOne Player или Windows Media Player. У каждого из них есть плюсы и минусы (подробнее обо всех видео-модулях — в приложении), но по большому счету все они имеют одну главную задачу: воспроизводить видео-содержание, будучи встроены в браузер пользователя,

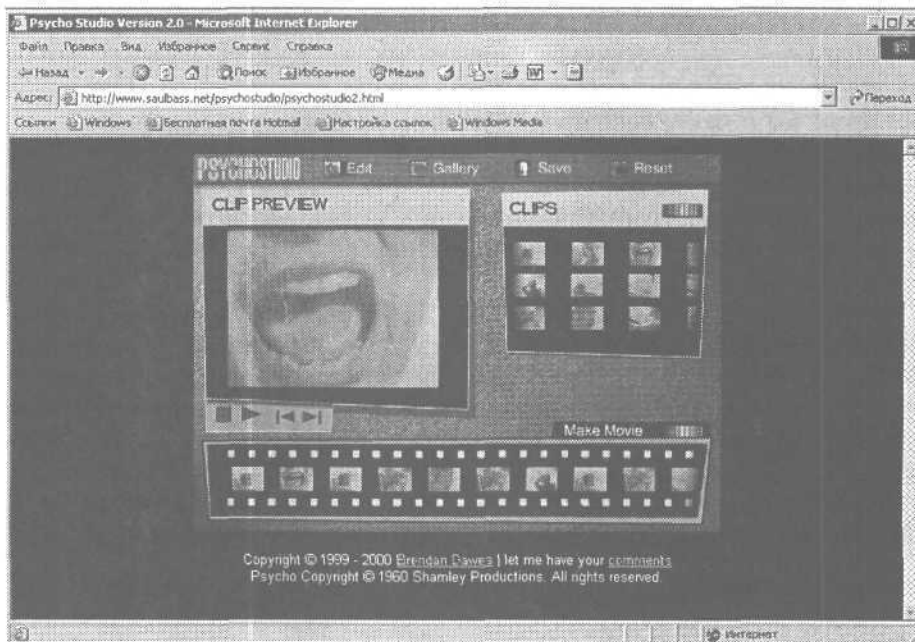
QuickTime входит в комплект поставки для всех «Макинтошей». Для PC, работающих под управлением Windows, стандартным модулем является Windows Media Player, а вот какого-либо межплатформенного стандарта не существует. Так что вы не можете знать заранее, насколько большая часть вашей аудитории будет иметь «правильный», такой же, как у вас, модуль.

Хорошие новости: Flash MX дает ответ

Если вы взялись за изучение Flash впервые и сразу начали с версии MX, вы просто счастливчик. Сколько мастерства требовалось для работы с видео в предыдущих версиях, и вот благодаря Flash MX с возможностью внедрения видео мы избавлены от этих мучений.

Конечно, отдельные разработчики создавали шедевры и при помощи предыдущих версий Flash, как, например, известная программа Брендана Доуза, дающая вам возможность по своему отредактировать зна-

менитую сцену в душе из фильма Хичкока *Psycho* (найти это приложение можно по адресу www.saulbass.net/psychostudio/):



Мы говорили, что возможность прямого внедрения видео во Flash сильно облегчает труд разработчиков, но ведь главное не это, правда? Гораздо важнее, чтобы другие смогли увидеть и оценить плоды наших усилий в области веб-дизайна!

Наконец-то действительно широко распространённый видео-модуль

Поскольку Flash Player способен представить веб-страницу и видеоролик как единый пакет, пользователь избавлен от необходимости загружать из сети модуль одного из других проигрывателей (RealOne, QuickTime и Windows Media). И уж если говорить о загрузке, обратим внимание на факт, который может показаться маловажным, однако в действительности имеет большое значение: Flash Player б значительно меньшего размера (400 kb для PC и около 800 kb для Макинтош), чем любой из упомянутых видео-проигрывателей — к примеру, RealOne Player «весит» 8,52 Mb.

Собственные статистические исследования Macromedia показывают, что Flash Player очень широко распространен, 98,3% всех пользователей «паутины» (436 миллионов во всем мире) имеют какую-либо из версий. По последним данным (на март 2002) около 90% пользователей обладают проигрывателем до версии 5 включительно. Эти цифры однозначно показывают, что ныне Flash — гигант «всемирной паутины». При сравнении со статистикой по трем другим проигрывателям становится очевидным, что на сегодня Flash — самый популярный и распространенный модуль для браузеров. Другими словами, людям больше нравится смотреть веб-видео с помощью Flash Player, чем в любой иной форме.

Еще одна проблема, которую смог преодолеть Flash MX — объединение видео материала с содержанием веб-страницы, ведь Flash внедряет видео прямо в SWF файл. Тем самым появляется возможность простой интеграции, в отличие от неуклюжих решений, когда для просмотра видео материала каждый раз требуется запускать RealOne или Windows Media проигрыватель. Пусть это не покажется вам малозначительным, ведь для пользователя это чрезвычайно удобно! Средний пользователь не должен беспокоиться о конфликтах проигрывателей или временных окнах; информация должна быть доставлена наиболее простым способом — и Flash MX делает это возможным.

Наилучший компромисс между размером файла и загрузкой процессора

ЕСЛИ вас еще не вдохновили новые возможности Flash в работе с видео, вспомните о том, благодаря чему это приложение приобрело известность — о векторной графике и интерактивности. Flash MX дает возможность создавать мультимедийные среды и интерфейсы, комбинируя «увесистое» (говоря о килобайтах) видео и сверхлегкую векторную графику. Он возвещает появление в «паутине» нового средства, объединяющего все разновидности мультимедиа, возможности интерактивности и многослойной композиции информационных материалов.

Комбинирование различных элементов открывает не только новые творческие возможности, но и новые пути в деле достижения компромисса между размером файла и загрузкой процессора: если видео материал идет у вас не все время, то остальные элементы фильма вы можете оптимизировать, исходя из пропускной способности линии.

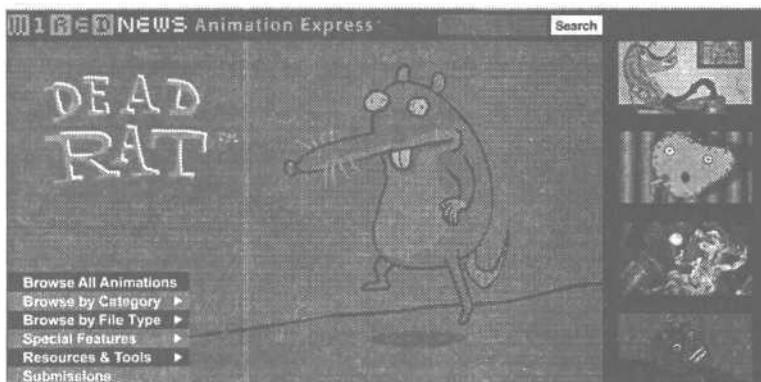
Дни статичных информационных материалов во всемирной паутине сочтены. С Flash MX вы сможете создавать телепрограммы, которые зритель посмотрит, когда пожелает, видеофильмы, в которых зритель сможет выбирать направление развития сюжета. Вы ограничены только воображением!

Возможность обработки видео материала при помощи Flash MX открывает и другие пути, и мы окинем взглядом три из них, прежде чем закончить вступительную главу.

Анимация (мультипликация)

Если вы никогда не имели удовольствия (а также мучений) работать с 16 или 8-миллиметровой кинокамерой, то и ни чему братья за это дело теперь! Ведь если съемка фильма — занятие фантастически увлекательное, то процесс монтажа, когда четыре дня сидишь безвылазно в темном чулане — совсем нет!

Ну, а при чем же здесь Flash и WWW? Любая запись на киноплёнке или ином аналоговом носителе должна быть оцифрована, прежде чем появиться в сети, и после того, как над фильмом поработает великое множество аниматоров и мультипликаторов, он должен быть преобразован в файл, подходящий для «паутины». Hotwired Animation Express (www.animationexpress.com) — один из сайтов, где вы найдете множество оцифрованных мультфильмов наряду с фильмами, созданными при помощи Shockwave и Flash:



Как здорово, что Flash MX может импортировать такой оцифрованный материал, который вы затем можете переделывать по своему вкусу в рабочей среде Flash! Наконец-то можно создать собственный римейк мультфильма *«Кто подставил Кролика Роджера!»*

Образовательные материалы

Flash MX позволяет включать в видеоролики интерактивные элементы, и превосходное применение такой возможности — диалоговые учебные пособия. Flash теперь способен создать готовый CD-ROM, так что уже

не нужно для внедрения видео использовать Director. Flash имеет и такое преимущество, как возможность создания проекционных файлов для обеих платформ, нет необходимости, как с Director, делать для каждой платформы отдельную программу.

Фильмы

Благодаря Flash 5 уже появились свежие творческие находки в деле рекламы кинофильмов. Взгляните на веб-сайты, посвященные фильмам *Requiem For A Dream* (www.requiemforadream.com) или *American Beauty* (www.dreamworks.com/ab/):

омимо того, что, будучи внедренными в ролики Flash, клипы и фильмы легко интегрируются в веб-страницы, они теперь могут быть интерактивными. Вы можете позволить пользователю выбрать финал, или разделить экран и показывать события, развивающиеся параллельно, причем зритель сам сможет выбирать, какая из «половинок» должна сопровождаться звуком в тот или иной момент; в конце концов, вообще даже уйти от линейного развития сюжета, когда взаимодействие со зрителем и повествование будут одинаково важны. Кроме того, Flash поможет легко и быстро создать многоязычные субтитры, поместив информацию в соответствующие переменные или в базу данных, а затем накладывая ее поверх самого фильма.

Заключение

Macromedia Flash MX на сегодня — действительно мощное мультимедийное приложение, допускающее внедрение видео материала. Как вы наверняка поняли из этой главы, веб-видео получило широкое распространение, а с помощью Flash оно может быть представлено наилучшим образом. Flash дает веб-дизайнерам возможность разрабатывать мультимедийные веб-страницы, не заботясь о временных окнах, модулях проигрывателей, совместимости с различными платформами и прочих внешних факторах. Пользователям внедренное видео, поддерживаемое Flash MX, дает возможность просто сидеть и наслаждаться содержанием сайта без надоедливой загрузки модуля проигрывателя и сбивающих с толку переадресаций. И все довольны!

Если этого мало, подумайте о творческих возможностях, которые принес нам Flash MX. В отличие от стандартных веб-видео проигрывателей, Flash MX умеет не только воспроизводить оторванный от всего остального видеоматериал. В этой книге мы поговорим о том, как можно с помощью Flash MX использовать видеоматериал с **выдумкой**, дать волю воображению. Также мы постараемся дать вам некоторые прак-

тические навыки: как создавать компактные Flash-фильмы и видеофильмы; как угодить требованиям той или иной аудитории, учесть различную пропускную способность линий; как обрабатывать видео кадры и манипулировать ими; как импортировать во Flash нужный видеоматериал; и многое другое.

Как мы уже упоминали, информация в книге систематизирована по трем разделам. Часть 1 с помощью нескольких интересных примеров даст вам навыки использования потенциальных возможностей Flash MX по работе с видео. Во 2-й части вы окажетесь в настоящей студии и понаблюдаете изнутри процесс создания видео-проекта Flash MX от начала до конца. Наконец, в части 3 мы углубимся в изучение передовых методик работы с видео, открывающихся благодаря мощи ActionScript, покажем, как создать совершенно особый, удовлетворяющий именно вашим требованиям компонент для воспроизведения видео и звука.

А следующую главу мы начнем с рассказа о редактировании видеоматериалов при помощи некоторых (бесплатных) приложений для обеих платформ — Mac OS и Windows. Ну, поехали...

Подготовка видео материала для Flash

Краткое содержание главы:

- *захват видео и сохранение на жестком диске;*
- *редактирование клипов с помощью Movie Maker и iMovie;*
- *экспортирование материала, готового для Flash.*

Прежде чем погрузиться в изучение Flash, нам нужен какой-либо видео материал. Конечно, можно попытаться перенести «сырую» видеозапись сразу во Flash и там ее обрабатывать, но я бы так делать не советовал — Flash для этого не предназначен, вам лишь придется делать много лишней работы, а итоговый файл получится слишком большим.

Лучше воспользоваться одной из программ, специально предназначенных для обработки видеозаписей. Возможно, прочитав эту фразу, вы уже насторожились — существуют мощные, сложные и дорогостоящие программы для комплексной обработки видео (такие, как Premiere, Final Cut Pro, Avid), есть и книги, полностью посвященные им. Возможно, вы уже знакомы с этими программами, или захотите познакомиться с ними, однако на самом деле нам такая сложность не требуется. В этой главе мы собираемся воспользоваться простыми и — что важно — бесплатными приложениями, которые входят в стандартные пакеты Windows XP и Mac OS.

Не ущемляя ни одну из платформ, мы включили в книгу отдельные руководства по каждому из упомянутых приложений — Movie Maker для Windows XP и iMovie 2 для «Макинтош». Обычно мы советуем пропускать упражнения, но в данном случае лучше будет выполнить только одно — для выбранной вами платформы, а другое пропустить. Общая схема процесса в обоих случаях будет выглядеть так:



Видео материал, который я использовал в этой главе, был отснят на камеру формата miniDV на состязаниях байкеров, проходивших в Великобритании, в закрытом помещении. Я снимал без штатива, без какого-либо плана, имея довольно слабое представление о том, что будет происходить. Конечно, я не советую вести съемку таким образом, но ведь невозможно всегда и все спланировать заранее!



Прежде, чем начать — кое-какие полезные сведения.

Нелинейный монтаж

Если вы когда-нибудь имели удовольствие заниматься монтажом видео или аудиозаписей на аналоговых системах, то непоследовательный способ видеомонтажа станет для вас приятным сюрпризом. В отличие от аналоговых систем, при аранжировке видео способом **нелинейного монтажа** автоматической записи на ленту и визуализации не происходит.

В аналоговых видеосистемах запись копируется с исходной ленты (оригинала, мастер-ленты) на другую ленту, новые сегменты последовательно добавляются в конец записи. Это означает, что вам придется по-

возиться, чтобы добавить сегмент куда-нибудь в середину, не исключено, что придется перемонтировать весь фильм заново.

В системе нелинейного (т.е. непоследовательного) видеомонтажа материалы хранятся на жестком диске компьютера в цифровом виде, и процесс монтажа заключается просто в размещении клипов (эпизодов, отрезков видеозаписи) на временной шкале в нужной последовательности. При таком способе не составляет труда в любой момент поменять клипы местами или поместить дополнительный клип в любое место фильма. Окончательно записать весь фильм целиком вы можете уже после того, как будете удовлетворены результатами монтажа.

Процесс обработки видео на компьютере упрощается не только благодаря возможности непоследовательного монтажа. Такие программы, как Final Cut Pro и Premiere позволяют создавать полноценные фильмы, обходясь без дорогостоящего монтажного оборудования. Сегодня кино может делать каждый, кто располагает цифровой видеокамерой и компьютером.

FireWire и DV

Разработанная компанией Apple технология FireWire (известная также под именами IEEE-1394, iLink и DV In/Out!) избавляет от многих неудобств при работе вне студии. FireWire — это порт, обеспечивающий передачу цифровых данных со скоростью 400 Мб в секунду, что дает возможность его применения для захвата видео с цифровой камеры в непрерывном режиме.

До появления FireWire и DV (Digital Video — цифровое видео) приходилось конвертировать аналоговые видеозаписи на магнитной пленке (например, формата VHS) в цифровые посредством платы видеозахвата. Не говоря уж о более высоком качестве цифрового видео, при оцифровке аналоговых записей возникает много досадных проблем — например, частая потеря сигнала.

Видеокамера формата DV дает качественный, устойчивый сигнал, она удобна в любых условиях — ведь с ней легче управиться (не надо ни дополнительного персонала, ни лишних «жестянок»). Я уж не говорю о том, что записанный на цифровую камеру материал можно редактировать непосредственно, не требуется что-либо проявлять или переписывать.

Прежде, чем мы начнем работу, несколько слов о понятии, общем для всех монтажных программ — о переходах.

Переходы

Переход (transition) — визуальное пересечение, наложение двух отрезков, в противоположность простой «склейке». Давайте на простом примере посмотрим, как это выглядит.

Вообразите, что у нас есть два клипа-эпизода по полторы секунды каждый. Мы склеили их встык, и наш монтаж выглядит примерно так (здесь каждый кадр представляет S секунды видеозаписи):



1 2 3 A B C

...общая продолжительность составляет три секунды, сначала воспроизводится первый клип, затем второй.

А вот тот же самый монтаж с эффектом наплыва («растворения», плавного перехода) — первый клип постепенно тускнеет, а второй проясляется:



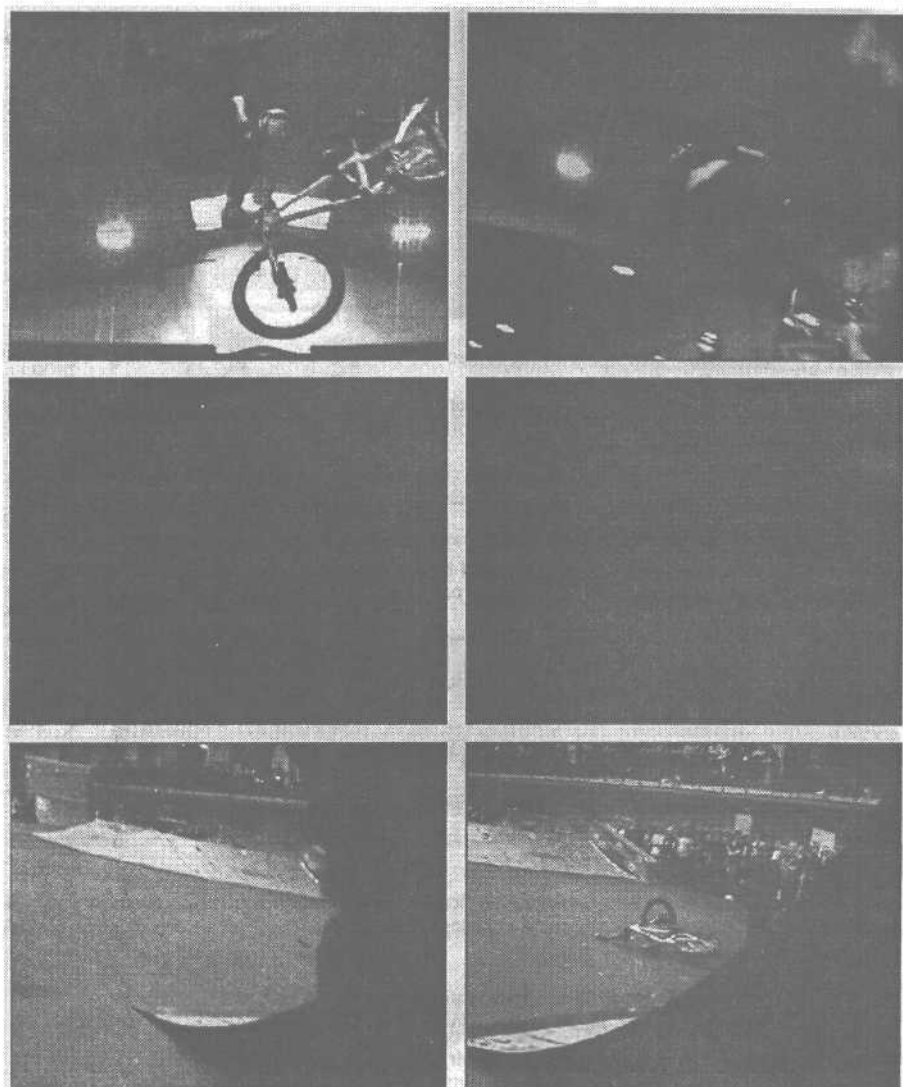
1 2 A 2 B C

Поскольку клипы наложились друг на друга, продолжительность стала меньше, чем в первом случае (две секунды против трех), зато смена фокуса не бросается в глаза благодаря плавному наплыву второго эпизода.

Переходы хороши, однако не стоит использовать их всегда и везде. Например, несколько медленных переходов при показе пейзажей могут создать меланхолическое, созерцательное настроение, хотя во время съемки это отнюдь не планировалось.

С помощью переходов можно также прикрыть кое-какие дефекты. В моем фильме был эпизод, который нельзя было просто вклеить — велосипедист, за полетом которого следила моя камера, пропал из вида, поскольку кто-то загородил мне объектив!

Вот фрагмент оригинального клипа (он продолжается чуть больше трех секунд):





Как видите, в середине передо мной оказалась чья-то спина и я пропустил момент приземления!

Поскольку первая половина этого клипа — до того, как все поглотила тьма — мне понравилась, я решил ее использовать. К сожалению, попытка вырезать из клипа середину ни к чему хорошему не привела, и я применил переход для совмещения со следующим клипом.

То, что получилось, вы можете увидеть, просмотрев файл `dissolve.mov`, либо окончательный монтаж — `bm_x_stylee.mov` (или `bm_x_stylee.avi` для Windows).

Создавать переходы позволяют оба наших приложения, и теперь настало время заняться ими вплотную. Мы начнем с Movie Maker — обладатели «Макинтошей» могут пролистать несколько страниц, пока не наткнутся на заголовок «iMovie для Mac».

Кстати, я использовал систему PAL, поэтому в упражнениях вы увидите 25 fps (кадров в секунду). Учтите, что в системах NTSC используется стандарт 30 fps. Для этих упражнений это не имеет значения, однако различия могут появиться, когда мы начнем работать с Flash — впрочем, мы обсудим это в следующей главе.

Movie Maker для Windows XP

Movie Maker — приложение начального уровня для непоследовательного монтажа, которое поставляется в комплекте установки Windows XP. Опытные пользователи могут счесть его возможности несколько ограниченными, однако Movie Maker содержит все необходимые компоненты для импорта, редактирования, экспорта мини-фильмов и управления ими.

Если у вас есть цифровая камера, можете использовать отснятый материал, если же нет — не переживайте, Movie Maker умеет работать со многими форматами видео для Windows.

Начинаем

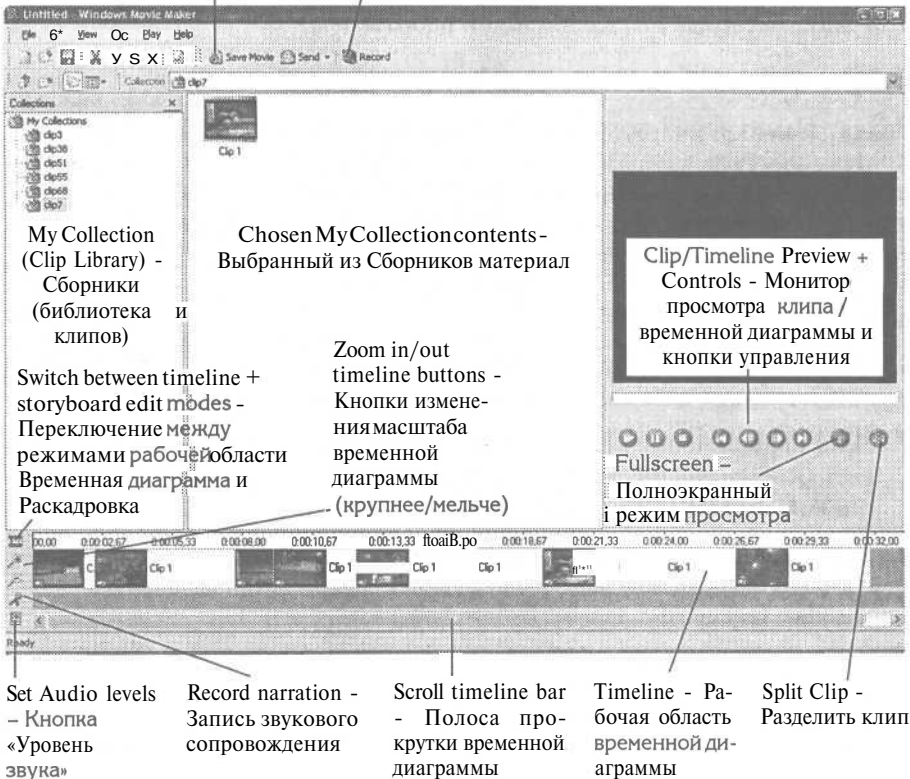
Стандартное расположение Movie Maker — **Пуск > Программы > Movie Maker (Start > Program Files > Movie Maker)**. Если вы не смогли его найти, установите Movie Maker с системного диска XP.

Установили? Тогда запускайте программу и начнем. Окно программы будет выглядеть примерно так:

Как видите, тут не так уж много элементов, и затруднений возникнуть не должно — разработчики старались создать такую среду, которая позволит работать легко и быстро. Я не стану сейчас описывать все опции, а лучше буду давать необходимые пояснения в ходе выполнения упражнения.

Save Movie (Export) -
Сохранить фильм (Экспорт)

Record (from camera) -
Записать (с камеры)



В Movie Maker мы имеем дело с несколькими ключевыми элементами — это сборники, временная диаграмма и монитор. Сборник есть хранилище всех используемых в проекте материалов, отсюда мы выбираем клипы и помещаем их на временную диаграмму — монтажную последовательность. Монитор используется, само собой, для предварительного просмотра клипов или того, что смонтировано на основной временной диаграмме.

Существует два способа ввода видео материала в Movie Maker:

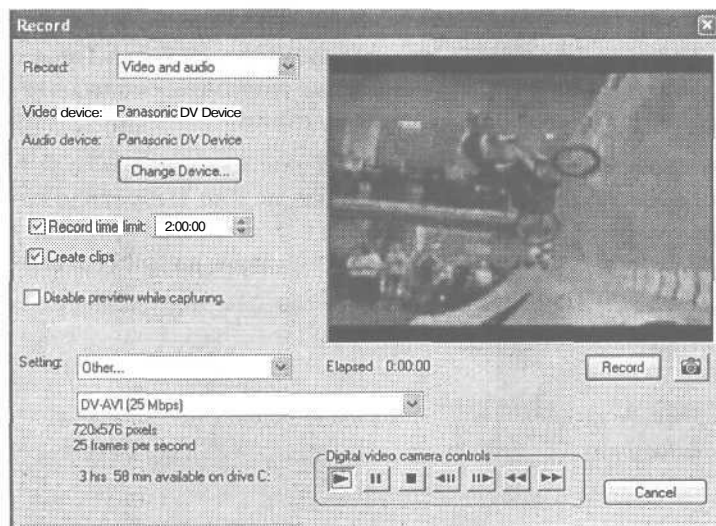
- захват видео с источника: изображение с подключенной к компьютеру цифровой или обычной аналоговой камеры записывается на жесткий диск;
- импортирование файла: попросту находим на жестком диске нужный видео файл и указываем на него Movie Maker, внося в наш проект.

Давайте рассмотрим обе возможности и поместим в наш проект какой-либо материал для дальнейшей работы.

Захват видео

Захват видео через порт Fire Wire при помощи Movie Maker делается проще простого. Fire Wire позволяет управлять цифровыми камерами, поэтому включать воспроизведение, перемотку и т.п. можно прямо с компьютера. Если цифровой камеры у вас нет, и вы собираетесь использовать заранее сохраненный видео материал, *можете* пропустить это упражнение, хотя оно вообще-то небольшое, а эти знания могут вам когда-нибудь пригодиться.

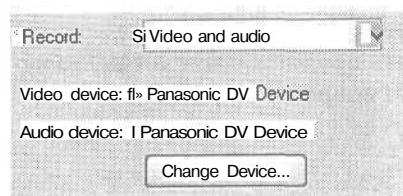
1. Включите вашу цифровую камеру, вставьте кассету с нужной записью и подключите камеру к порту Fire Wire вашего компьютера.
2. Если подключение камеры прошло успешно, нажмите кнопку Записать (Record) на панели инструментов либо выберите команду меню Файл > Запись (File > Record).
3. В открывшемся окне нажмите кнопку воспроизведения:



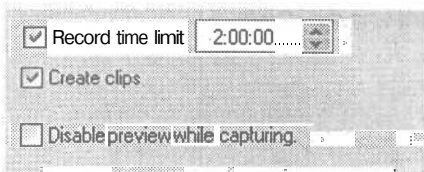
Если камера послушно зажужжала, значит, ваш компьютер успешно установил контроль над камерой и дал ей команду воспроизводить то, что записано на кассете. Если при этом на **мониторе** ничего путевого не отображается, тому может быть несколько причин. Чаще всего они таковы.

- Система Windows XP не распознала камеру. Об этом вы узнаете, взглянув на левую верхнюю часть окна.
 - Лента в кассете перемотана до конца ленты или до конца записи. В таких случаях Movie Maker ведет себя довольно странно: иногда отображает на мониторе какой-то «мусор», а иногда предпочитает вообще ничего не показывать. Нажав соответствующую кнопку, перемотайте ленту назад, и затем снова попробуйте воспроизвести.
4. Если все нормально, и вы наблюдаете записанное изображение, что ж... найдите на кассете что-нибудь интересное, красивое (по крайней мере, нечто такое, что не стыдно показать другим) и нажмите паузу.

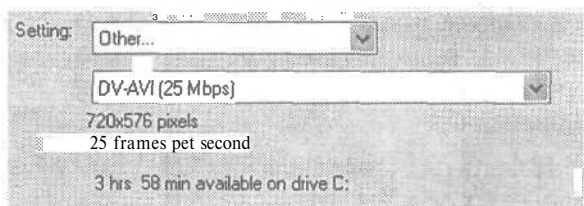
В Movie Maker имеется несколько опций импорта.



- **Запись (Record).** Позволяет выбрать дорожки для импорта — Видео и звук (Video and Audio), Только видео (Video only) или Только звук (Audio only). Лучше отсечь ненужные элементы сейчас, чтобы избавиться от лишней работы впоследствии.
- **Сменить устройство (Change Device...).** Если есть несколько устройств ввода, распознанных Movie Maker, можно выбрать, с какого из них следует импортировать. Это важно только для тех, у кого в системе несколько камер или устройств — например, веб-камера и плата захвата видеосигнала.



- **Ограничить время записи (Record time limit).** Позволяет установить предел длительности импортируемой записи. Это может быть полезно, если нужно, чтобы в вашем проекте все клипы были одинаковой длины, а также в тех случаях, когда нужно импортировать запись большой продолжительности, и вы хотите, пока это происходит, заняться чем-то другим. Movie Maker, очень кстати для таких случаев, показывает в левой нижней части окна, сколько всего видео материала может поместиться на ваш жесткий диск.
- **Создать клипы (Create clips).** Эта установка позволяет до некоторой степени контролировать работу Movie Maker с цифровой камерой. Если флажок установлен, Movie Maker будет создавать новый клип каждый раз, когда получает сигнал с камеры. Сигнал может прерваться во время сеанса записи, может встретиться ошибка или метка, помещенная на ленту камерой. Если флажок **снять**, Movie Maker все это проигнорирует и запишет один большой клип.
- **Disable preview while capturing.** Установка этого флажка позволит снизить нагрузку на процессор в ходе импортирования клипа, это поможет уменьшить вероятность пропуска кадров. Изображение отображаться не будет, но звук при этом будет воспроизводиться, так что вы сможете получить представление о том, до какого места дошла запись, и сможете вовремя остановить ее.



- **Настройка (Setting).** Позволяет настроить качество создаваемого клипа. В верхнем списке имеется четыре пункта: Низкое (Low), Среднее (Medium), Высокое (High) и Другое (Other). Первые три — предустановленные, а вот пункт Другое (Other) позволяет установить собственные параметры.

Кроме особых случаев, я бы рекомендовал при импорте выбрать настройки с наилучшим возможным качеством и без компрессии звука и изображения. Причина проста — желательно, чтобы отснятый материал сохранился в изначальном виде, пока вы будете его обрабатывать. А уж потом, когда вы будете экспортировать готовый фильм, можно будет подумать о компрессии.

На рисунке вы видите, что при импорте своей видеозаписи с цифровой камеры я использовал опцию Microsoft DV-AVI. Этот формат позволяет сохранить качество, наиболее близкое к записи на ленте.

Число, указанное в скобках (в данном случае — 25 Mbps), есть скорость, необходимая для рационального захвата формата записи. Как мы упоминали ранее, FireWire пропускает 400 Mbps, так что вы не рискуете потерять кадры. При использовании более медленных систем захвата — таких, как веб-камера, подключенная к USB, вам, возможно, придется выбрать настройку помедленнее.

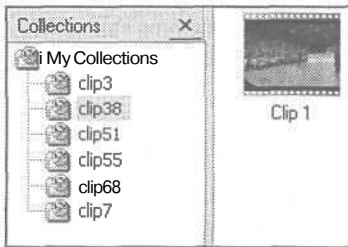
Импортирование файла

Хоть Movie Maker и дает возможность захвата видеосигнала, весьма возможно, что вам захочется — как сейчас, так и в будущем, — использовать материал из других источников: со своего жесткого диска, из сети, с компакт-диска и т. д. Кроме того, Movie Maker позволяет импортировать звуковые файлы и статичные изображения. Это могут быть заставки, титры к вашему фильму, записанная отдельно звуковая дорожка.

Файлы импортируются в Movie Maker командой **Файл > Импорт (File > Import)**. Movie Maker может импортировать следующие типы файлов:



После того, как файл выбран и импортирован, он помещается в библиотеку клипов **Сборники (My Collections)**:



Отсюда клип можно перетащить на временную шкалу, посмотреть или просто так полюбоваться. Клипы помещаются в папку Сборники (My Collections) на постоянное хранение, то есть вы можете создать библиотеку клипов для каждого фильма.

Даже если вы уже получили какой-то материал посредством захвата, давайте попрактикуемся в импортировании файлов.

1. Найдите какой-нибудь клип на своем жестком диске или загрузите клипы для данной главы с веб-сайта friends of ED.

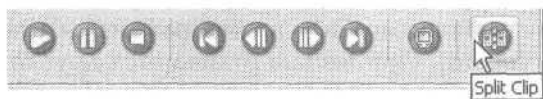
2. После этого выполните команду **Файл > Импорт (File > Import)** и выберите нужный клип. Чтобы выбрать сразу несколько клипов, удерживайте нажатой клавишу **Ctrl** и отмечайте нужные клипы мышью. Попробуйте, по возможности, выбрать 4 — 5 клипов, с которыми вы будете работать.
3. Выбрав клип или клипы, нажмите **Импорт (Import)**. Теперь клипы появились в библиотеке **Сборники (My Collections)**. Если щелкнуть на сборнике в панели библиотеки, клип(ы) появятся на центральной панели экрана.
4. Если вы довольны клипами, которые вы выбрали, сохраните проект **Файл > Сохранить проект (File > Save Project)** в надежном месте и мы начнем уже что-нибудь монтировать.

Работа со сборником клипов

Можно кое-что сделать с клипами еще до начала монтажа, чтобы немного упростить дальнейшую работу. **Movie Maker** позволяет делить и объединять клипы.

Деление клипа на две части

Если ваш клип содержит два отрывка, каждый из которых представляет самостоятельный интерес, лучше представить его как два отдельных клипа. Сделать это легко: щелкните на клипе в сборнике, переместите указатель воспроизведения на панели просмотра к «точке деления» и нажмите кнопку **Разделить (Split Clip)**.



После того, как это будет сделано, в сборнике появятся два клипа с похожими именами:



Clip 1



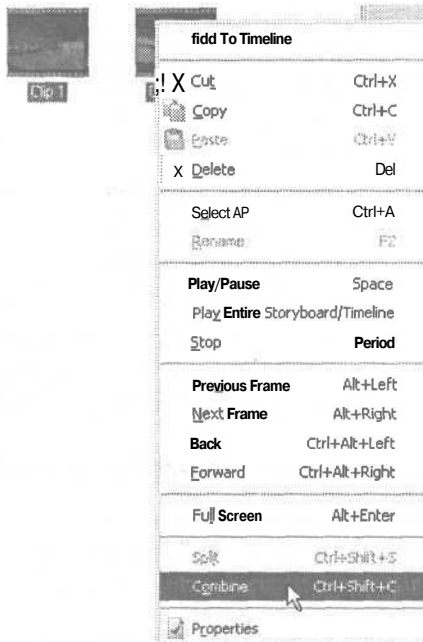
Clip 1(1)

Все просто. Ну, а если вы поспешили и разрезали клип неудачно, как же склеить части опять?

Объединение нескольких клипов

Выбрав клипы, которые нужно соединить, щелчком правой кнопки мыши вызовите контекстное меню, выберите команду Объединить (Combine) — и готово.

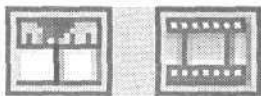
Оба приема позволят вам слегка упорядочить клипы и несколько упростить следующий шаг.



Режимы монтажа в Movie Maker

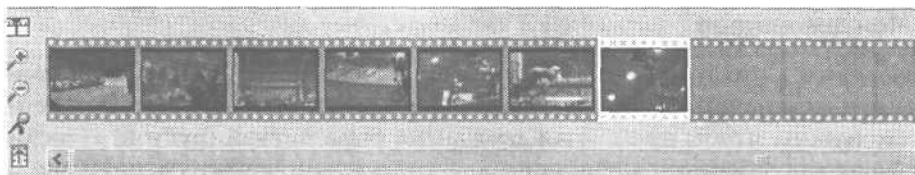
Монтаж в Movie Maker производится в основном посредством «перетаскивания», этот процесс прекрасно знаком пользователям Windows. «Взяли» клип с панели клипов — и «положили» его в нужное место.

Фактически процесс монтажа, расположения клипов в нужном порядке в Movie Maker имеет две разновидности — временная диаграмма (timeline) и раскадровка (storyboard). Они весьма отличаются друг от друга, однако оба чрезвычайно полезны в «технологическом процессе» Movie Maker. Вы можете переключаться с одного режима на другой в любое время в ходе работы над проектом, пользуясь для этого переключателем, расположенным чуть выше и левее рабочей области:



Режим «раскадровка» (storyboard)

В режиме раскадровки клипы располагаются в хронологической последовательности — фильм будет проигрываться по порядку слева направо.

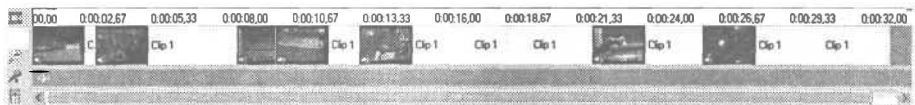


Режим раскадровки великолепен для быстрого монтажа без всяких «хитростей». Это простой процесс: мы перетаскиваем клипы с панели, «укладываем» их в «кадры» фильма, а затем нажимаем кнопку воспроизведения и смотрим, устраивает ли нас такая последовательность.

Этот режим становится непригодным, когда необходима точнейшая обработка, отделка и хронометраж клипов. В этом случае на помощь приходит режим временной диаграммы.

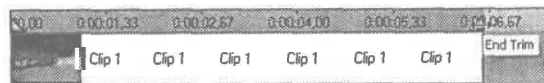
Режим «временная диаграмма» (timeline)

Этот режим позволяет гораздо более тонкий монтаж, чем в режиме раскадровки; дает возможность обрезать клипы и применять некоторые основные типы переходов, обеспечивает контроль над общим хронометражем проекта. Режим временной диаграммы выглядит сложнее, но только из-за способа отображения длительности клипов:



Как и в режиме раскадровки, клипы перетаскиваются на временную диаграмму. При этом Movie Maker проявляет некоторый «интеллект», предлагая поместить клип до или после клипа, уже присутствующего на временной шкале. Затем можно сделать «тонкую настройку» положения клипов на шкале, перетаскивая их уже внутри временной диаграммы. Здесь пригодятся кнопки изменения масштаба (увеличения и уменьшения), которые позволят с большой точностью отрегулировать расположение клипов и переходы.

Кроме того, в режиме временной диаграммы можно немного обрезать наши клипы путем передвижки моментов начала и окончания, а также добавить некоторые основные переходы.



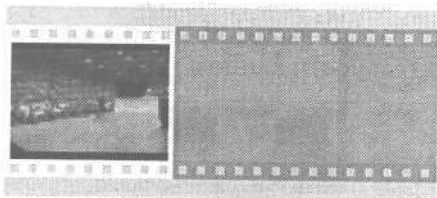
Теперь, когда мы ознакомились с каждым из двух режимов, давайте, наконец, что-нибудь смонтируем!

Монтаж клипов

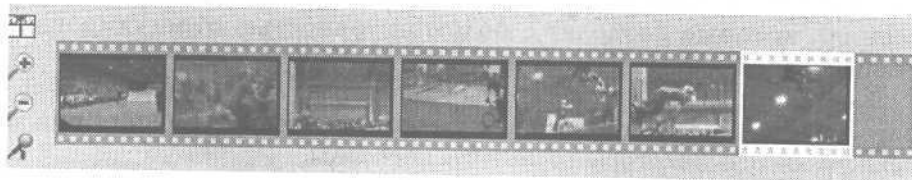
Надеюсь, к этому моменту у вас уже есть несколько клипов, с которыми мы можем начать работать, создавая наш шедевр. Кроме того, нам сейчас нужно, чтобы временная диаграмма была чистой (пустой) — если это не так, уберите с нее клипы: для этого нужно щелкнуть на клипе и нажатием клавиши Delete удалить его.

Итак, наша цель сейчас состоит в том, чтобы набрать четыре-пять клипов и быстро их смонтировать. Не стоит *слишком* задумываться о порядке размещения клипов — сейчас важно просто приобрести навыки работы с Movie Maker. Даже если каждый из ваших клипов является частью некоего связного повествования, все равно не нужно уделять много внимания их порядку.

1. Выберите первый клип из сборника на панели Сборники (My Collections).
2. Установите режим раскадровки и перетащите клип из сборника в первый кадр рабочей области, как показано на следующем рисунке:



3. Вернитесь в сборник и перетащите на рабочую область следующий клип. Переместив его к позиции, следующей за предыдущим клипом, вы увидите маркер, указывающий, куда будет помещен клип:
4. Перетащите еще несколько клипов из сборника, чтобы получить последовательность, ряд из пяти (примерно) клипов.



Что ж, до сих пор мы работали вслепую, но теперь можем посмотреть, как выглядит наша последовательность. Стоит упомянуть, что просматривать свой фильм вы можете когда угодно и сколько угодно, никаких ограничений на это нет, при этом просмотр начинается сразу, без каких-либо задержек.

5. Чтобы просмотреть фильм, выберите команду Воспроизведение > Воспроизвести всю раскадровку или временную диаграмму (Play > Play Entire Storyboard/ Timeline). Вы видите воспроизведение нашей последовательности, фильм, вероятно, имеет кое-какие изъяны, но не переживайте — мы его сейчас подчистим.

Весьма вероятно, что в нынешнем виде наша последовательность (особенно если вы использовали клипы из разных источников) как целое смотрится плохо, и отдельные клипы необходимо обрезать. Если ваши клипы озвучены, вы, к тому же, обнаружите, что это еще более подчеркивает беспорядочный, хаотичный характер вашего первого фильма. Ну, если хотите, можете обвинить в этом меня!

К сожалению, способность Movie Maker отделять звук от видеоклипов несколько ограничена. Пока придется оставить все как есть; впоследствии вы сможете «убить» звук — в процессе экспорта, об этом пойдет речь в следующей главе... ну, можете пока отключить колонки, если звук так мешает!

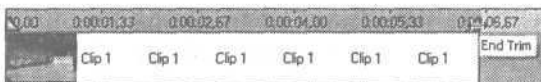
6. Итак, что же мы имеем наданный момент? Мы имеем пять (или сколько там у вас) клипов, размещенных на монтажной доске бессвязно, в беспорядке. Давайте изменим их порядок. Переключитесь в режим раскадровки, затем щелкните на первом клипе, перетащите его в рабочей области на позицию за последним клипом и отпустите.

Наш первый фрагмент стал теперь последним. Ловко, а? Это иллюстрация к словам о том, как прост нелинейный монтаж, от скольких трудностей мы избавляемся благодаря гибкости подобных приложений. Теперь, просмотрев фильм, вы заметите, что некоторые куски надо бы обрезать и укоротить.

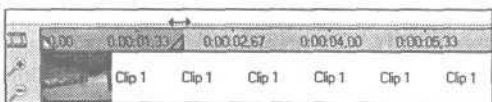
7. Переключитесь в режим временной диаграммы. Здесь мы подрежем нужные клипы и добавим несколько переходов, чтобы избавиться от резкой смены фрагментов.

Работая в Movie Maker, вы можете переключаться с одного режима на другой в любое время — никаких неприятностей это не принесет.

8. Определите, какой из клипов необходимо обрезать и выберите его на временной диаграмме.
9. Переместите курсор к правому краю клипа и чуть выше, к шкале отсчета времени. Курсор примет вид двунаправленной стрелки, показывая, что вы можете увеличить или уменьшить продолжительность клипа:

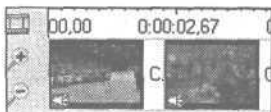


10. Нажмите кнопку мыши и перетаскивайте двойную стрелку. При этом обратите внимание, что на панели просмотра отображается кадр, к которому вы переместили курсор.
11. Действуя таким образом, выберите места обреза клипа с конца и с начала. Теперь Movie Maker сдвинет все остальные клипы так, чтобы не было промежутков. Для укороченных клипов на шкале отсчета времени показывается также их полная продолжительность — обратите внимание на серую полосу, появившуюся на временной шкале.



Если вы хотите автоматически восстановить продолжительность клипа, выделите клип на временной шкале, а затем выполните команду Клип > Убрать точки монтажа (Clip > Clear Trim Points).

12. Продолжая в этом духе, обрежьте и переставьте свои клипы. Не бойтесь оставлять чуть-чуть «лишнего» в начале или в конце, так как мы будем добавлять переходы и эти «лишние» кусочки нам пригодятся. Просмотрите фильм, обращая внимание на внесенные изменения. Про себя отмечайте места, которые требуют дополнительной обрезки или могут быть улучшены при помощи переходов.
13. Теперь мы добавим один из основных переходов — эффект наплыва (взаимного «растворения») в те места, где это необходимо. Чтобы создать такой переход, просто щелкните на клипе и тащите его влево по временной диаграмме, чтобы он частично перекрыл другой клип (помните, переход происходит между окончанием одного клипа и началом следующего):



Не забудьте воспользоваться увеличением, что позволит точнее отрегулировать участок перехода по времени, а закончив, просмотрите, как получился переход.

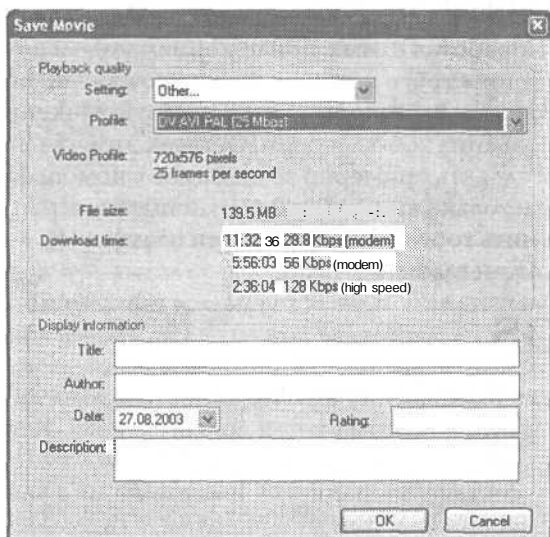
14. Завершите монтаж фильма, используя, по необходимости, переходы и обрезку.

Если вы считаете, что монтаж завершен, нажмите кнопку Во весь экран (Fullscreen) для просмотра. Если хотите добавить в ваш проект заставки и титры, вспомните о том, что в Movie Maker можно импортировать файлы изображений.

Экспорт

Если вы удовлетворены вашим фильмом, мы можем экспортировать его.

1. Нажмите кнопку Сохранить (Save Movie) на панели инструментов. На экране появится следующее диалоговое окно:



Как и при импортировании, Movie Maker предоставляет несколько стандартных настроек и пользовательскую, обозначенную как Другое (Other). Кроме того, Movie Maker весьма кстати подсказывает нам, какого размера фильм получится при таких настройках, и сколько времени займет его загрузка при различных значениях ширины полосы пропускания.

Если мы с вами намереваемся использовать экспортируемый материал во Flash, я бы посоветовал экспортировать с наилучшим возможным качеством (как показано на снимке), но на этот раз давайте выберем среднее — все-таки фильм у нас учебный.

2. Выберите из списка Параметры (Setting) пункт Среднее (Medium). Вы заметите, что размер изображения (в точках) для нашего фильма уменьшится, и размер файла тоже станет меньше.
3. Прежде, чем экспортировать, мы можем разместить в нашем фильме «водяные знаки», чтобы люди знали, кто его создал. Обычно это делают, если фильм предполагается распространять или размещать в сети. Для этого внесите нужную информацию в секции Отображаемые сведения (Display information).
4. Теперь нажмите ОК и немного подождите, пока фильм экспортируется.
5. По завершении этого процесса вам будет предложено просмотреть его. Нажмите ОК и насладитесь плодами своего труда.

Вот и все о Movie Maker. Поработав с этим приложением подольше, вы обнаружите пару-тройку тонкостей, о которых мы не упомянули, но все самое главное было рассказано. Вы теперь — мастер Movie Maker и обладаете могуществом для создания веб-классики. Остаток этой главы посвящен рассказу о том, как сделать примерно то же самое с помощью iMovie для Mac, так что, — если только вы не возжелали испытать странное чувство дежа-вю или сменить горячо любимую вами платформу, — можете пролистать до следующей главы.

iMovie для Mac OS

С введением на «Макинтошах» технологии Fire Wire в «настольном» видео свершилась революция, и iMovie сыграл в ней большую роль. Как вы, возможно, знаете, iMovie был первым приложением для монтажа, поставившимся бесплатно в составе операционной системы, и с той поры он используется и профессиональными дизайнерами, и школьниками, время от времени даже бросая вызов Голливуду.

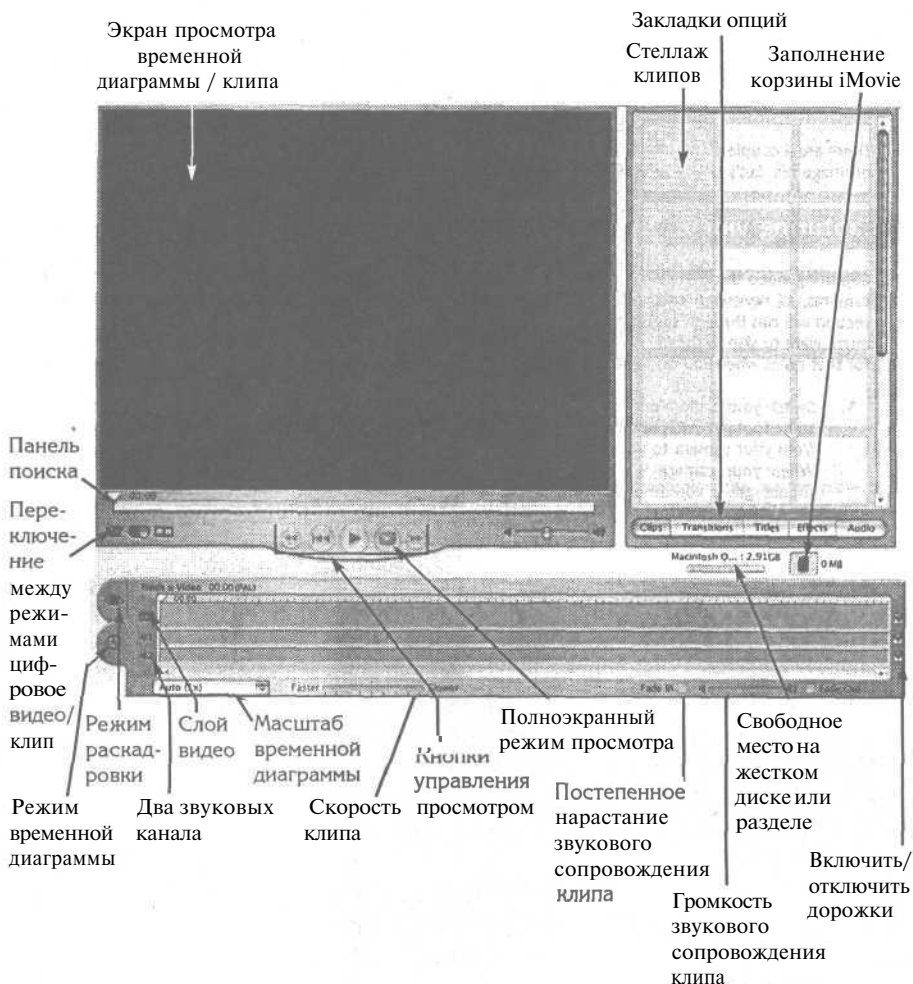
Немного спустя после выхода iMovie появился iMovie 2, а вместе с ним — улучшенные переходы, титры, цифровые эффекты (цветность, контрастность и т.п.), управление скоростью и такая новинка, как реверсирование клипа!

Чтобы не отставать от времени и показать некоторые новые функциональные возможности iMovie, мы будем использовать iMovie 2, однако большинство монтажных работ, которые мы выполним, будут применимы и для iMovie.

Начинаем

ЕСЛИ вы работаете под Classic OS, то iMovie по стандарту установлен на вашем Macintosh HD. Те, кто работает под Mac OS X, найдут iMovie в Macintosh HD > Applications Folder. Если же iMovie у вас еще не установлен, выполните процесс установки, и мы продолжим.

Если все готово, открывайте iMovie, и вы увидите вот что:



Интерфейс, быть может, кажется поначалу немного сбивающим с толку, однако впоследствии вы убедитесь, насколько он прост и будете получать удовольствие от работы с программой! Вместо того чтобы опи-

сывать сейчас все элементы интерфейса, я лучше буду давать необходимые пояснения по ходу выполнения упражнения — тогда вы одновременно увидите, как ими пользоваться и поймете, что они делают.

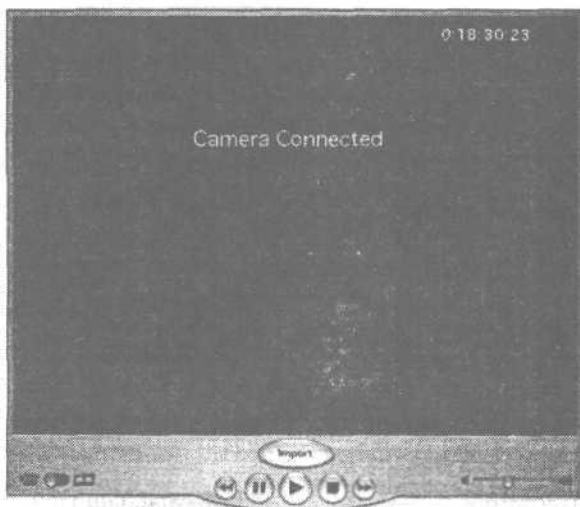
iMovie — приложение, которое можно использовать как для простенького монтажа, так и для чего-то посерьезнее, работа в нем основана на принципе *drag and drop* (перетаскивания). Мы не станем использовать все возможности iMovie, и, тем не менее, сможем манипулировать материалом со всеми удобствами.

Существует два способа ввода видео материала в Movie Maker — захват видео с источника либо импортирование файла, содержащего цифровую видеозапись или изображение. Давайте рассмотрим обе возможности и поместим в наш проект какой-либо материал для дальнейшей работы.

Захват видео

Захват видео через порт FireWire в iMovie выполняется очень легко. Поскольку FireWire позволяет управлять цифровыми камерами, включать воспроизведение, перемотку и т.п. можно прямо с компьютера посредством iMovie. Данный раздел посвящен захвату изображения с цифровой видеокамеры. Если у вас такой нет, можете пропустить это упражнение, а можете остаться с нами и посмотреть, на тот случай, если вам удастся однажды заполнить в свои руки цифровую камеру.

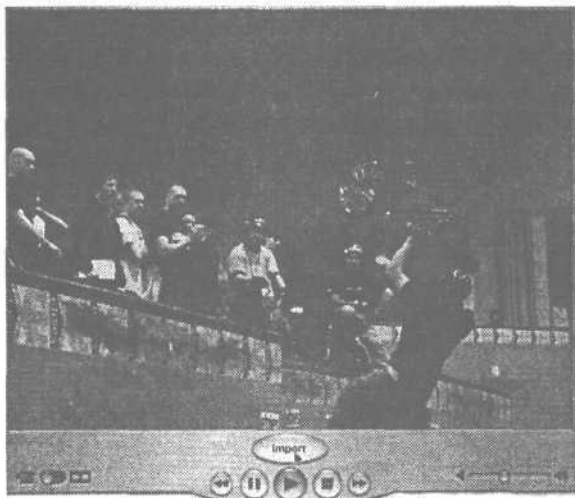
1. Включите вашу камеру, вставьте кассету с нужным материалом и подключите камеру к порту FireWire вашего компьютера. Если подключение прошло успешно, вы получите подтверждение от iMovie, похожее на это:



Если нет, убедитесь, что переключатель в левом нижнем углу панели просмотра установлен на DV — изображение камеры. Если с этим все в порядке, проверьте соединение камеры с FireWire, убедитесь, что камера включена, установлена в режим VCR и готова к воспроизведению.

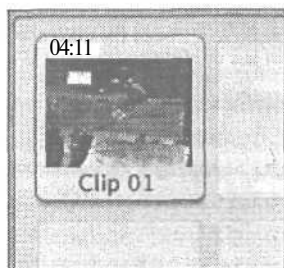
2. Как только получите подтверждение о подключении камеры, нажмите кнопку воспроизведения на панели просмотра. Если камера послушно зажуужжала, значит, ваш Mac установил контроль над камерой и дал ей команду воспроизводить то, что записано на кассете. Если при этом на экране просмотра ничего путного не отображается, похоже, что лента в кассете перемотана до конца записи. Нажав нужную кнопку, перемотайте ленту назад, и затем снова попробуйте воспроизвести. Те из вас, кто работает под версией Mac OS, более ранней, чем OS9, могут также поискать последние обновления FireWire на <http://kbase.info.apple.com>.

Если вы хотите отснять какой-либо материал на камеру, подключенную к компьютеру, просто переключите ее в режим записи и — улыбайтесь!



3. Импорт в iMovie происходит просто — нужно всего лишь нажать крупную кнопку Import под экраном просмотра. Так что когда найдете на кассете что-нибудь интересное, нажимайте эту самую кнопку. iMovie начнет импортировать видео материал в клип, который появится на панели клипов в правой стороне экрана. Продолжительность клипа будет расти в процессе импорта.

- Чтобы остановить импорт, нажмите кнопку Import еще раз. Теперь на стеллаже клипов у вас имеется некий клип:

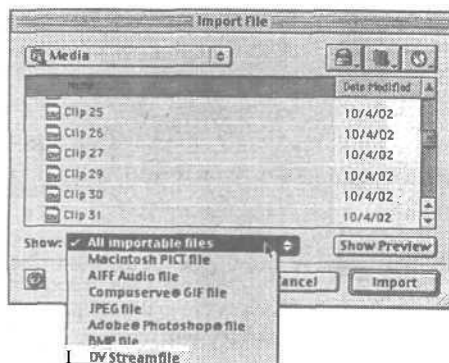


По умолчанию iMovie сам определяет начало нового сеанса записи на вашей кассете и создает новый клип, после чего продолжает импорт.

Хотите — верьте, хотите — нет, но это все о захвате видео посредством iMovie!

Импорт файлов в iMovie

Если цифровой камеры у вас нет, тогда возникают некоторые проблемы, так как iMovie способен импортировать только файлы DV видео, и не воспринимает ни MOV, ни MPEG, ни другие форматы, традиционно поддерживаемые QuickTime. Вот частичный список форматов, которые поддерживает iMovie:



Кроме этих, iMovie может импортировать еще кое-какие типы файлов, например, MP3. Это значит, что в этом списке возможности импорта показаны не полностью. Если вы не уверены, поддерживается ли какой-то конкретный формат, просто попробуйте. Но что касается видео, придется использовать только файлы DV.

Можно конвертировать другие файлы в формат DV. Правда, для этого придется потратиться на QuickTime Pro. QuickTime Pro поставляется в составе установочного пакета надстройки и проигрывателя QuickTime (одна из версий которого предустановлена на всех *Макинтошах*). Для разблокирования Pro требуется приобрести у Apple регистрационный номер. Когда печаталась эта книга, его цена составляла \$29,99.

QuickTime Pro можно будет использовать как утилиту для преобразования файлов из одного формата в другой, в том числе для экспорта в формат DV. Таким образом, вы получите возможность импортировать видео файлы всех форматов, поддерживаемых QuickTime, и конвертировать их в формат, который «понимает» iMovie. Весьма полезное дополнение к вашему набору для работы с цифровым видео.

Ну а здесь мы воспользуемся DV-файлами, входящими в комплект iMovie.

1. В составе iMovie поставляется учебный проект, изобретательно названный iMovie Tutorial; находится он в той же папке, что и iMovie. Выполните File > Import, перейдите в папку обучения (tutorial), и в ней найдите папку Media.
2. Несмотря на то, что файлы в этой папке не имеют расширения DV, все они — DV файлы. Используя Shift, выделите шесть файлов и нажмите Import. Файлы будут помещены на панель клипов, что справа.

Шесть учебных DV файлов — это довольно-таки продолжительные клипы о том, как двое детей купают свою лохматую собаку. Возможно, такой сюжет вам не по вкусу, зато клипы длинные, качественные, и звук очень чистый — отличный материал для наших экспериментов.

Работа с клипами

Теперь, когда у нас есть несколько клипов для работы, давайте подготовим их к монтажу... нет, стоп. Всегда давайте клипам имена! В данный момент это может показаться излишним, однако вы оцените мой совет, когда придется работать с большим количеством клипов, или если вы вернетесь к файлам некоего проекта спустя долгое время. Чтобы переименовать клип, нужно щелкнуть на его имени в библиотеке и ввести новое имя взамен оригинального.

Этот мой VMX-проект — отличный пример. Я импортировал для своей работы 71 клип. Первое, что я сделал — распределил все клипы по категориям, исходя из следующих критериев:

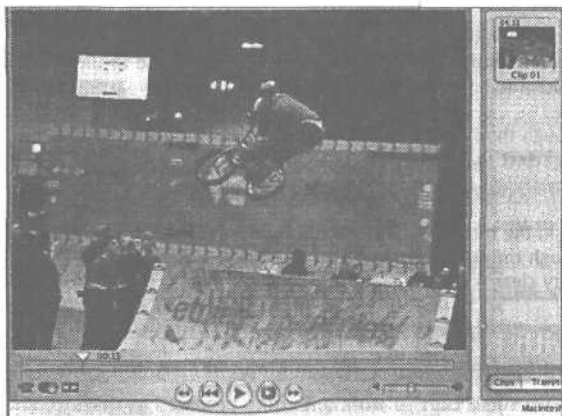


- Место — т.е., желоб, пол, ограждение
- Действие/тип — т.е., спуск, полет, проезд, прыжок, съемка статичной камерой

Конечно, не все клипы подойдут под такие правила, однако сама концепция вполне надежная и помогает намного ускорить работу. Я рассортировал клипы именно по таким критериям, поскольку фильм предполагался многоплановым, и эти метки помогли мне монтировать быстрее, сосредоточиться на композиции, вместо того, чтобы выискивать то статичный кадр, то клип с движением, то еще что-нибудь. Когда имена клипам были розданы, я некоторые из них объединил, а потом, по мере использования, перемещал в конец.

Ну, отступление закончено, теперь приступим к подготовке наших клипов для монтажа.

1. В панели клипов щелкните на каком-нибудь клипе. При этом iMovie переключится в режим клипа (если до этого он был в режиме видеозахвата), и покажет клип на экране просмотра:



На панели просмотра вы можете сделать следующее.

- Воспроизвести / перемотать клип.
- Воспроизвести клип во весь экран.
- Обрезать клип.

Последнее для нас наиболее важно, поскольку дает возможность обрезать ненужный «жир» и оставить только самое вкусное «мясо». Давайте это и сделаем.

2. Определите приблизительно, что вы хотите вырезать из вашего клипа, пользуясь кнопками перемотки, и установите курсор на начало вырезаемого фрагмента. Для перемещения кадр за кадром используйте клавиши-стрелки на клавиатуре.
3. Поместите курсор мыши на временную шкалу под экраном просмотра, нажмите кнопку мыши и тащите. Вы увидите, что появились две белые стрелки, и участок шкалы между ними окрасился в желтый цвет. Так обозначается выделение:



Продолжительность выделенного участка показана прямо над ним.

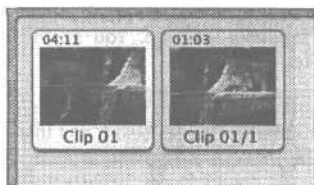
4. Перетащите белые стрелки в нужные точки начала и конца вырезаемого участка.

5. Теперь выполните `Edit > Stop`, чтобы ограничить клип выделенной областью. Вы заметите, что продолжительность клипа, показанная на панели клипов, уменьшилась. Если вы просмотрите его теперь, то увидите только вырезанный материал.

Необходимое предупреждение — поскольку iMovie работает с нашими импортированными клипами, неплохо было бы сохранить резервные копии оригиналов. Лучший способ — дублирование папки проекта, однако это не всегда возможно, если ваш проект слишком велик. Другой способ — создавать копии отдельных клипов в iMovie. Для этого, удерживая нажатой клавишу `OPTION/ALT`, «оттащите» копию от оригинала.

Помимо стандартного обрезания, существует другой способ деления клипов — разрезание на две части. Давайте посмотрим, как это делается.

6. Выберите один из клипов на панели. Его содержимое отобразится на экране просмотра.
7. На панели просмотра переместите курсор к нужной точке деления.
8. Выполните команду `Edit > Split Video Clip at Playhead`, и вместо нашего клипа на панели клипов появится два с похожими именами:



Если вы хотите вернуть клип в исходное состояние, выполните `Advanced > Restore Clip Media`.

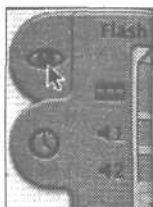
Ну, а теперь, когда мы рассмотрели некоторые способы улучшения нашего необработанного материала, давайте займемся обещанным мною монтажом.

Режимы монтажа в iMovie

Как уже было сказано, монтаж в iMovie представляет собой в основном процесс «перетаскивания». «Берем» клип с панели клипов — и «кладем» его в нужное место.

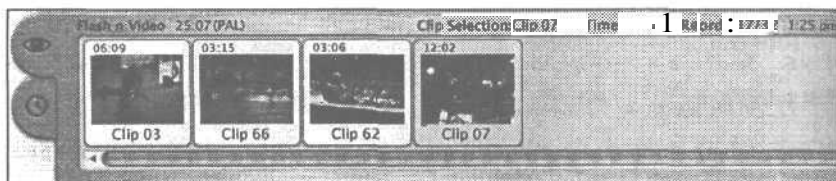
Фактически процесс монтажа, расположения клипов в нужном порядке в Movie Maker имеет две разновидности — временная диаграмма (timeline) и раскадровка (storyboard). Хотя они весьма отличаются друг

от друга, однако оба чрезвычайно полезны в «технологическом процессе». Вы можете переключаться с одного режима на другой в любое время в ходе работы над проектом, пользуясь для этого двумя закладками слева от рабочей области:



Режим «раскадровка» (storyboard)

Режим раскадровки позволяет расположить клипы в хронологической последовательности (в фильме они будут идти по порядку слева направо):



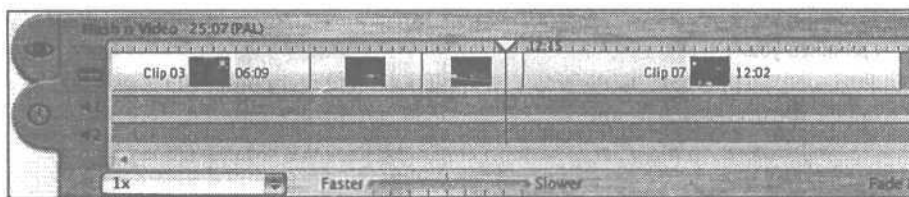
Как видно на снимке, iMovie отображает кое-какую полезную информацию о выбранном клипе (в любом из двух режимов монтажа) непосредственно над последовательностью клипов. Вы можете найти весьма полезной эту информацию, ведь она помогает вспомнить детали, если процесс монтажа занимает долгое время.

Режим раскадровки великолепен для быстрого монтажа без всяких «хитростей». Просто перетаскиваем клипы с панели, «укладываем» их в «кадры» фильма, а затем нажимаем кнопку воспроизведения и смотрим, устраивает ли нас такая последовательность.

Этот режим становится непригодным, когда необходима точнейшая обработка, отделка и хронометраж клипов. В этом случае на помощь приходит режим временной диаграммы.

Режим «временная диаграмма» (timeline)

Этот режим позволяет гораздо более тонкий монтаж, чем в режиме раскадровки; дает возможность обрезать клипы и применять некоторые основные типы переходов, обеспечивает контроль над хронометражем проекта. Режим временной диаграммы выглядит посложнее, но только из-за способа отображения длительности клипов:



Как и в режиме раскадровки, клипы перетаскиваются на временную диаграмму. При этом iMovie проявляет некоторый «интеллект», предлагая поместить клип до или после клипа, уже присутствующего на временной диаграмме. Впоследствии можно сделать «тонкую настройку» положения клипов на шкале, перетаскивая их уже внутри временной диаграммы. Меню выбора масштаба (внизу) позволит с большой точностью отрегулировать расположение клипов и переходов.

Добавить переходы в iMovie можно в любом из двух режимов.

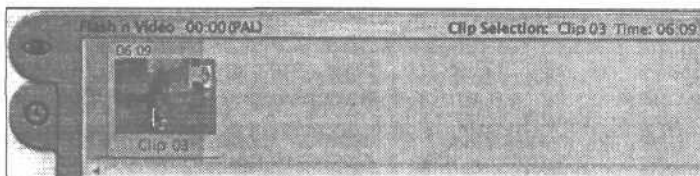
Теперь, когда мы ознакомились с режимами монтажа, давайте, наконец, что-нибудь смонтируем!

Монтаж клипов

Надеюсь, к этому моменту у вас уже есть несколько клипов, с которыми мы можем начать работать, создавая наш шедевр. Кроме того, нам сейчас нужно, чтобы временная диаграмма была чистой (пустой) — если это не так, уберите с нее клипы: для этого нужно щелкнуть на клипе и нажатием клавиши Delete удалить его.

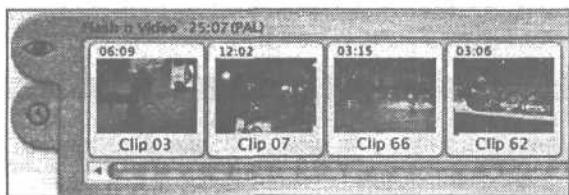
Итак, наша цель сейчас состоит в том, чтобы набрать четыре-пять клипов и быстро их смонтировать. Не стоит слишком задумываться о порядке размещения клипов. Если ваши клипы озвучены (наподобие учебных файлов iMovie), все равно не нужно уделять много внимания их порядку, сейчас важно просто приобрести навыки работы с iMovie.

1. Переключитесь в режим раскадровки. Сейчас мы создадим базовую компоновку из клипов, а потом, в режиме временной диаграммы, выполним тонкую настройку и добавим переходы.
2. Первым делом, надо обрезать наши клипы (не забудьте сначала скопировать их, чтобы сохранить оригиналы). Пару клипов обрежьте так, чтобы остался небольшой излишек в начале и в конце — эти «зазоры» мы используем при создании переходов.
3. Выберите клип из панели клипов и перетащите его на рабочую область раскадровки. При перетаскивании раскадровка будет подсвечена, пока вы не «отпустите» клип.



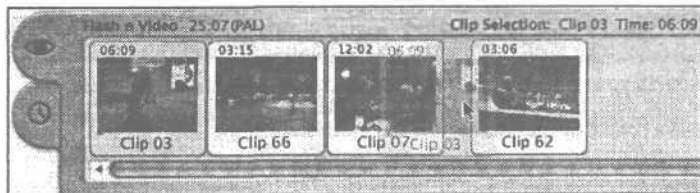
Обратите внимание, что когда вы помещаете клип в раскадровку или на временную диаграмму, он исчезает из библиотеки. Чтобы клип остался в библиотеке, при перетаскивании удерживайте нажатой клавишу **OPTION/ALT**. Если первый клип вы уже потеряли, отмените перемещение с помощью команды **Edit > Undo**.

4. Повторите шаг 3 несколько раз, чтобы получить последовательность клипов:



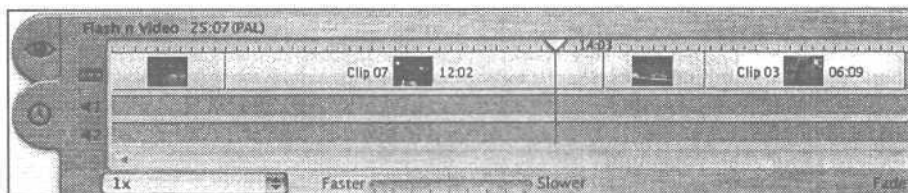
Теперь посмотрим, как выглядит наша последовательность. Стоит упомянуть, что просматривать свой фильм вы можете когда угодно и сколько угодно, никаких ограничений на это нет, при этом просмотр начинается сразу, без каких-либо задержек.

5. Проследите, чтобы ни один клип не был выделен, и нажмите кнопку воспроизведения. Итак, предварительный просмотр нашего фильма — вернее, последовательности раскадровки выглядит, вероятно, несколько убого, но не переживайте — мы его быстро подчистим.
6. Давайте сместим некоторые клипы, немного изменим последовательность. Выберите первый клип раскадровки и перетащите его в промежуток между третьим и четвертым клипами:



Призрачный клип так напугал другие клипы, что они шарахнулись в разные стороны!

7. Снимите выделение со всех клипов и воспроизведите весь фильм. Даже если вы обрезали клипы с удивительной точностью, наверняка получилось так, что от одного клипа к другому сюжет как бы «скачет». Здесь нам на помощь приходят переходы (transitions).
8. Давайте переключимся теперь в режим временной диаграммы. Для этого щелкните на изображении часов слева от раскадровки. Вы увидите те же самые клипы, только они растянуты или сжаты в соответствии с продолжительностью каждого:



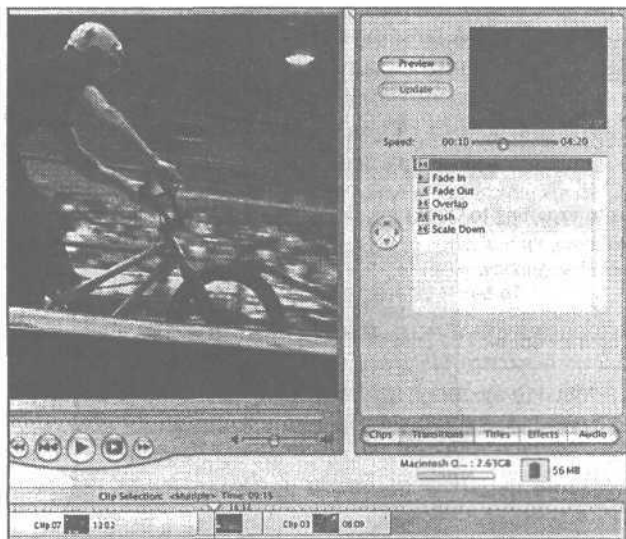
Первым делом надо избавить наши уши от шума. К тому же, шум мешает воспринимать визуальный ряд нашего монтажа, поэтому давайте-ка вычистим звук из наших клипов.

9. Выделите все клипы на временной диаграмме, нажав комбинацию клавиш **APPLE + A** или удерживая нажатой клавишу **SHIFT** и щелкая на каждом клипе.
10. Выполнив команду **Advanced > Extract Audio**, мы удаляем звуковые дорожки из клипов и помещаем их в звуковой слой на временной диаграмме:

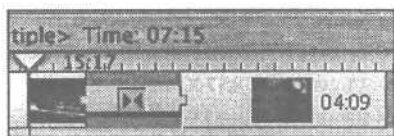


11. Пока аудио клипы выделены, нажатием клавиши **BACKSPACE** удалите их.
12. Еще раз воспроизведите весь фильм — теперь должно быть гораздо легче сконцентрироваться на последовательности.

13. А теперь возьмемся за переходы! Выберите два соседних клипа на временной диаграмме и нажмите кнопку Transitions. Появится маленький список переходов, а также регулятор скорости Speed и экранчик просмотра:



14. Предоставляю вам самостоятельно поэкспериментировать с различными переходами. Я же в данном случае хочу использовать переход cross dissolve (эффект взаимного растворения). Не снимая выделения с двух клипов, выберите переход Cross Dissolve. Вам покажут «превьюшку» (просмотр), но настолько маленькую, что вы, если в этот момент моргнете, можете ее и не заметить.
15. Можно вывести просмотр и на основной экран. Для этого достаточно нажать кнопку Preview. При этом много кадров будет пропущено, но это имеет значение для просмотра, не для самих клипов.
16. Перемещая регулятор Speed, попробуйте несколько вариантов. Цифры показывают продолжительность переходного эффекта.
17. Если вы все отрегулировали, просмотрели, все вас устраивает, пусть теперь iMovie сформирует переход. Берем переход Cross Dissolve и перетаскиваем его на временную диаграмму.
18. Отпускаем в промежутке между двумя клипами, и на временной диаграмме появляется такой вот символ сцепки:



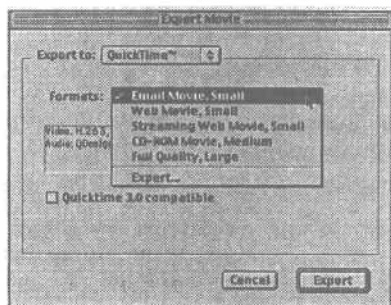
Пока iMovie формирует эффект, отображается красно-черный индикатор выполнения. Как только процесс завершится, нажмите кнопку воспроизведения и насладитесь созданным вами переходом.

Теперь я вас на некоторое время оставлю — закончите монтаж, пользуясь изученными приемами, а когда все будет готово — просмотрите фильм в полноэкранном режиме. Чуть позже мы узнаем, как озвучить наш фильм.

Экспорт фильма

Если вы удовлетворены вашим фильмом, мы можем опубликовать его, чтобы затем можно было импортировать во Flash (или сразу разместить на CD, DVD или в Сети). iMovie экспортирует в формат QuickTime, так что в итоге вы получите файл MOV, совместимый с большинством надстроек и систем.

При экспорте в QuickTime имеется несколько стандартных предустановленных режимов,

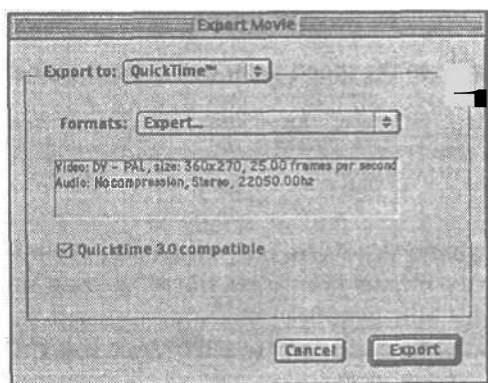


а также пользовательский режим, обозначенный как Expert. Стандартные режимы предусматривают сжатие и уменьшение физического размера кадра.

Если вы импортируете ваше видео во Flash, я бы порекомендовал сохранить фильм в формате, обеспечивающем наилучшее качество, поэтому используйте, по возможности, режим Full Quality, Large. При этом файл сохранится в формате DV, без сжатия, с тем же размером и частотой кадров.

Если вы хотите записать ваш фильм на цифровую камеру, это можно сделать, выполнив команду File > Export Movie и настройку Export to Camera. Это удобно, если вы собираетесь распространять видео в других форматах, таких, как VHS.

Однако при импорте во Flash может оказаться, что файл великоват. Если так, используйте в iMovie режим Expert, уменьшите размер кадра и используйте DV-NTSC или DV-PAL (в зависимости от вашей системы) в качестве компрессора. Качество при этом сохранится, но файл станет меньше за счет уменьшенного размера кадра — теперь импортировать в Flash он будет заметно быстрее. В показанном примере (на снимке) я уменьшил физический размер кадра вдвое:



Все же для нашего фильма давайте пока выберем один из средних режимов — скажем, CD-Rom Movie Medium. Нажмите кнопку Export и сохраните файл. Немного подождя, найдите этот файл и откройте, чтобы просмотреть. Если все нормально, оставьте файл наготове для следующей главы, если нет — выбросьте в помойку и начните все сначала. Врядли в следующей главе вам захочется работать над материалом, который вызывает лишь чувство досады.

Итак, если вы хотите добавить в фильм музыку, звук и другие украшения, читайте дальше — сейчас мы этим займемся.

Прочие возможности и трюки iMovie

С выходом iMovie 2 в программе появилось несколько новых функций, расширяющих возможности монтажа. Давайте посмотрим.

Направление клипов

ViMovie 2 имеет возможность реверсировать клип или последовательность клипов — чтобы изображение и звук воспроизводились «задом наперед». Для этого нужно, выбрав клип из библиотеки или временной диаграммы, выполнить команду **Advanced > Reverse Clip Direction**. Реверсированный клип в библиотеке обозначается направленной влево стрелкой (вверху справа):

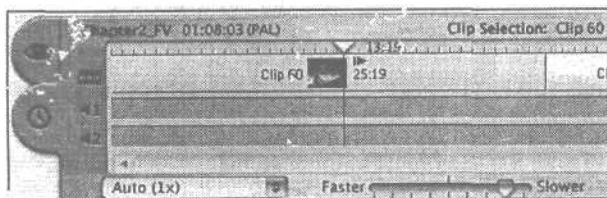


Для восстановления нормального направления воспроизведения клипа, выполните команду **Advanced > Restore Clip Media** или просто реверсируйте его еще раз.

Скорость клипов

Клипы на временной диаграмме можно ускорить или замедлить (коэффициент ускорения или замедления может достигать пяти). Хотя я не пользовался этим эффектом в своем видеофильме ВМХ (а это было бы здорово в клипах с полетами), он великолепен для драматичных, быстро развивающихся сцен. Вспомните всевозможные взрывы в кинобоевиках.

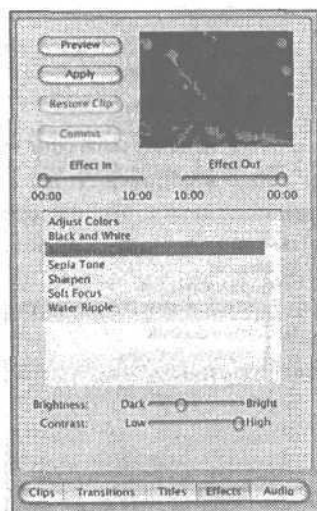
Выберите клип на временной диаграмме, затем воспользуйтесь движковым регулятором внизу для увеличения или уменьшения скорости воспроизведения этого клипа:



Также можно получить удивительный эффект путем замедления всевозможных звуков, голосов (как у Дэвида Линча в «Eraserhead» или «Blue Velvet» — звуковые эффекты (всевозможные вопли и крики) созданы путем замедления и ускорения голосов животных). Не забудьте, что вы можете извлечь звук из клипа, чтобы обработать его отдельно (подробнее об этом чуть позже).

Специальные эффекты

ЕСЛИ вы хотите каким-то особым образом стилизовать клип, или просто сделать цвета поярче, тогда к вашим услугам закладка **Effects**. Здесь много возможностей для манипуляций с клипами, от полезного всем фильтра **Brightness/Contrast** (яркость/контраст) до эффекта **Water Ripple** (интересно, кому может такой понадобится):



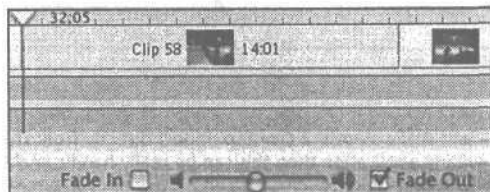
Можно получить дополнительные эффекты и фильтры, скачав пакет **iMovie 2 Plug in Pack** (только для Mac OS X) по адресу www.apple.com/imovie/enhance.html.

Чтобы применить эффект, выберите клип (или несколько) на временной диаграмме, перейдите на вкладку **Effects** и выберите эффект из списка. Для большинства эффектов можно отрегулировать интенсивность, и тут же увидеть, что получается, в просмотровом окошке. Нажатием на кнопку **Preview** запускается просмотр в основном окне, при этом можно будет установить временные рамки применения эффекта (на снимке мой эффект применен ко всему клипу целиком). Изменение приведет к тому, что в определенное время эффект постепенно будет ослабляться, пока не исчезнет.

Поэкспериментируйте с этими эффектами, определите, какие из них могут вам пригодиться. Поначалу может показаться, что во всех этих тонкостях не просто разобраться, однако со временем, привыкнув, вы станете обращаться с ними вполне свободно, даже с лихостью.

Звук: громкость и фединг

iMovie 2 позволяет управлять федингом (fading, постепенное нарастание или затухание) и громкостью звука в клипах:



Выберите клип на временной диаграмме, при этом органы управления звуком будут установлены на текущие для этого клипа значения. Фединг включается и выключается просто установкой или снятием соответствующих флажков — Fade In (нарастание) и Fade Out (затухание). Для установки громкости служит движковый регулятор. Как видите, можно включить одновременно и нарастание, и затухание. Эффекты применяются сразу же и запоминаются, если вы вернетесь к этому клипу впоследствии, то увидите на временной диаграмме те же самые установки.

Импорт звука с CD

Если вы озаботились звуковым сопровождением, и вам хотелось бы использовать в своем проекте музыку, вы можете импортировать аудио записи с компакт-диска. Вставьте CD, и список его треков (песен) появится на вкладке Audio:



Чтобы импортировать часть трека, нажмите кнопку воспроизведения для прослушивания звука и кнопку Record Music для его захвата. В звуковом слое временной диаграммы появится увеличивающаяся полоса пурпурного цвета. Чтобы импортировать CD трек целиком, просто перетащите его в звуковой слой временной диаграммы. Появится длинная пурпурная полоса.

Если вы собираетесь использовать в своем проекте как музыку, так и другое звуковое сопровождение, удобнее будет разделить их. Попробуйте использовать аудио слой 1 для звуков, а слой 2 для — музыки. Так будет гораздо легче разобраться, где что лежит, и не придется отвлекаться от работы, чтобы поиграть в игру под названием «угадайка».

Запись с микрофона

Помимо CD треков, файлов MP3 и AIFF iMovie может получать звук прямо с микрофона, подключенного к вашему компьютеру. Это может быть удобно, если у вас нет специальной программы для записи и обработки звука — воспользовавшись разъемом для подключения микрофона, вы сможете записать голос или еще что-либо.

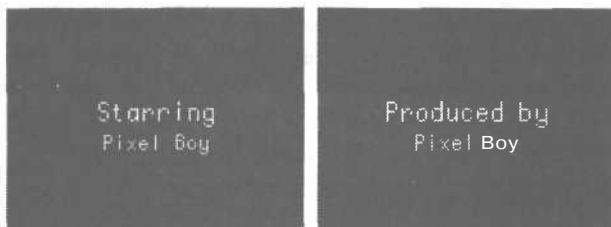
Если ваш компьютер модели iMac G3, знайте, что вон то маленькое отверстие в центре корпуса вашего монитора и есть разъем для микрофона. Может, стоит подумать о его использовании при работе с iMovie?

Чтобы записать звук с микрофонного входа, перейдите на вкладку Audio и нажмите кнопку Record_Voice. Вы увидите две «прыгающие» полоски уровня записи, а на временной диаграмме, в слое звука, появится и станет увеличиваться оранжевая полоса.

Титры

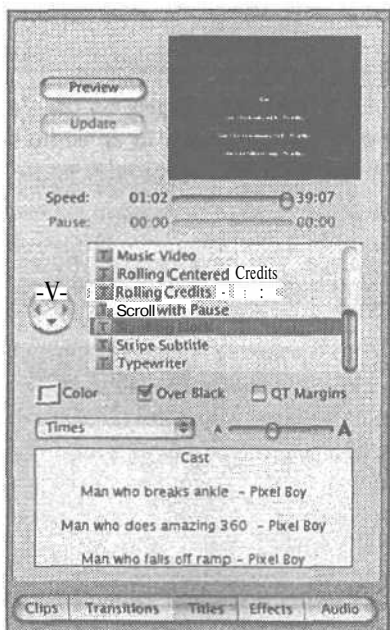
Закончив создание шедевра, следует прославить свое имя, указав его в титрах в начале и в конце фильма. Обычно дают несколько отдельных титров с перечислением главных создателей,...





... в таком количестве, которое вы сочтете достаточным для самовосхваления!

На вкладке Titles вы найдете много всего. Разнообразные эффекты, опции и старое доброе окно просмотра:



Титры можно пустить поверх клипа, на черном фоне или на фоне импортированного изображения. Окно в нижней части вкладки — самое важное: здесь вы вводите нужный вам текст. Чуть выше — опции выбора шрифта: начертание, размер, цвет. Это может показаться тривиальным, но выбор шрифта тоже важен для придания вашему фильму ощущения, так сказать, стилистической целостности — согласитесь, шрифт «Times» слишком классический для такого ролика, как мой ВМХ.

Еще более важно подобрать эффект для ваших титров. Некоторые виды эффектов вполне соответствуют традициям — такие, как Scrolling Block для финальных титров, или Music Video для подачи информации в стиле MTV, однако есть и другие.

Если вы озадачены кружком со стрелками, напоминающим компас, объясняю, что с его помощью задается направление эффекта. Покрутите его — нечто нетрадиционное вам обеспечено.

Чтобы пустить титры по черному фону, проверьте, чтобы ни один клип не был выбран, выберите эффект и введите текст. По необходимости изменяйте параметры, смотрите, что получается, пока не добьетесь того, что нужно. Затем перетащите эффект на временную диаграмму. Тем самым вы создадите новый клип, так что программе потребуется немного времени на обработку.

Чтобы поместить титры поверх клипа, выберите клип, отрегулируйте нужный эффект. Как только с настройками будет покончено, перетащите эффект на временную диаграмму и поместите его перед клипом.

После обработки вы получите нечто вроде этого (здесь я использовал эффект Music Video):



*Чтобы сделать титры немного шире, можно уменьшить поля кадра. Для этого установите флажок *QuickTime margins*.*

Если вам захотелось сделать что-то более вычурное, тогда, возможно, лучше будет создавать титры отдельно — например, во Flash, — поскольку в iMovie титры предназначены для достаточно общих несложных проектов.

Заключение

Теперь, когда вы получили представление о монтаже, предлагаю вам позаниматься самостоятельно. Убедитесь, что процесс создания фильма для вас не содержит больше тайн, что с инструментами вы обращаетесь достаточно непринужденно. В следующей главе мы импортируем наши видеоролики во **Flash**, и вам, возможно, было бы интереснее работать со своим собственным материалом.

Перенос видео материала во Flash

Краткое содержание главы:

- форматы видео, какие из них можно импортировать во Flash;
- установка параметров импорта видео во Flash;
- скорость передачи данных и интервал ключевого кадра;
- экспорт видео в формате FLV;
- экспорт видео с использованием Sorenson Spark;
- сравнение Sorenson Spark и Sorenson Squeeze;
- оптимизация видео.

Итак, у вас есть некий видеоролик, смонтированный и отредактированный, и вы готовы импортировать его во Flash, верно? Ну... я бы сказал, почти готовы! Прежде чем мы обратимся к импорту видео, вам следует узнать еще кое-что. В этой главе мы рассмотрим способы получения максимальной отдачи от вашего исходного материала путем оптимизации — уменьшения размера файла при сохранении, по возможности, наилучшего качества.

Результаты оптимизации могут как огорчить, так и обрадовать. При небольшом изменении той или иной настройки может получиться маленький файл с приемлемым качеством изображения, а может — огромный, да еще и качество будет отвратительным. Чтобы достичь наилучших результатов, нужно вникнуть в смысл таких понятий, как скорость передачи данных, ключевые кадры, методы компрессии.

Прежде всего

Macromedia Flash MX сможет импортировать видео форматы только в том случае, если в вашей системе уже имеются соответствующие проигрыватели, поэтому позаботьтесь об их установке заранее, иначе что-нибудь в нужный момент не сработает.

Необходимо установить QuickTime версии 4 (можно получить с www.apple.com/quicktime) или выше для импортирования следующих форматов (для обеих платформ, Windows и Macintosh):

- AVI — Audio Video Interleaved (переменяющиеся аудио- и видеоданные);
- DV — Digital Video (цифровое видео);
- MPG или MPEG — формат, разработанный Motion Picture Expert Group;
- MOV — QuickTime Movie (фильм QuickTime).

Пользователям Windows необходимо установить DirectX версии 7 или выше (последнюю версию можно получить с www.microsoft.com/windows/directx/):

- AVI;
- MPG или MPEG;
- WMV или ASF — файл Windows Media.

Многообразие форматов сжатия видео и аудио может привести к тому, что первый результат импортирования видео во Flash окажется обескураживающим, однако не переживайте — мы поможем вам преодолеть все трудности.

Первое, о чем следует помнить: очень желательно, чтобы файл-источник был в несжатом формате. Нет, это не опечатка! Если нужно сжатие, то удобнее использовать встроенный компрессор Flash — Sorenson Spark, который превосходно выполнит компрессию любого видео. Sorenson Spark — это так называемый кодек (англ. codec, сокращение от «компрессор-декомпрессор») для видеофильмов.

Тем, кто хотел бы получить еще больший контроль над размером и качеством импортируемого видео файла, стоит обратить внимание на продукт под названием Sorenson Squeeze. Возможно, вы уже видели рекламу Squeeze на сайте Macromedia; это приложение разработано специально для сжатия видео во Flash MX. Подробнее о Squeeze поговорим позже, а сейчас я познакомлю вас с многочисленными опциями, которые вы сможете использовать при импорте видео во Flash.

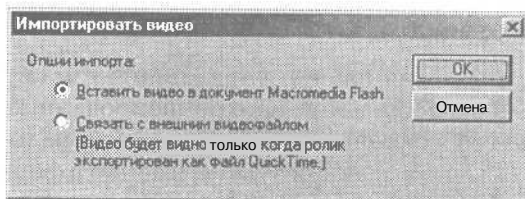
Импортирование видео: подготовка

Источником, или оригиналом, видео для нашего упражнения служит файл в стандартном формате для видеофильмов, синхронизированных со звуком — Audio Video Interleaved, сокращенно AVI. Профессиональные программы видеомонтажа обычно выдают результат именно в формате AVI либо MOV (формат QuickTime). Данный AVI-файл продолжительностью 10 секунд со звуком имеет размер 92 МВ! Это, конечно, без компрессии, поэтому для использования в данном упражнении я взял этот 92-мегабайтный AVI и применил сжатие типа Motion JPEG A, получив файл формата QuickTime MOV, названный `driver_final.mov`.

Это не согласуется с тем, что мы только что говорили о компрессии, однако получившийся файл имеет размер 4,6 МВ, что позволяет вам скачать его с сайта www.friendsofed.com. Если у вас нет модема и/или возможности им воспользоваться, либо вы предпочитаете использовать свое собственное видео, тогда просто замените в этом упражнении `driver_final.mov`любым другим видео файлом.

Нам нужно установить во Flash такую же частоту кадров, как и в нашем видео файле.

1. Создайте новый фильм Flash и, воспользовавшись инспектором параметров (Property) (клавиши Ctrl + F3), установите частоту (frame rate) такую же, как в нашем исходном файле — 30 кадров в секунду (fps). Если вы используете другой исходный файл и не знаете, какова частота кадров в нем, загляните на пару страниц вперед, в раздел «Частота кадров».
2. Если мы создадим новый символ видеоклипа в нашем Flash-фильме и импортируем видео в него, нам будет удобнее манипулировать видео материалом, чем если бы мы импортировали видеоклип прямо на главный монтажный стол. Выполните команду Вставить > Новый символ (Insert > New Symbol) (или нажмите Ctrl + F8). В диалоговом окне Создать новый символ (Create New Symbol) выберите опцию Фильм (Movie Clip) и дайте своему клипу подходящее имя.
3. Для импорта видео выполните **Файл > Импорт (File > Import) (Ctrl + R)**. Выберите видео файл — `driver_final.mov` в нашем случае; в диалоговом окне Импорт видео (Import Video) нужно выбрать опцию Внедрить видео в документ Macromedia Flash (Embed video in Macromedia Flash document):

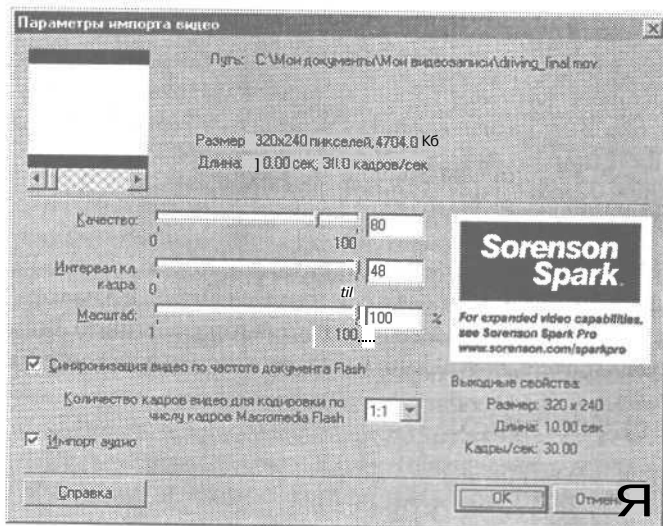


Другая опция здесь — Связать с внешним видеофайлом (Link to external video file). Эта опция доступна только в том случае, если импортируется фильм QuickTime. Ей можно воспользоваться в тех случаях, когда вы хотите представить видеозапись в изначальном виде, например, рекламный видеоролик или речь какого-либо деятеля. В этом случае файл Flash сам по себе не будет содержать видео файла, видео будет загружаться отдельно, в нужный момент. Такой метод очень ограничивает возможности, к примеру, если возникнет надобность экспортировать содержимое Flash в формат Flash 4.

Параметры видео

Сейчас мы приостановим на минутку выполнение нашего упражнения, чтобы рассмотреть некоторые из этих опций. Не волнуйтесь, ничего не отменяется; мы продолжим, как только узнаем о том, как задать наилучшие параметры нашему видео.

Итак, перед нами диалоговое окно Параметры импорта видео (Import Video Settings) — окно параметров Sorenson Spark:



Как видите, здесь указаны Размер (Movie Size) и Длина (Length) фильма, есть также окно просмотра. Кроме того, здесь имеется несколько изменяемых настроек.

Частота кадров

Прежде чем импортировать видео, нужно подумать о частоте кадров, установленной во Flash. Если, к примеру, ваш видео файл снят с частотой 30 кадров в секунду (fps), тогда, чтобы воспроизвести его наилучшим образом, нужно установить во Flash такую же частоту — 30 fps. Частота кадров импортируемого фильма указана рядом с его длиной, справа от окна просмотра.

Причина, по которой необходима настройка частоты во Flash, заключается в том, что звуковая часть видеofilма всегда воспроизводится с частотой кадров изначального видео файла. Импортированное видео будет воспроизводиться с той частотой, которую вы укажете во Flash, однако звук — только с частотой оригинала. Возможная проблема очевидна. Например, вы импортируете видеозапись, на которой крупным планом показан говорящий человек, тогда, если установить частоту кадров, отличную от оригинала, движение губ не будет соответствовать звучащим словам. В принципе, компенсировать различие в частоте кадров при импорте видео возможно, позднее мы поговорим об этом.

Качество

Первый настраиваемый параметр — Качество (Quality). Этот регулятор определяет степень компрессии видеоклипа. Нет, установить Качество на 100% — это совсем не удачная идея, поскольку клип, который получится в результате, может оказаться даже большего размера, чем оригинальный файл! Лучше немного снизить значение Качества, ну, хотя бы до 95%.

Для интернета уровень сжатия чем больше, тем лучше, зато для презентаций на CD-ROM идеальной будет наименьшая компрессия. При чтении с CD-ROM скорость передачи данных куда больше, чем при передаче видео по сети. Важно сознавать, что на старых компьютерах могут возникнуть проблемы даже при просмотре видео с CD-ROM, — из-за низкой скорости обработки данных процессором, поэтому перебарщивать не стоит.

Интервал ключевого кадра

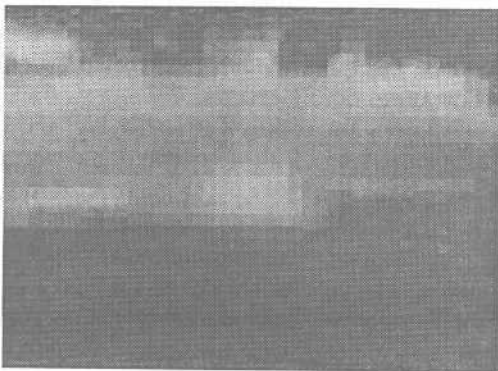
Следующий параметр — Интервал ключевого кадра (Keyframe interval). Этим регулятором настраивается частота ключевых кадров (кадров, со-

держащих полные данные) видеоклипа. Именно этот параметр практически определяет размер и «воспроизводимость» вашего будущего файла! Ключевые кадры очень важны, ведь именно в них содержатся полные кадры видеоклипа. При установке значения 30 ключевым будет каждый тридцатый кадр, а если установить, например, 5, тогда ключевым будет каждый пятый. Самое крайнее значение интервала ключевого кадра — 1. Я бы такое не рекомендовал.

Почему так важны ключевые кадры? Начнем с того, что существует два типа компрессии. Та, что используется в Sorenson Spark, работает так: в каждом кадре видео отыскиваются области, которые от кадра к кадру не перемещаются. Такой вид сжатия называется временным, или темпоральным (temporal), еще его называют межкадровым (interframe). При таком сжатии стараются определить, какие области в каждом кадре остаются неизменными, чтобы не отрисовывать их заново.

Это — одна из причин того, что клипы, в которых много движения, а также снятые нестационарной камерой, занимают много места и требуют широкой полосы пропускания — все движется, мало что остается неизменным от кадра к кадру. Сравните клип, в котором человек что-то говорит, находясь на статичном синем фоне, и съемку какой-нибудь спортивной игры. При одинаковой продолжительности первый клип окажется гораздо меньшего размера!

А вот пример использования чересчур большой межкадровой компрессии. Это кадр из видеоклипа, с которым мы сейчас упражняемся, я применил к нему очень высокую степень сжатия:



В процессе сжатия программа пытается найти в кадре области, соответствующие предыдущим кадрам, но при столь быстро меняющемся изображении не может этого сделать. Такая компрессия ни к чему, так что в этой настройке нужна особая тщательность.

Другой тип сжатия называется пространственным (*spatial*), или, иначе, *внутрикадровым* (*intraframe*). Заданный уровень компрессии применяется ко всем кадрам без исключения, невзирая на соседние кадры. Кодек Sorenson Spark использует в основном межкадровое, или временное сжатие, однако применяет также и пространственное (*внутрикадровое*) — в ключевых кадрах. Видите, какой умелый кодек этот Sorenson Spark! При помощи межкадровой компрессии он старается сжать основную часть видео, и еще при этом использует *внутрикадровую*, чтобы сохранить качество изображения.

Таким образом, сжатие, заданное параметром качества, прямо отражается на том, как будет действовать настройка для ключевых кадров. Здесь нужно найти какое-то равновесие, и это поле для экспериментов с целью получить наилучшее качество при наименьшем размере файла. Увеличение компрессии означает снижение качества изображения, а добавление ключевых кадров помогает поддержать постоянство качества от кадра к кадру.

Чтобы дать вам представление о том, как число ключевых кадров влияет на размер файла, я импортировал свое видео с использованием различных значений этого параметра. Вот два наиболее важных сочетания:

- при импортировании с качеством 90% при интервале ключевого кадра, равном 1, получился файл размером 1,2 МВ;
- импортирование того же самого клипа, но с интервалом ключевого кадра 48, дало в результате файл размером 712 КВ. Но качество этого второго видео ужасно отличается от первого!

Следует помнить, что отнюдь не с каждым клипом будет происходить то же самое. Мой фильм — это сплошное движение, а это значит, что невозможно так же эффективно использовать технологию ключевых кадров, как если бы это была видеозапись выступления диктора новостей, где движения практически никакого нет.

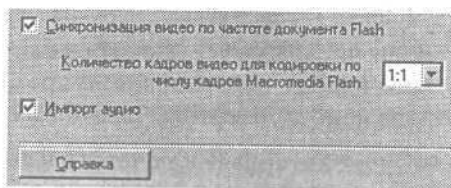
Если вы поднаторели в технике, вам, возможно, интересно будет узнать, что Sorenson Spark экономит место еще и за счет цветового пространства: «Для каждого блока данных 2x2, YUV4:2:0 содержит 4 образца яркости (Y), но только 2 образца цветности (U, V).» Человеческий глаз много чувствительнее к различиям светимости, или яркости, чем к хроматической насыщенности (цвету), и такой метод позволяет на много уменьшить объем информации, необходимой для реконструкции изображения, что дает более высокую степень сжатия. На этом мы наше маленькое отступление закончим, а вы найдете больше информации о YUV 4:2:0 компрессии на www.adamwilt.com/DV-FAQ-tech.html.

Масштаб

Параметр Масштаб (Scale) определяет размеры изображения (ширину и высоту) видеоклипа после его импорта, и с этой настройкой все куда проще, чем с двумя предыдущими! Если уменьшить изображение по сравнению с оригинальным, то снизится размер файла, а следовательно, и воспроизводиться он будет лучше.

Синхронизация

Следующий параметр — Синхронизация видео по частоте документа Flash (Synchronize video to Macromedia Flash document frame rate). Здесь можно компенсировать различие в частоте кадров, таким образом, что в итоге видео будет воспроизводиться с той же частотой кадров, что и фильм Flash.



Например, я захотел для своей мультимедийной презентации установить частоту кадров 60 fps (вообще, это многовато, так что если захотите последовать такому примеру, убедитесь **сначала**, что ваша целевая аудитория сможет воспроизвести презентацию на такой **скорости**). Естественно, такая частота кадров не будет идеальной в том случае, если я захочу импортировать видео с частотой кадров, например, 30 fps. Десятисекундный клип с частотой 30 fps, импортированный в фильм Flash, имеющий частоту 60 fps, превратится в **пятисекундный**. Звук тоже будет не синхронизирован.

Установив Синхронизацию, мы добьемся того, что в процессе импортирования видео разница в частоте кадров будет автоматически устранена. То есть если видеоклип 30 fps импортировать в фильм Flash 15 fps, то при включенной Синхронизации часть кадров будет отброшена, видео станет, пожалуй, немного «дергаться», зато его продолжительность и синхронизация со звуком сохранятся. Sorenson Spark в этой ситуации будет просто отбрасывать каждый второй кадр.

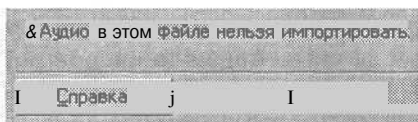
Звук

Наконец, последняя опция касается импортирования звука вместе с видео. Необходимо **сказать**, что не всякий звук можно импортировать: вы

сможете это сделать лишь в том случае, если ваш компьютер поддерживает кодек, который был использован для сжатия звука в оригинальном файле.

Как и с видео, во Flash лучше всего импортировать звук с наименьшим сжатием. Самый лучший выбор, если он вам доступен, это PCM, являющийся, по сути дела, записью звука вообще без сжатия. Однако при желании вы можете импортировать и сжатый звук. Хороший пример — фильмы формата QuickTime, в которых звук сжат при помощи 16-битовой компрессии с прямым порядком байтов при частоте 44,1 кГц — или просто MPEG Layer 3 (MP3).

Учтите, что не всякий сжатый аудио-формат может быть импортирован во Flash. В таком случае вместо опции Импорт аудио (Import audio) вы увидите сообщение об ошибке:



Как уже было сказано, ваша система должна поддерживать аудио кодек, использованный при изначальном кодировании звука. Кроме этого следует принять во внимание, что аудио треки файлов MPG и MPEG могут импортироваться неправильно. Попытка импортировать MPG-видео вместе со звуком может привести к тому, что не удастся импортировать весь фильм! Если у вас возникают проблемы при импортировании видео вместе со звуком, тогда при помощи видео редактора попробуйте перекодировать видео и звук в другой формат, используя кодеки и компрессоры, поддерживаемые вашей системой.

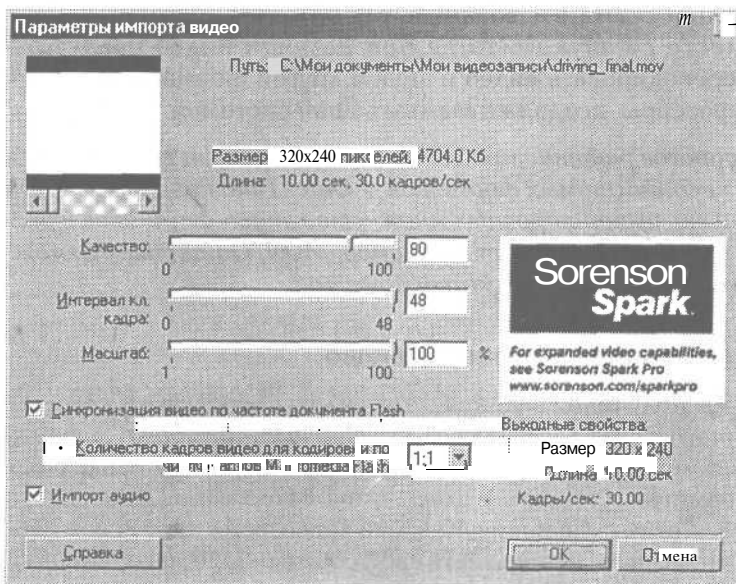
Быстрое и недорогое решение проблемы перекомпрессии ваших MPEG, MPG и других несовместимых форматов — QuickTime Pro от Apple. Он дает возможность импортировать множество видео-форматов и экспортировать их в другой формат, например, MOV. QuickTime Pro можно скачать с www.apple.com/quicktime/download.

Устанавливаем параметры импорта видео

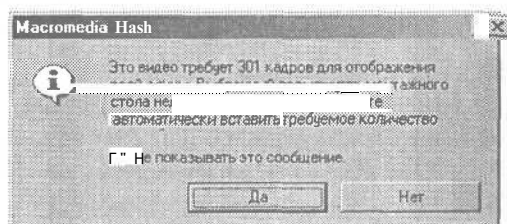
Теперь, когда мы получили сведения о тех силах, посредством которых параметры импорта видео управляют видео-вселенной Flash, давайте вернемся к нашему упражнению и зададим параметры для нашего видео файла. Параметры, указанные здесь, даны, естественно, для нашего файла-образца — driver_final.mov. Каждый видео файл имеет свои особенности, так что если вы используете свой собственный материал, вос-

пользуйтесь новоприобретенными знаниями (а также методом проб и ошибок), чтобы найти наилучшее решение для своего конкретного материала.

1. Для нашего примера установите Качество (Quality) на «90», чтобы попытаться сохранить хорошее качество, подходящее для широкополосного вещания.
2. Установите Интервал ключевого кадра (Keyframe interval) на «48» — наибольшее значение в Sorenson Spark, а параметр Масштаб (Scale) оставьте на 100% - 320x240 точек.
3. В этом упражнении мы установили частоту кадров нашего фильма Flash на 30 fps, что соответствует частоте кадров импортируемого видео, поэтому синхронизация не требуется.
4. Нам нужно выбрать Количество кадров видео для кодировки по числу кадров Macromedia Flash (Number of video frames to encode per number of Macromedia Flash frames). Например, чтобы воспроизводить 1 кадр видео для каждого кадра Flash, мы выбираем из ниспадающего списка 1:1. Если изменить это значение, то часть кадров импортируемого видео будет отброшена, что сделает это видео медленнее, однако может сделать его «дерганым» (менее плавным). Здесь мы выберем воспроизведение всех кадров с частотой кадров Flash, то есть, оставляем 1:1:



5. Нажмите кнопку **ОК** и дождитесь окончания процесса компрессии — Flash сжимает видео уже на этом этапе, вместо того, чтобы делать это при экспортировании готового фильма. Таким образом, процесс экспорта фильма Flash не будет замедлен из-за необходимости сжимать еще и видео.
6. В случае импортирования видео непосредственно в сцену (**stage**), может появиться следующее сообщение, если клип содержит больше кадров, чем монтажный стол, на который вы его помещаете. Нажмите кнопку **Да (Yes)**, чтобы добавить кадры, иначе видеоклип не будет показан вообще — видео воспроизводится только в том случае, если в фильме Flash имеется достаточное количество кадров.



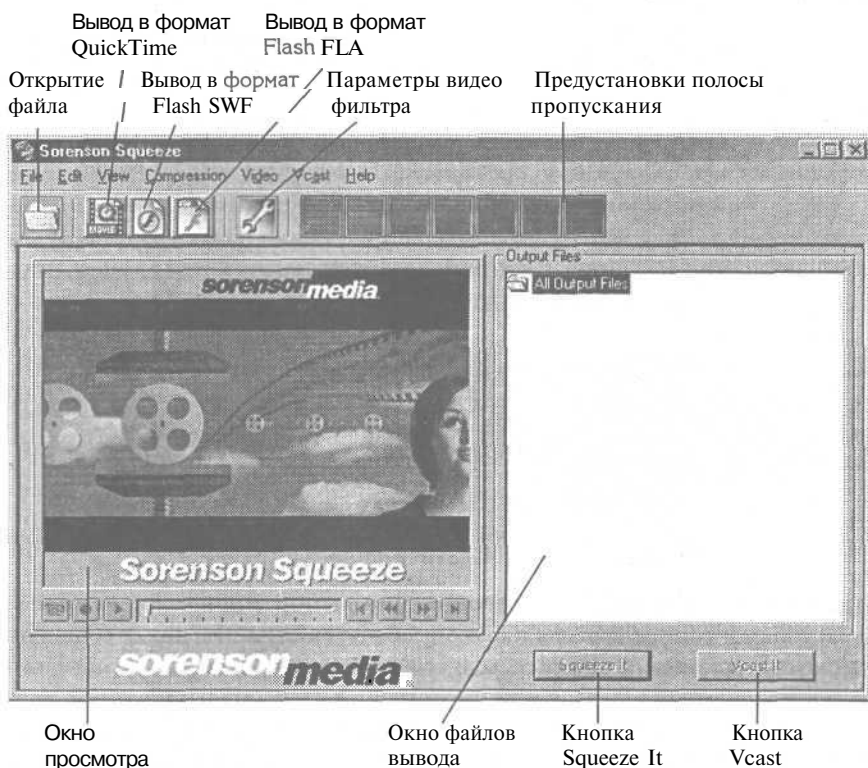
Итак, вы сделали это! Мы благополучно импортировали наш видеоклип во Flash и теперь готовы к его обработке в стиле Flash. Это был не самый захватывающий процесс, однако на нем покоится фундамент для построения удачных Flash-видео презентаций. Потратив некоторое время на подбор этих настроек, вы будете уверены в том, что в вашей презентации не содержится неоправданно больших видео-массивов, которые могли бы затормозить и скомкать все представление.

Sorenson Squeeze

Компрессор, встроенный во Flash MX — Sorenson Spark — хорош, однако существует нечто еще лучшее. Называется оно Sorenson Squeeze. Чтобы получать в результате сжатия действительно наименьший размер файла при сохранении наилучшего качества, вам стоило бы познакомиться с этой программой. Итак, предлагаю взглянуть на этот небольшой, но могучий образчик программного обеспечения; посмотрим, что он сможет сделать с нашим видео. Скачать пробную версию можно с www.sorenson.com (5,5 МВ для PC и 10МВ для Mac).

Sorenson Squeeze — самостоятельная программа компании Sorenson Media, способная кодировать высококачественное видео для Flash MX. Идея состоит в том, чтобы обработать изначальный видеоматериал с помощью Sorenson Squeeze, а потом, уже в сжатом виде, импортировать во Flash.

Вот что мы увидим, открыв Sorenson Squeeze:



Семь кнопок предустановленных режимов скорости передачи (Data Rate Presets) в верхней части интерфейса предназначены для задания минимальной ширины полосы пропускания, при которой можно будет просматривать готовый фильм. Окно просмотра (слева) — само собой, для просмотра импортированного файла. Сжатое видео однако просматривается не здесь. Справа имеется окно выходных файлов (Output Files), здесь можно посмотреть настройки компрессии, примененные к видео файлу. Внизу находятся кнопка Squeeze It, по нажатию на которую начинается процесс компрессии, и Vcast — эта кнопка дает возможность сетевой трансляции вашего видео.

Sorenson Squeeze обладает и такой великолепной способностью, как захват видео прямо с цифровой камеры. Правда, тут есть один недостаток — если вы захотите добавить звук, то придется воспользоваться внешним видео редактором. Конечно, вы можете наложить звук или музыку на видео прямо во Flash. Я в таких ситуациях обычно пользуюсь

программами нелинейного монтажа (non-linear editing — NLE), такими, как Adobe Premiere 6.0 или Sonic Foundry's Vegas Video 3.0.

Для начала давайте сравним два видео файла — первый закодирован встроенным во Flash кодеком Sorenson Spark, а второй — при помощи Sorenson Squeeze. Кодирование было организовано так, чтобы оба фильма имели примерно одинаковое качество изображения и звука.

Первое видео сжато с 4,7 МВ до 716 КВ, что весьма впечатляет:



закодировано
Sorenson Spark

Второе видео, однако, сжато с 4,7 МВ до 430 КВ!



закодировано
Sorenson Squeeze

Рассмотрим поближе два одинаковых кадра. То, что имеет место сжатие, заметно на обоих, хотя второе видео выглядит более «ровным». Интересная деталь: если эти кадры (из каждого фильма) сохранить как изображения JPEG, то для второй картинки размер файла получится больше, чем для первой. Но **подождите!** Ведь второе изображение получено из видеоклипа меньшего размера? Как же так?

Все просто: компрессор Sorenson Squeeze сумел произвести видеоклип меньшего размера, чем Sorenson Spark, но при этом с лучшим качеством изображения! Squeeze исключительно полезен, если вы собираетесь часто работать с видео и Flash MX, и хотите иметь по возможности наименьший размер файлов.

Сжатие видео

Здесь важно отметить, что раз уж мы применяем Sorenson Squeeze с целью получения наилучшего качества при наименьших размерах файла, то лучше всего использовать «сырое» (несжатое) видео. Если же мы импортируем видео, которое уже подверглось компрессии, и станем сжимать его еще, то качество изображения может пострадать. Вообразите такую аналогию: возьмем фотографию и скомкаем ее руками в шарик, затем развернем, расправим, опять скомкаем и так далее. С каждым разом изображение будет все хуже. То же самое и с видео. Если взять сырое видео и применить к нему компрессию, а затем импортировать то, что получилось, в другую программу и применить другой тип сжатия, и так далее, — потеря качества неизбежна.

Параметры сжатия

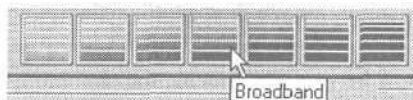
Давайте импортируем наше видео в Sorenson Squeeze.

1. Нажмите кнопку File Open (открыть файл) и выберите сырой видео файл (в нашем примере это `driving_final.mov`). Теперь вы можете увидеть этот клип в окне просмотра.
2. Нам нужно выбрать формат, в который мы будем кодировать. Здесь три варианта: QuickTime Video, Flash SWF и Flash FLV. Так как мы будем импортировать сжатое видео во Flash MX, выбираем формат Flash FLV, нажав соответствующую кнопку:



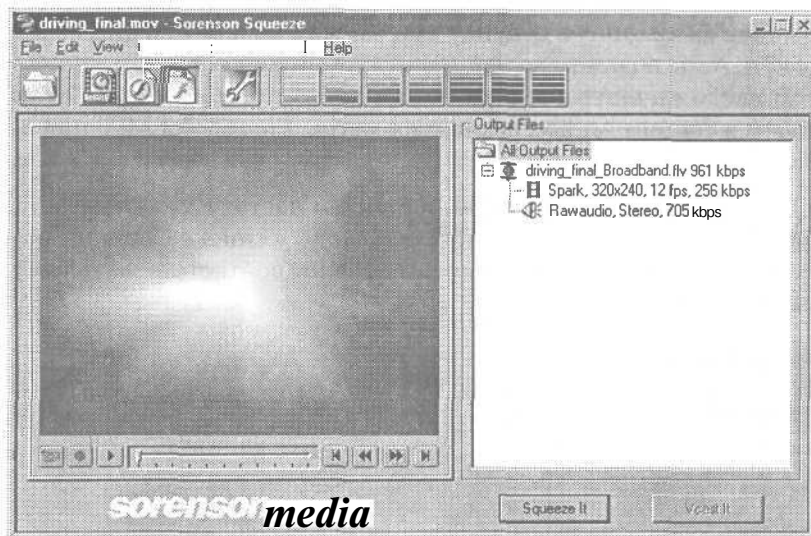
Flash FLV — новый формат файлов, присущий только Flash MX. Хотя существуют и другие форматы, которые можно использовать во Flash MX, этот формат наилучший для импортирования сжатого видео.

3. Теперь надо выбрать скорости передачи (ширину полосы пропускания), для которой мы будем кодировать видео. Вы, наверное, заметили, что кнопки выбора скорости передачи подсветились только после того, как мы выбрали формат видео. Теперь семь кнопок режимов кодирования активны. Давайте выберем четвертый режим — Broadband (широкополосная сеть):



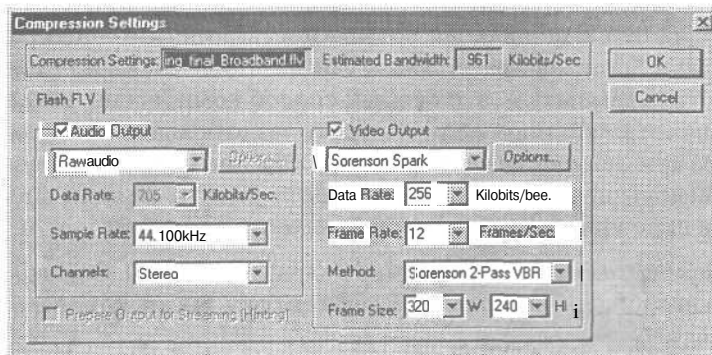
Такая настройка не совсем соответствует тому, что нам нужно, но не волнуйтесь — мы сможем быстро довести ее до совершенства. Обра-

тите внимание, что выбранные нами параметры отобразились в окне выходных файлов (Output Files):

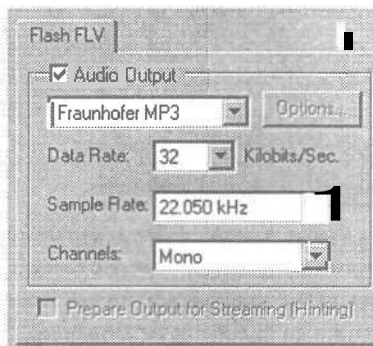


Если, ради интереса, экспортировать с этими предустановленными параметрами, в результате получится файл видео со звуком размером свыше 2 МВ! Не годится. Можно было бы выбрать другую предустановку, но мы предпочтем эту, так как она ближе к тем параметрам, которые мы хотим получить в итоге. Поэтому давайте-ка копнем поглубже!

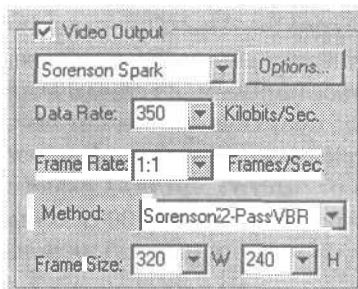
4. Обратимся к окну Output Files и выделим строку, начинающуюся со слова Spark. Теперь идем в меню Edit > Output Compression Settings..., открывая диалоговое окно Compression Settings (параметры компрессии):



5. Для нас не обязателен превосходный звук, достаточно просто хорошего; мы ведь все-таки собираемся передавать это видео по сети. Качество звука стандарта CD требует, как правило, скорости передачи не менее 48 килобит в секунду при частоте дискретизации не менее 44 кГц. А мы возьмем на одну-две ступени ниже, так что звук будет хорошего качества, и в то же не займет много места. Итак, выбираем в секции Audio Output параметры Fraunhofer MP3 при 32 килобит/с.
6. Здесь же, в секции Audio Output, изменяем частоту дискретизации (Sample Rate) на 22,050 кГц и выбираем Mono в списке Channels (каналы). Теперь ваши аудио параметры должны соответствовать тому, что вы видите на снимке:

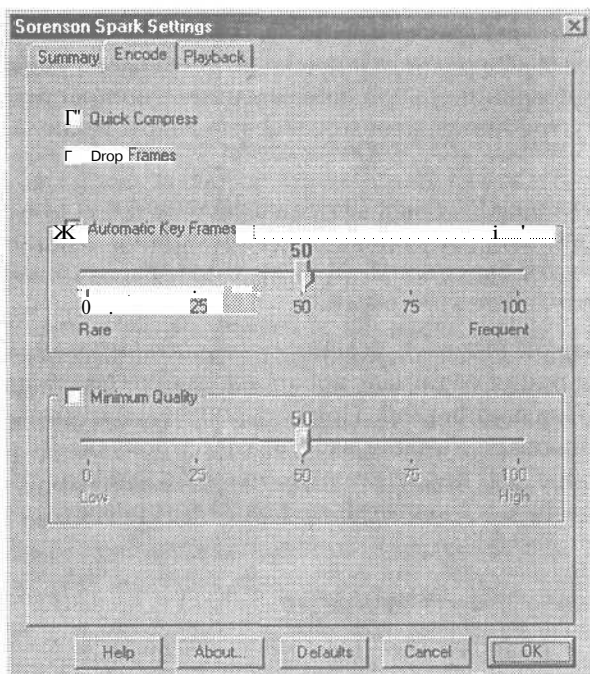


7. Теперь давайте переместимся в секцию Video Output. Установим Data Rate (скорость передачи) на 350 килобит/с, а Frame Rate (частоту кадров) — на 1:1 (вы можете также ввести в это поле «30»). Это означает, что при сжатии будет использоваться та же частота кадров, что и в оригинальном видеоклипе, то есть, в нашем случае, 30 fps.
8. Выберите из списка Method (метод) Sorenson 2-Pass VBR. Это, пожалуй, важная функциональная возможность Sorenson Squeeze, отсутствующая в Sorenson Spark. 2-Pass VBR (Variable Bit Rate — переменная частота данных) — это особый способ компрессии, когда сцены с активным движением кодируются гораздо большим количеством данных, чем статичные сцены. Таким образом, снижается размер файла при повышении производительности. Кстати, сегодня едва ли не каждый видеofilm DVD кодируется таким способом.
9. Наконец, установите Frame Size (размер кадра) на 320x240. Окончательный вид секции параметров видео должен быть таким, как здесь:

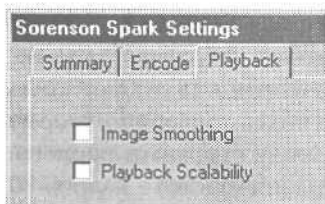


Обратите внимание на верхнюю часть диалогового окна Compression Settings — значение расчетной ширины полосы пропускания (Estimated Bandwidth) теперь составляет 382 килобит/с. Можно было бы удовольствоваться этими настройками, однако я хочу большего — полагаю, и вы тоже. Копаем дальше.

10. Нажмите кнопку Options... в секции Video Output — откроется диалоговое окно Sorenson Spark Settings (параметры Sorenson Spark). Здесь три вкладки, первая из них — просто общая сводка (Summary) установленных параметров. Перейдите на вкладку Encode (кодирование):



11. **Первый параметр** — Quick Compress (быстрое сжатие). При включении этой опции сжатие видео происходит примерно на 20% быстрее, но при этом приносится в жертву качество. Впрочем, на самом деле эта потеря качества минимальна и на глаз практически не заметна. Наш видеоклип длится всего 10 секунд, так что его сжатие займет от 30 секунд до 2 — 3 минут, в зависимости от быстродействия вашего компьютера. Много времени мы тут не выиграем, лишние 30 секунд у нас как-нибудь найдутся, так что оставляем эту опцию отключенной.
12. **Следующий флажок** — Drop Frames (пропускать кадры). Эта опция действительно важна только в том случае, если вы выбираете последнюю настройку на этой вкладке — Minimum Quality (минимальное качество). Регулятор Minimum Quality используется для настройки минимального качества изображения кодированного видео. При этом, естественно, многим придется пожертвовать: снизится частота кадров, а значит, и качество изображения в каждом кадре. В этом упражнении мы оставляем эту опцию отключенной.
13. **Главный орган управления** на этой вкладке — регулятор Automatic Key Frames (автоматические ключевые кадры). Он позволяет установить вероятность того, что при переходе от кадра к кадру будет вставлен ключевой кадр. Установите регулятор на 75. При этом автоматическое создание ключевых кадров будет происходить несколько чаще, чем при значении по умолчанию (50), в нашем клипе — непрерывное движение, поэтому требуется, чтобы ключевые кадры располагались чаще.
14. **Теперь перейдите** на вкладку Playback (Воспроизведение) и установите флажок Image Smoothing (Сглаживание изображения). Это помогает устранить или хотя бы снизить эффект «блоков», который может возникнуть в результате компрессии.
15. **Установите также флажок** Playback Scalability (масштабирование воспроизведения). Эта опция облегчает воспроизведение видео на различных компьютерных платформах. При воспроизведении фильма на медленном компьютере каждый второй кадр пропускается, компенсируя, таким образом, недостаточную производительность компьютера (например, фильм с частотой кадров 30 fps превратится в 15 fps и так далее).



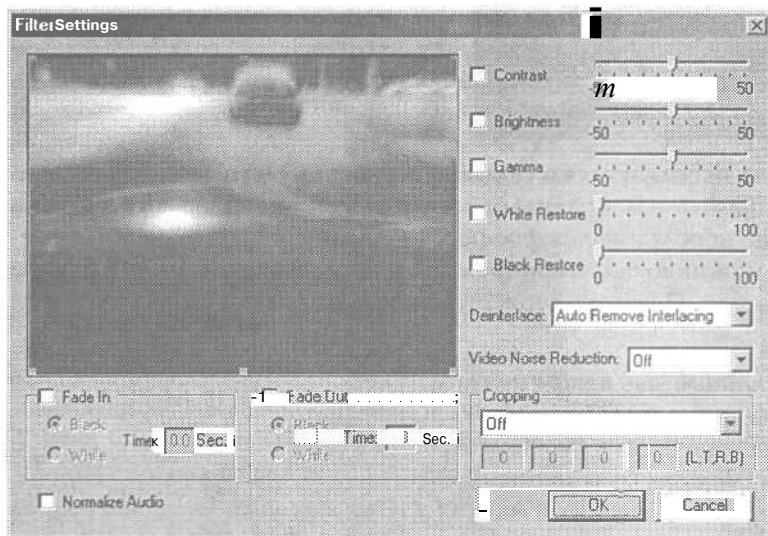
16. Нажмите ОК, затем еще раз ОК, подтверждая наши новые параметры компрессии.

Параметры видео фильтра

В Sorenson Squeeze мы можем также подредактировать наше видео — для этого на инструментальной панели основного интерфейса имеется кнопка Video Filter Settings (параметры видео фильтра).



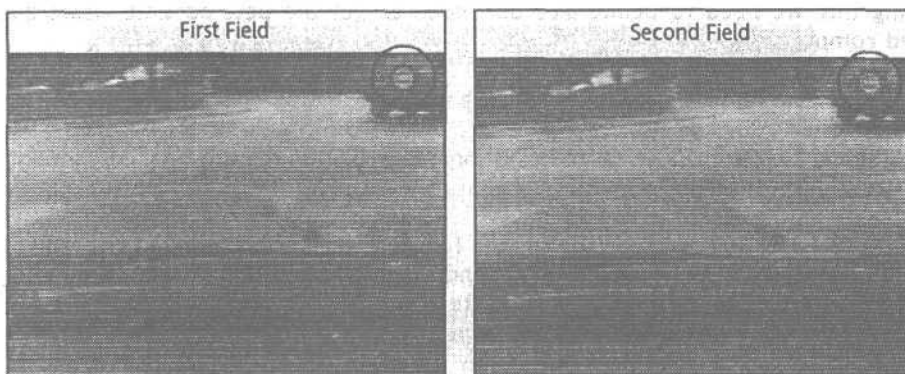
Давайте нажмем эту кнопку и посмотрим, что здесь имеется.



Слева — окно просмотра. Справа сверху органы управления Contrast (контраст), Brightness (яркость), Gamma (гамма-фактор — коррекция градаций яркости), White Restore (баланс белого) Black Restore (баланс черного). Можете подвигать эти регуляторы, все изменения будут немедленно отражаться в окне просмотра.

Дальше идет опция Deinterlace (Устранение чересстрочной развертки) — еще одна полезная функция Sorenson Squeeze. Вкратце объясню, что это такое: при чересстрочной (interlaced) развертке кадр делится на два поля. Сначала на экран выводится первое поле, построчно сверху вниз, а затем второе поле — в промежутки между строками первого. Вывод каждого из полей занимает $1/60$ секунды при частоте кадров 30 fps (если точно — 29,97), невооруженным глазом это не заметно.

Чересстрочная развертка используется в большинстве систем телевидения. Ниже на снимках вы можете увидеть отличия между двумя полями, внимательно рассмотрев запасное колесо «Лендровера» (обведено кружком):



Этот метод используется в телевидении потому, что в старых телевизорах были проблемы с правильным отображением видео: в те времена для этого еще использовался фосфор, телевизор еще не успевал вывести на экран нижнюю часть изображения, когда в верхней части экрана оно уже угасало. Поэтому чересстрочная развертка сегодня является стандартом.

Попутно заметим, что в мире на сегодня наиболее распространены два видео стандарта: NTSC и PAL. NTSC (National Television Standards Committee) используется главным образом в Северной Америке, Японии и на Тайване. Формат PAL (Phase Alternate Lines) применяется в Европе, Австралии, Новой Зеландии и многих других странах. Частота кадров в формате NTSC составляет 29,97 или 30 fps, в PAL — 25 fps.

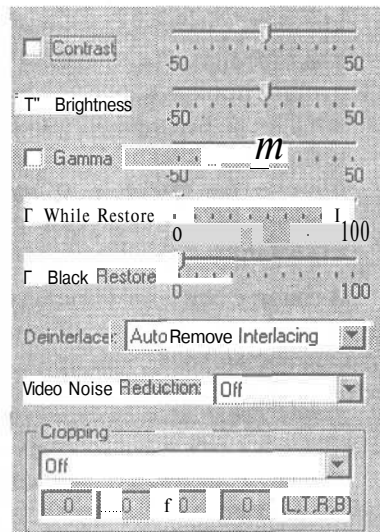
В компьютерных мониторах такой метод не применяется. Здесь используется построчная (non-interlaced), или, иначе, прогрессивная (progressive) развертка. Изображение построчно выводится на экран сверху вниз сплошным полем. Многие современные цифровые видеокамеры позволяют вести съемку в режиме построчной (сплошной) раз-

вертки 30 fps, — это стандарт для мультимедийного видео. Однако в большинстве своем видеозаписи имеют чересстрочную развертку и частоту кадров 29,97 либо 25 fps.

Параметры сжатия

Теперь вы понимаете, что нам нужно устранить чересстрочную развертку, чтобы получить наилучшее качество изображения после применения компрессии.

1. Из списка Deinterlace выберите Auto Remove Interlacing (Автоматически устранить чередование).
2. Следующий параметр — Video Noise Reduction (Видео шумоподавление). Позволяет устранить так называемый «песок», «зернистость» изображения, в общем, «посторонние шумы». В данном случае нам это не нужно.
3. То же касается функций Cropping (обрезка), а также эффектов фединга — постепенного «проявления» (Fade In) и «затухания» (Fade Out). Cropping позволяет именно обрезать изображение, то есть не просто изменить ширину и высоту — если вы, например, выберете соотношение сторон 16:9, то излишек изображения по ширине или высоте будет отброшен. Fade In и Fade Out позволяют добавить эффекты фединга, соответственно, в начале и в конце клипа, с настройкой продолжительности эффекта.
4. Наконец, есть здесь опция Normalize Audio. Эта функция анализирует клип в поисках самого громкого звука, а затем соответственно подстраивает весь звуковой трек так, чтобы наиболее громкая часть не имела искажений. Это мы тоже не трогаем. Теперь проверьте, соответствуют ли ваши настройки моим (единственная опция, которую вы должны были изменить — это Deinterlacing), и нажмите кнопку ОК.
5. Теперь мы готовы экспортировать наше видео в формат FLV. Нажмите на кнопку Squeeze It в правом нижнем углу главного окна. Появится информационное окно Squeezing



(чтобы наблюдать сложный процесс обработки вашего видео, нажмите кнопку Preview On — включить просмотр):



Процесс с использованием Sorenson 2-pass VBR происходит в два прохода (этапа). При первом проходе результат довольно невзрачный, второй проход занимает побольше времени, и результат гораздо лучше. После его завершения мы получаем видеоклип FLV, готовый для импорта во Flash MX. Формат FLV сам по себе не предназначен для просмотра, его можно просмотреть только внутри Flash MX. Этот формат был специально разработан для вызова из Flash в качестве отдельного видеоролика или для импорта в среду разработки Flash.

В связи с этим нам предстоит вернуться к вопросу о частоте кадров. Одно из главных отличий импортирования FLV во Flash состоит в том, что вы не сможете, как прежде, в Spark, синхронизировать FLV по частоте кадров с основным фильмом Flash. Поскольку теперь вы об этом знаете, правильным будет заранее установить во Flash частоту кадров, соответствующую вашему FLV.

Да, конечно, если установить во Flash более высокую частоту кадров, например, 60 fps, то FLV на 30 fps будет воспроизводиться без проблем,

тем не менее, делать так совершенно не обязательно. В противоположном случае, если вы считаете необходимым понизить частоту кадров видеоклипа с целью уменьшения размера файла, это следует делать в Sorenson Squeeze, до импорта клипа во Flash.

Еще одно важное замечание: если вы снизите частоту кадров в Squeeze, то размер файла может и не уменьшиться. Причина в том, что Squeeze, заметив, что частота кадров снижена, использует освободившийся объем данных для улучшения качества изображения. Поэтому, если вы хотите уменьшить размер файла, эффект даст одновременное понижение частоты кадров, скорости передачи и частоты ключевых кадров в настройках Squeeze.

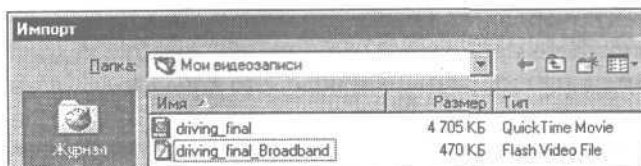
Перенос видео во Flash

Итак, мы все обсудили, а теперь давайте подведем итог нашего упражнения — импортируем FLV файл во Flash.

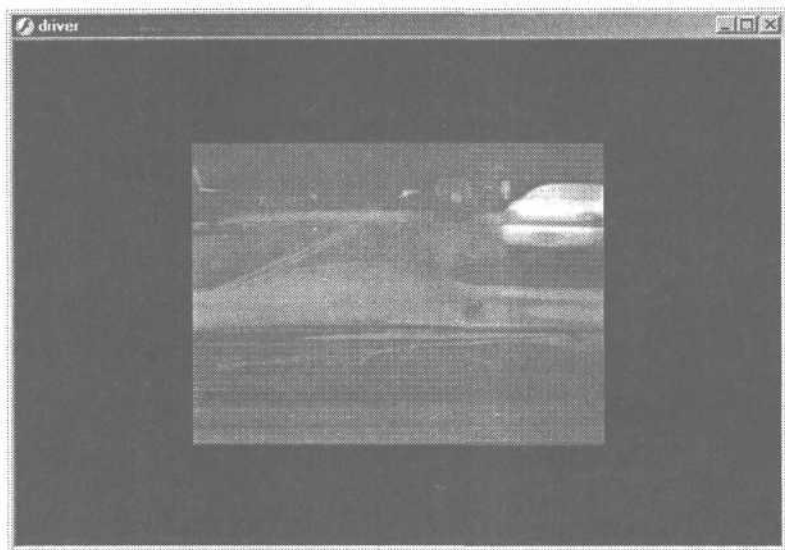
1. Откройте фильм Flash, в который вы импортировали видеоклип, выполняя упражнение в начале этой главы. Откройте Библиотеку (Library) (F11) и двойным щелчком на значке фильма (Movie Clip) откройте его для редактирования.



2. Выберите кадр 1 и удалите имеющееся видео. Не забыли — это исходный driver_final.mov, видео файл QuickTime, который мы импортировали в начале главы.
3. Теперь можно импортировать файл FLV, только что нами созданный. Выполняем команду **Файл > Импорт (File > Import)**, и видим рядом с исходным видеоклипом файл FLV (кстати, обратите внимание на огромную разницу между размерами файла-источника MOV и сжатого FLV):



4. Выберите `driving_final_Broadband` для импорта его во Flash.
5. На панели, расположенной чуть ниже монтажного стола, щелкните на ссылке Сцена 1 (Scene 1), чтобы вернуться к основному монтажному столу. Выделите кадр 1 в текущем слое (layer) и перетащите копию фильма из Библиотеки (Library) в сцену.
6. Вот и все! Командой Управление > Проверить фильм (Control > Test Movie) запустите просмотр вашего оптимизированного видео во всей его красе:



7. Сохраните свой файл Flash и закройте его.

Заключение

Итак, мы это сделали! Мы оптимизировали наше Flash MX видео, теперь оно готово для широкополосного показа в интернете. Конечно, для этого нам пришлось несколько снизить качество изображения и звука, а также уменьшить размер изображения.

Следует напомнить о некоторых моментах, имеющих отношение к импорту видео во Flash MX. Возможно, будет немало проб и ошибок, прежде чем ваше видео получится таким, как нужно, поэтому заранее будьте готовы экспериментировать с различными конфигурациями параметров. Хорошо бы вести протокол ваших находок, чтобы потом сравнивать различные настройки, влияющие на размер изображения, воспроизведение, возможность масштабирования и так далее.

При этом следует учитывать, где и как будет в конечном итоге просматриваться ваше видео, представлять себе, каким «железом» (аппаратными средствами) и полосой пропускания может располагаться ваша целевая аудитория. Как правило, при сжатии и импортировании видео во Flash MX приходится искать компромиссы: качество изображения и звука против маленьких, компактных, удобных для загрузки видео файлов.

Действительно важно при разрешении этих вопросов стараться ограничить размер файлов, и об этом много говорилось в данной главе. Очень многие пренебрегали этим моментом в нетерпеливом стремлении поскорее использовать видео во Flash, а потом расплачивались, обнаружив, что их файлы попросту чересчур велики. Что ж, теперь вы можете перейти к следующей главе, и там мы повеселимся по-настоящему...

Обработка видео во Flash

Краткое содержание главы:

- средства потоковой передачи Flash — видео;
- управление видеофильмами при помощи простейших программ ActionScript;
- маскирование видео.

Две предыдущие главы были посвящены тому, что необходимо сделать, чтобы получить возможность использования видео во Flash. В этой главе мы сделаем шаг вперед и узнаем, как обрабатывать видео материал и управлять им во Flash. Другими словами, настал час пожинать плоды наших трудов.

Начнем с рассмотрения некоторых способов, позволяющих удостовериться, насколько эффективно наш фильм будет передаваться пользователям, а затем поговорим о том, каким образом Flash управляет видео-клипами и монтажными столами. Вооружившись этими знаниями, мы, при помощи нескольких простых кнопок и ActionScript создадим несколько видеоэффектов.

Потоковая обработка видео во Flash

Во Flash потоковая передача информации несколько отличается от традиционной — серверной. Поток серверы аккуратно выдают пакеты информации, когда они требуются проигрывателю. При нормальном ходе этого процесса пользователь должен подождать, пока происходит буферизация, а затем начинается воспроизведение всего фильма целиком. Поток серверы достаточно «умны», чтобы рассчитать полосу

пропускания пользователя и предоставить тот или иной вариант материала, подходящего размера.

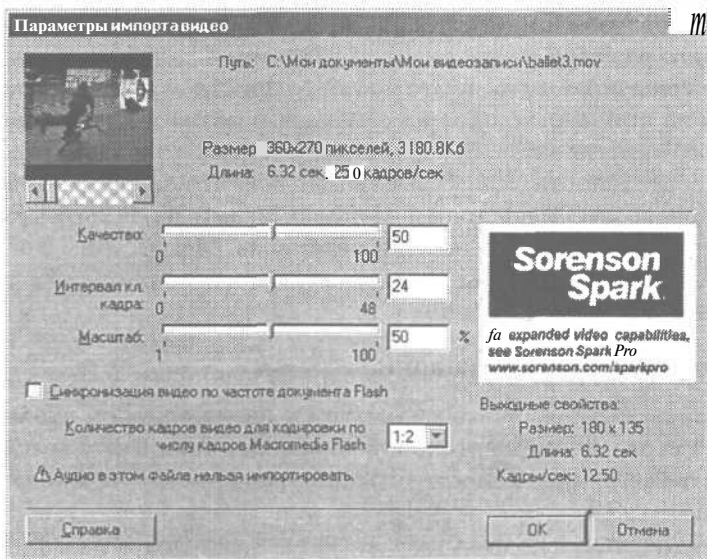
Во Flash такое тоже возможно, но это не так-то просто — видео поток во Flash основан на принципах, схожих с реальной потоковой передачей, но без всякого «интеллекта». Flash попросту загружает все подряд, согласно последовательности, и воспроизводит все, что загружено. В результате мы нередко имеем воспроизведение рывками, из-за того, что Flash ожидает загрузки очередной порции материала. Мы обсудим, как можно этого избежать, но сначала давайте познакомимся поближе с проблемой «рваного» воспроизведения.

Потоковая передача без буферизации

Потоковая передача видео во Flash реализуется на монтажном столе `_root` (главном). Для реализации потоковой передачи в любой форме вы должны поместить ваше видео непосредственно на главном монтажном столе, а не в клипе.

В данном примере нашей целью будет потоковая передача со скоростью 56k.

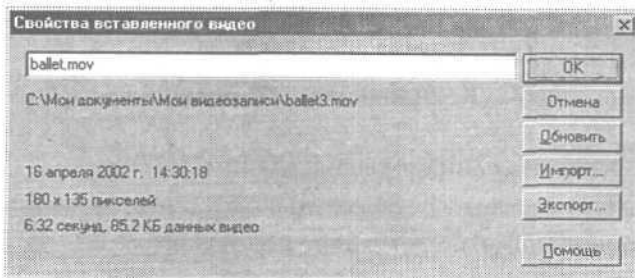
1. Найдите короткий видеоролик (продолжительность моего — шесть секунд) и выясните частоту кадров. Если хотите использовать тот же самый файл, что и я, скачайте `streaming.flac` сайта friends of ED.
2. Откройте новый фильм Flash и установите такую же частоту кадров, как в предназначенном для импорта видеоролике.
3. Выполните команду **Файл > Импорт в библиотеку (File > Import to Library)**.
4. В появившемся диалоговом окне выберите опцию **Вставить видео в документ Macromedia Flash (Embed video in Macromedia Flash document)**.
5. Установите параметры импорта в зависимости от детализации и подвижности изображения в вашем ролике. Я установил параметры, которые вы видите на снимке, исходя из следующих соображений:
 - **качество (Quality):** здесь нет особой детализации;
 - **интервал ключевого кадра (Keyframeinterval):** подвижность умеренная;
 - **масштаб (Scale):** оригинал слишком велик для модема 56k;
 - **соотношение кадров:** мне совсем не обязательно оставлять все кадры;
 - **импорт аудио (Importaudio):** в моем клипе нет звука.



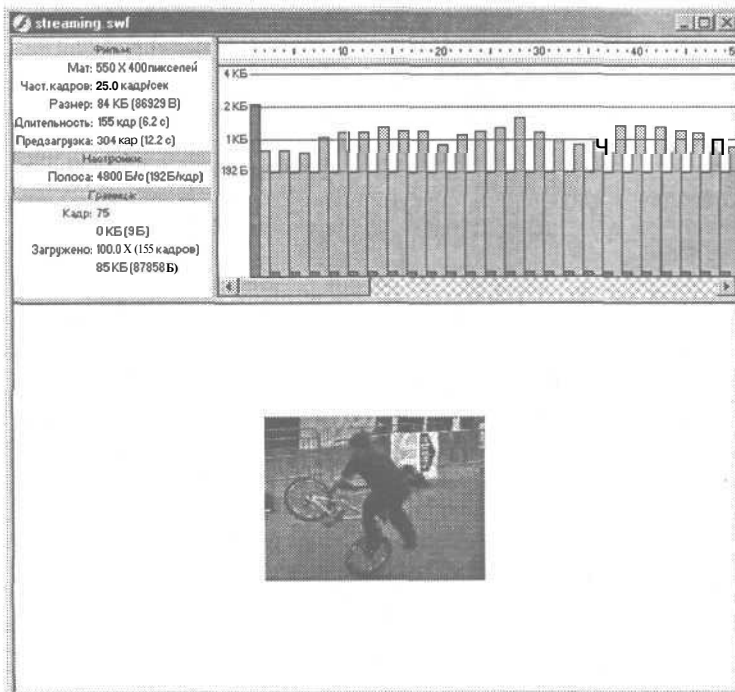
В конечном итоге у меня получился фильм размером 85 килобайт — из 22,2 мегабайтного несжатого DV-PAL фильма QuickTime. Такой результат достигнут, главным образом, благодаря пропуску каждого второго кадра, компрессии, а также уменьшению размера изображения.

Если вы использовали другой клип, постарайтесь добиться примерно такого же размера файла — около 100 KB. Не импортируйте звук вместе с клипом — он нам сейчас не нужен, а размер файла увеличил бы очень заметно. К тому же, может проявиться раздражающий эффект «тикания». Разговор об аудио нам еще предстоит — в главах 6 и 7, так что, если интересуетесь этим вопросом, наберитесь терпения.

Напоминаю, что вы можете пере-импортировать клип или просто посмотреть его параметры, дважды щелкнув на клипе в Библиотеке (Library):

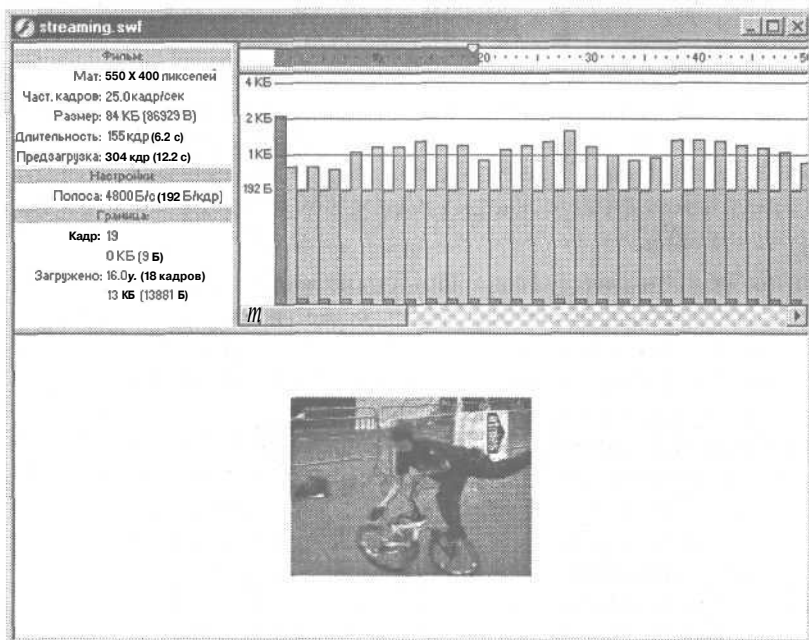


6. Получив клип подходящего размера, перетащите его копию из Библиотеки (Library) в сцену (stage). Разрешите Flash растянуть монтажный стол до длительности клипа.
7. Протестируйте фильм с помощью команды Управление > Проверить фильм (Control > Test Movie) или **Ctrl+Enter**. Видео должно проигрываться быстро и гладко, без задержек — Flash загружает данные достаточно быстро, пока не идет речь о симулировании потоковой передачи.
8. Flash имеет несколько предустановленных значений ширины полосы пропускания, кроме того, вы можете создать новый набор параметров сами, с нуля. Находясь в режиме проверки фильма, выберите в меню Отладка (Debug) пункт 56К (4,7 КБ/с). Мы сейчас симулируем загрузку вашего фильма через модем 56К со средней скоростью (4,7 КБ в секунду).
9. Теперь выберите команду Вид > Профили полосы (View > Bandwidth Profiler). Откроется окно, а в нем — нечто, напоминающее очертания города на горизонте. На самом деле, здесь отображен размер отдельных кадров и другая полезная информация о вашем фильме:



Прежде, чем рассматривать профили полосы в деталях, давайте сделаем то, что собирались — запустим наш поток.

10. Выполните Вид > Показать поток (View > Show Streaming) — этой командой будет запущена симуляция загрузки нашего Flash-фильма модемом 56К. Как только вы это сделаете, вы увидите на диаграмме кадров растущую зеленую полосу:



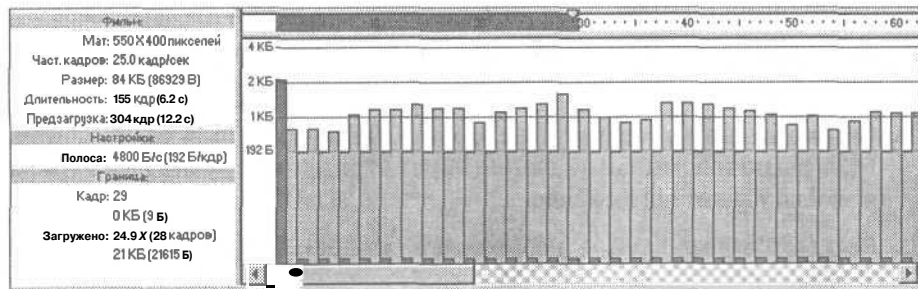
Эта полоса показывает, сколько материала уже загружено. Маленькая стрелочка над кадрами — индикатор воспроизведения (представьте магнитофонную головку).

Посмотрите, как воспроизводится фильм — он стал прерывистым, «рваным». Содержание воспроизводится по мере загрузки, и Flash, воспроизведя очередную порцию, останавливается. В результате мы получаем такой вот фильм — старт-стоп-старт-стоп, совсем не этого мы хотели!

Сохраните свой фильм под именем `streaming fla`. Нам надо будет улучшить процесс загрузки и воспроизведения при помощи буферизации.

Профиль полосы

Он может показаться китайской головоломкой, однако в действительности профиль полосы невероятно полезен в деле рационализации загрузки и воспроизведения.



Так что же показывает диаграмма справа? Ничего сложного: здесь наш файл поделен на отдельные кадры, их размер и отражается на диаграмме; так мы можем понять, где сосредоточена основная «масса» нашего фильма. В моем примере она распределена достаточно ровно — средний размер кадра чуть меньше 1 КБ.

Прямоугольники на диаграмме профиля полосы могут рассказать еще кое-что о моем фильме. Если помните, при импорте я установил соотношение кадров 1:2. Это означает, что Flash после каждого кадра помещает его дубликат, чтобы заполнить промежуток. Это и отражают пики и пробелы; теперь, глядя на них, мы обнаруживаем ужасную красную линию (на снимке она отходит от надписи «192 Б»).

Прямоугольник, расположенный выше красной линии, означает, что для кадра необходимо загрузить данные. Если ниже — для этого кадра ничего загружать не надо. С моим соотношением кадров 1:2 получается, что для каждого второго кадра ничего загружать не надо, поскольку это просто повторение предыдущего.

Казалось бы, ну подумаешь, всего 1 КБ на каждый кадр. Но ведь в моем фильме 25 кадров в секунду. Простой подсчет показывает, что если на каждый кадр приходится 1 КБ, то загрузка должна происходить со скоростью 12,5 КБ в секунду. Сравните это с 4,7 КБ в секунду, которые способен обеспечить модем 56К, и сразу станет ясно, что модем не справится, и мы встретимся с нашими старыми друзьями, которых зовут мистер Стоп, мистер Старт и мистер Дёрг.

Вот тут-то на сцену выходит буферизация, и число рядом с надписью Предзагрузка (Preload) заставляет обратить на себя внимание. Учитывая частоту кадров и общий размер нашего файла, выдается предпола-

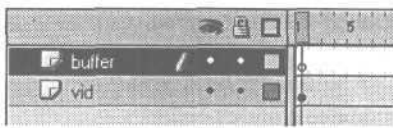
гаемое количество кадров (и время) для загрузки в буфер, чтобы можно было начать воспроизведение фильма, загружая тем временем остальное. Значение это не всегда точное, но в первом приближении вполне подойдет.

Запишите указанное количество кадров, и мы продолжим.

Буферизация потока

Сейчас мы продемонстрируем, насколько просто во Flash MX сгладить видеопоток.

1. Откройте файл `streaming fla`, созданный в предыдущем упражнении. Если вы не запомнили значение параметра предзагрузки из профиля полосы, узнайте его сейчас.
2. Вставьте новый слой — назовем его «buffer». Первый слой назовем «vid».



Первое, что нужно сделать — переместить видео материал на монтажном столе так, чтобы он начинался с кадра предзагрузки (указанного в профиле *полосы*).

3. Двойным щелчком выделите видеоролик на монтажном столе. Теперь перетащите его к кадру предзагрузки (в моем случае это кадр 304) и отпустите.

Теперь наш фильм будет начинаться чуть позже — мы дадим Flash Player время на загрузку.

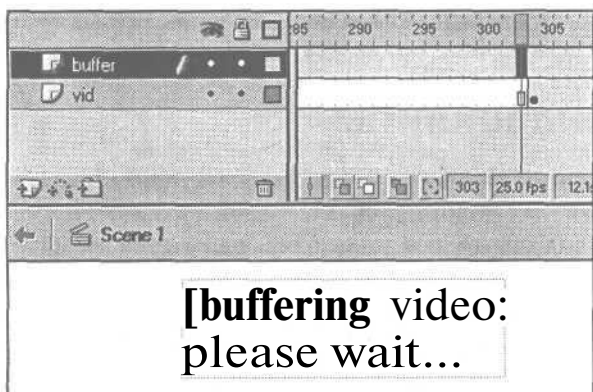
4. Щелкните на кадре 1 слоя «buffer» и, воспользовавшись инструментом Текст (Text), изобразите какую-нибудь надпись, информирующую о предзагрузке, буферизации, ну, например...

Яри использовании Flash-видео в Сети хорошо бы занять пользователя чем-нибудь на время буферизации вашего фильма. Это может быть некая анимация (но не настолько сложная, что для нее тоже понадобится предзагрузчик — предзагрузчик!предзагрузки!),какой-то текст, или игрушка, если вы хорошо знакомы с ActionScript.



Теперь, если прокрутить монтажный стол на всю длину (или протестировать фильм), мы заметим, что текст предзагрузки остается на экране слишком долго, и буферный слой на монтажном столе такой же длины, как весь фильм. Нам нужно обрезать его, чтобы он исчез до того, как начнется наш фильм.

5. Найдите на монтажном столе начало фильма (у меня это по-прежнему кадр 304). В слое «buffer» выделите кадр, предшествующий началу фильма и вставьте пустой ключевой кадр при помощи Вставка > Пустой ключевой кадр (Insert > Blank keyframe) или F7. Теперь, если вы протащите курсор воспроизведения («магнитофонная головка» — красный прямоугольник вверху) по этому месту, то увидите четкий переход.



6. Протестируйте фильм с помощью Управление > Проверить фильм (Control > Test movie) и откройте окно профиля полосы, если оно не открыто.

7. Как видите, весь наш материал уже загружен — проследите за симуляцией потока, Вид > Показать поток (View > Show Streaming).

Теперь проследите за увеличивающейся зеленой полосой при приближении к концу монтажного стола. Если наша предзагрузка настроена точно, то к тому моменту, когда зеленая полоса прекращает расти, достигнув конца монтажного стола, индикатор воспроизведения должен уже почти догнать ее. Если же он достигает конца много позже, значит, наш буфер чересчур велик. Ну, а если индикатор успевает догнать зеленую полосу и фильм начинает дергаться, тогда буфер слишком мал.

Значение параметра Предзагрузка (Preload), указанное в Профиле полосы, порой не очень-то соответствует действительности. Лучшее, что тут можно сделать, это проследить за профилем полосы в процессе симуляции потока и попытаться вычислить, сколько нужно дополнительного времени. Если, скажем, нужна лишняя секунда буферизации, тогда добавляем секундное число кадров и сдвигаем видео.

Я хочу подчеркнуть, что нельзя заставлять пользователя ждать больше, чем это действительно необходимо. Не забудьте, что данный пример мы оптимизировали для пользователей с модемом 56к, а для клиентов с более быстрым соединением такая буферизация будет излишней. Чтобы избежать такого несоответствия, вы можете создать несколько разных SWF-файлов для различных пользователей, понижая или повышая качество, размер и буферизацию для той или иной ширины полосы.

А теперь речь пойдет об управлении видеоклипами и некоторых трюках монтажного стола.

Автономные видеоклипы

В предыдущем упражнении мы размещали наш видео материал на главном монтажном столе. Как уже говорилось, это удобно для потоковой передачи, однако если бы мы захотели использовать один и тот же материал — и это относится не только к видео — несколько раз, то наш файл мог бы раздуться до неприличных размеров.

Чтобы избежать этого, нам нужно преобразовать наш видеоматериал в клип (фильм). Сделать это несложно, но тут тоже есть подводный камень — после такого преобразования мы лишимся возможности потоковой передачи видео. Любой клип во Flash должен быть загружен полностью, прежде чем начнется воспроизведение его содержимого. Впрочем, вы можете, конечно, определять общую стратегию загрузки

клипов. Клипы загружаются во Flash в том порядке, в котором они расположены на основном монтажном столе (например, клип, размещенный в кадре 1, будет полностью загружен, прежде чем начнется загрузка клипа, находящегося в кадре 2).

Если мысль об утрате потоковых возможностей приводит вас в уныние, задумайтесь над тем, какую роль играет видео в контексте вашего проекта. Бывает, что основной акцент делается именно на видео, и в таком случае, возможно, стоит разместить его на основном монтажном столе с возможностью потоковой передачи. Если же вы собираетесь как-то манипулировать видео материалом с целью создания какого-либо эффекта, то понадобится клип (об эффектах будет рассказано в главе 5).

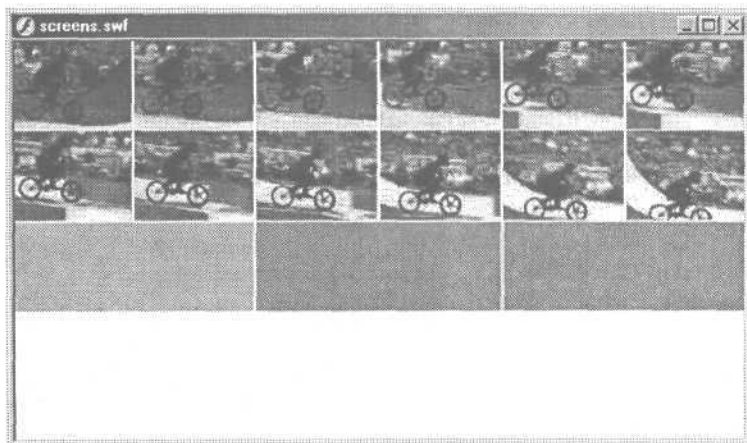
А сейчас давайте посмотрим, как сделать наше видео автономным и универсальным.

Видео-стена

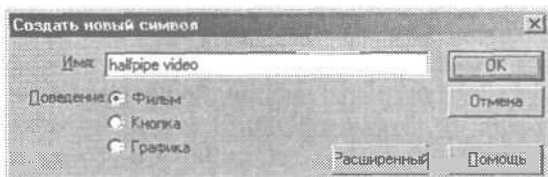
Сейчас мы создадим эффект, напоминающий так называемые видео-стенки, которые вам наверняка приходилось видеть — в торговых пассажах, на выставках и зрелищных мероприятиях.

1. Скачайте и откройте файл `halfpipe fla`. Этот файл содержит только внедренный видео файл в Библиотеке и ничего более. В предыдущих главах вы узнали предостаточно об импортировании, поэтому, если хотите использовать собственный видео материал, пожалуйста — импортируйте его в Библиотеку (Library) нового фильма Flash.

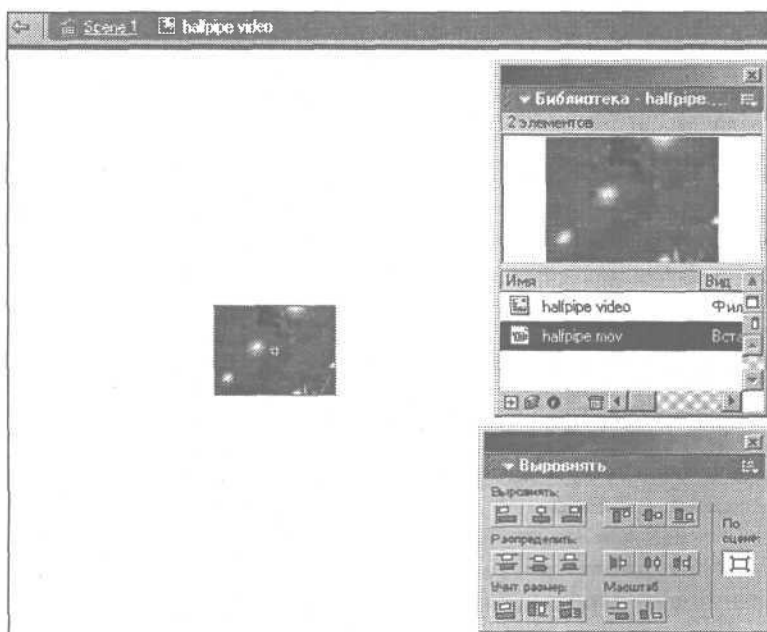
Взгляните, что у нас должно получиться — одновременно воспроизводится несколько копий одного и того же клипа, причем не синхронно, а со сдвигом на один кадр.



- Первым делом нам нужно преобразовать наш видеоклип в клип Flash. Создайте новый клип при помощи команды Вставить > Новый символ (Insert > New symbol). Назовем его «halfpipe video»:

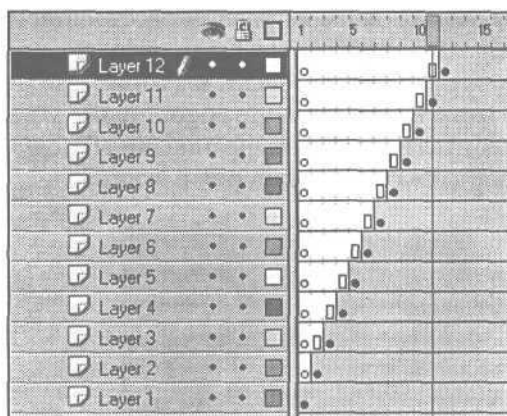


- Вы теперь находитесь в режиме редактирования символа. Перетащите копию halfpipe.mov из Библиотеки (Library) в клип-символ. В появившемся диалоговом окне нажмите кнопку ОК.
- Отцентрируйте видео в сцене, пользуясь панелью Выровнять (Align) — она открывается нажатием клавиш Ctrl + K. Убедитесь, что в панели выравнивания установлен режим По сцене (To stage), как показано на снимке.

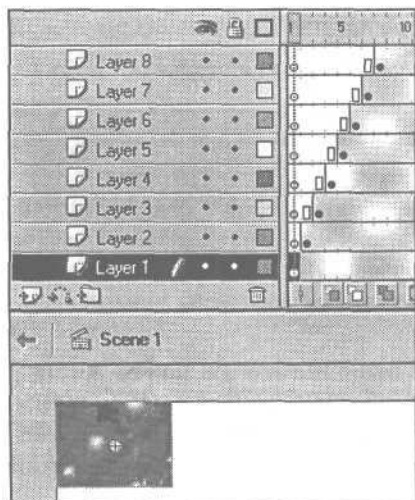


- Нажав кнопку Сцена 1 (Scene 1), расположенную чуть ниже монтажного стола, вернитесь к основному монтажному столу. Здесь мы будем творить наше маленькое волшебство.

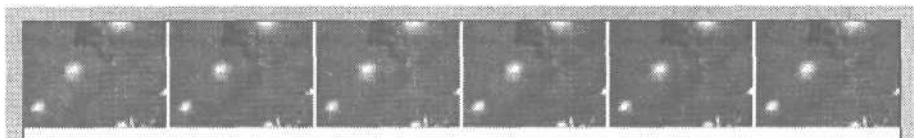
6. Добавьте на монтажный стол одиннадцать (да-да, одиннадцать!) новых слоев — всего у вас получится двенадцать слоев. В каждом из слоев видео будет начинаться с иного кадра, чтобы создать ступенчатый эффект. Двенадцать копий нашего видео уютно расположатся в верхней части Flash-сцены.
7. Теперь вставьте ключевой кадр в кадр 2 слоя 2, кадр 3 слоя 3 и так далее, до кадра 12 слоя 12. Эти кадры создадут аккуратный ступенчатый эффект, в котором каждый клип сдвинут относительно другого на один кадр:



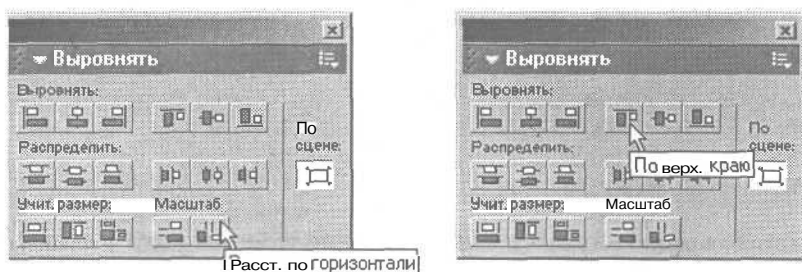
8. Перетащите копию клипа-символа «halfpipevideo» в кадр 1 слоя 1. Поместите его в левый верхний угол сцены:



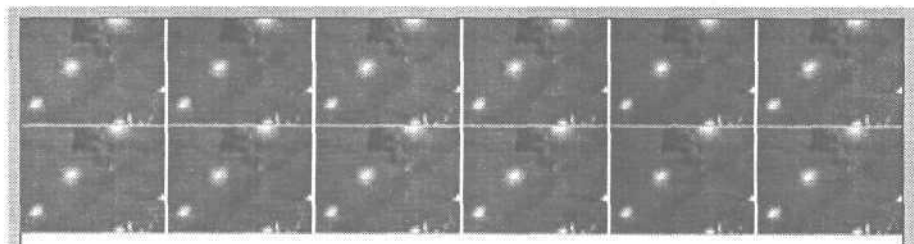
9. Перетащите копию клипа «halfpipevideo» в кадр 2 слоя 2 и поместите его рядом с предыдущим. Не старайтесь сейчас поместить его вплотную, поместите пока приблизительно. Повторяйте то же самое, пока не получится первый ряд из шести копий, как здесь:



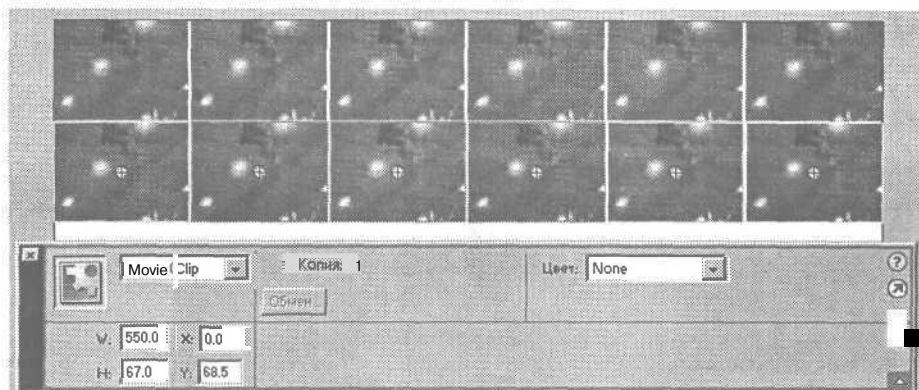
10. Теперь давайте их подравняем, расположим на равном расстоянии и прижмем к верхнему краю. Выделите все клипы и выстройте их с помощью кнопок панели Выровнять (Align) — Расст. по горизонтали (Space evenly horizontally и По верх. краю (Align top edge), только убедитесь сначала, что панель находится в режиме выравнивания По сцене (To Stage). Теперь клипы расположены равномерно и аккуратно.



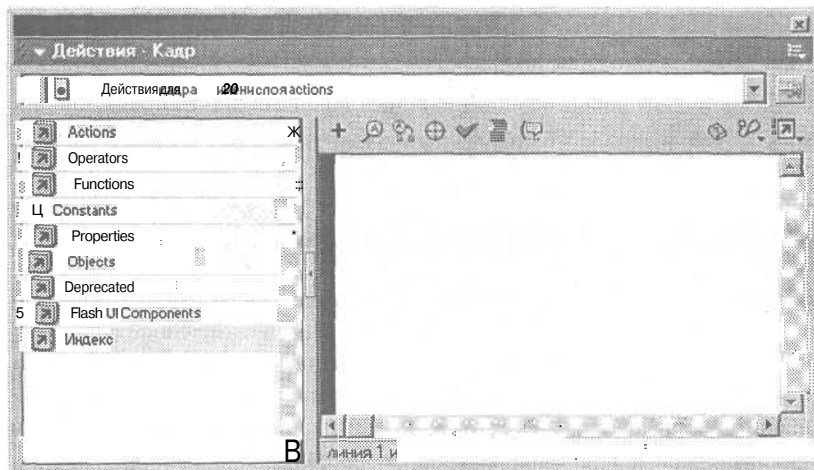
11. Теперь — остальные шесть слоев. Перетащите клипы в кадры 7 — 12 соответствующих слоев и расположите их под верхним рядом:



12. Выделите все клипы второго ряда и выровняйте их, нажав кнопку Расст. по горизонтали (Space evenly horizontally на панели Выровнять (Align). Не снимая выделения, введите значение «68,5» в поле Y инспектора Параметры (Property), чтобы клипы расположились ровной линией непосредственно под вторым рядом:



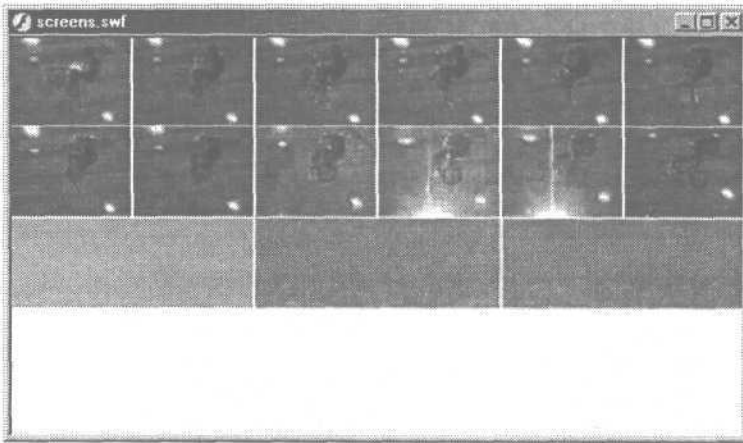
13. Растяните каждый клип на монтажном столе до кадра 60.
14. Прекрасно, теперь самое время протестировать фильм — Управление > Проверить фильм (Control > Test Movie). Полагаю, вы заметили, что клип зациклен, но как-то неправильно. Нужно кое-что изменить — остановить клип в первом окошке, чтобы в остальных он продолжался.
15. Создайте новый слой поверх остальных и назовите его «actions».
16. В кадр 20 слоя «actions» вставьте ключевой кадр. Вообще-то его можно поместить в любой кадр, начиная с 12, но мы возьмем небольшой отступ, чтобы не вносить путаницу на монтажный стол. Теперь выделите новый ключевой кадр и командой Окно > Действия (Window > Actions) откройте окно Действия (Actions):



17. В области скрипта введите:

```
stop();
```

18. Снова проверьте фильм. Теперь намного лучше, правда?



Я в своей версии добавил под видео-окошками простенькую графику, как вы можете видеть на снимках (или в окончательном файле — `halfpipe_final fla`), а вы можете заполнить нижнюю половину сцены чем хотите.

19. Сохраните фильм. Мы, пожалуй, найдем ему применение в следующей главе... эй, не подглядывать!

Такой эффект способен здорово оживить какую-нибудь веб-страницу. Причем весь эффект занимает всего 86 килобайт — благодаря тому, что вновь и вновь используется один и тот же клип-символ, который нужно загрузить всего один раз.

А теперь — несложные манипуляции с монтажным столом плюс немножко `ActionScript`.

Манипуляции с монтажными столами

Думаю, обычное воспроизведение видео с нормальной скоростью уже несколько поднадоело, поэтому давайте изучим что-нибудь еще. Поскольку клипы-символы видео мало чем отличаются от обычных клипов `Flash`, мы можем забавляться с их монтажными столами при помощи скриптов. Например, можем велеть им показать определенный кадр, или перейти на кадр назад, или на пять кадров вперед.

Давайте создадим блок управления, или контроллер, видео, позволяющий выполнять следующие действия:

- воспроизводить / останавливать;
- перематывать вперед / назад;
- переходить на один кадр вперед / назад;
- переходить к случайному кадру.

Таким образом, у нас получится нечто похожее на проигрыватели QuickTime или Windows Media, или на видеоплеер:



Кнопка «пауза»

Начнем с создания кнопки «пауза».

1. Скачайте и откройте файл `controller fla`. Как и в прошлом упражнении, этот файл содержит единственный видеоклип и ничего больше.

Видеоклип немаленький, с небольшим сжатием. Если бы мы собирались использовать его для не-широкополосного веб-сайта, понадобилось бы большее сжатие, но ведь нам он нужен лишь для проверки нашего контроллера.

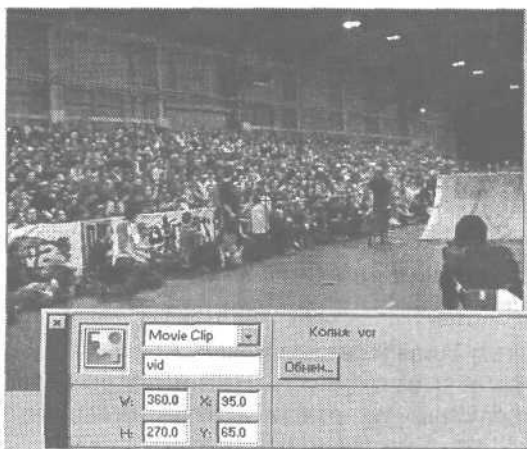
2. С помощью команды Вставить > Новый символ (Insert > New Symbol) создайте новый клип, пусть он называется «vcr». На монтажный стол «vcr» перетащите копию клипа `cross_ramps.mov`. В ответ на запрос о расширении монтажного стола нажмите кнопку Да (Yes).



3. Отцентрируйте видео в клипе с помощью инструментальной панели Выровнять (Align). Вставьте новый слой, а в нем — ключевой кадр параллельно последнему кадру слоя видео (у меня это кадр 117).
4. Щелкните на этом новом ключевом кадре и откройте панель Действия (Actions) (Окно > Действия (Window > Actions)). Введите действие остановки:

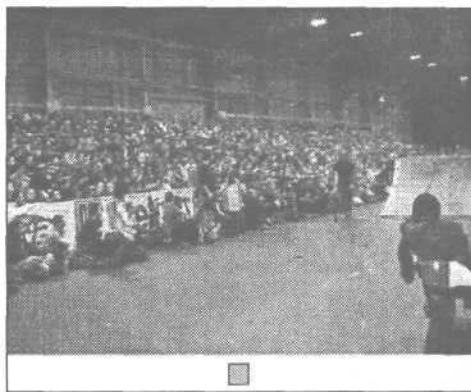
```
stop();
```

5. Клип-символ видео готов, поэтому вернитесь на главный монтажный стол и перетащите из Библиотеки (Library) копию клипа «vcr». Отцентрируйте клип в сцене и дайте ему ссылочное имя «vid» в инспекторе Параметры (Property):

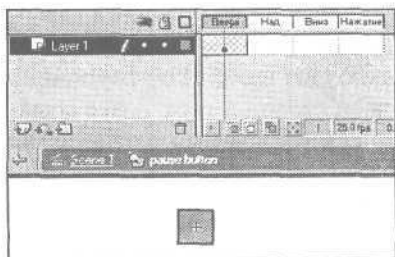


Ссылочное имя — это уникальный идентификатор клипа или иного объекта во Flash. Дав клипу уникальное имя, мы сможем впоследствии ссылаться на него в ActionScript.

6. Наше видео готово к манипуляциям над ним при помощи еще не созданных кнопок. Переименуйте оригинальный слой в «video» и создайте новый слой под названием «buttons» (кнопки).
7. В слое «buttons» изобразите красный квадратик с черной обводкой (с помощью инструмента Прямоугольник (Rectangle)), поместив его немного ниже нашего видео. Это будет кнопка паузы, и она положит начало нашему кнопочному царству.

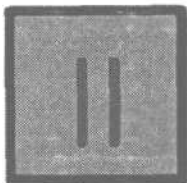


8. Выделите квадратик вместе с обводкой и преобразуйте его (F8) в графический (graphic) символ под названием «button» (кнопка).
9. Этот символ мы сможем использовать для всех наших кнопок. Сначала давайте создадим кнопку паузы. Выполните Вставить > Новый символ (Insert > New Symbol). Создайте символ типа Кнопка (Button) с именем «pause button».
10. На монтажный стол «pause button» перетащите копию графического символа кнопки и отцентрируйте его:

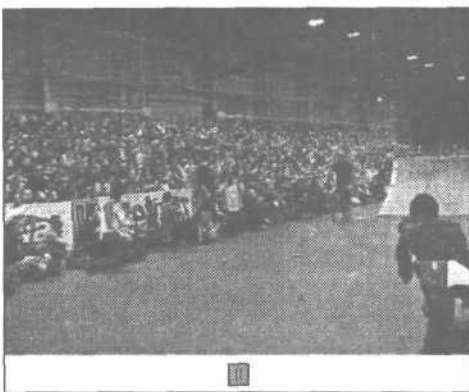


О других кнопках пока не беспокойтесь, их созданием мы еще успеем заняться.

11. Вставьте на монтажный стол кнопки новый слой под названием «detail» (отделка). Здесь изобразите универсальное обозначение паузы:



12. Вернитесь к основному монтажному столу. У вас должно получиться следующее:



13. Нам понадобится для кнопки маленький кусочек ActionScript. Выделите кнопку паузы и откройте панель Действия (Actions). Введите следующий программный код ActionScript:

```
on (release) {  
    _root.vid.stop ();  
}
```

Этот скрипт означает: «когда эта кнопка будет нажата и отпущена... остановить видео на основном монтажном столе». `_root` — это ссылка на главный монтажный стол, а `vid` — ссылочное имя, которое мы ранее дали видеоклипу.

14. Теперь проверьте фильм. Начнется воспроизведение видео, когда вы нажмете и отпустите кнопку паузы, воспроизведение остановится.

К сожалению, на данный момент это единственная функция. Нам этого явно недостаточно, надо что-то делать...

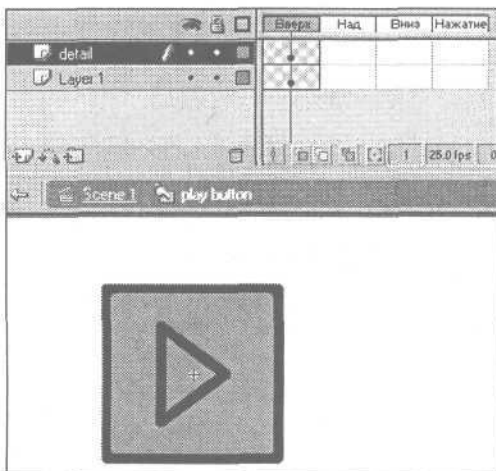
Кнопка воспроизведения

Теперь нам нужна кнопка воспроизведения.

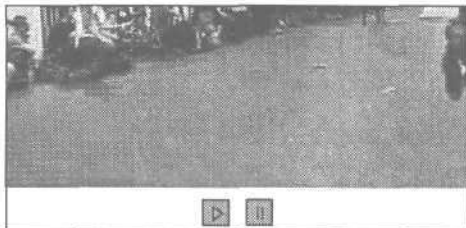
1. Щелкните правой кнопкой мыши (для Макинтошей — Ctrl + щелчок) на «pause button» в Библиотеке, и выберите в контекстном меню Дублировать (Duplicate):



2. В появившемся диалоговом окне введите «playbutton» в качестве имени и нажмите кнопку ОК. В Библиотеке двойным щелчком откройте только что созданную кнопку для редактирования.
3. В слое «detail» удалите символ паузы и изобразите знак воспроизведения:



4. Вернувшись на основной монтажный стол, перетащите сюда копию кнопки воспроизведения из Библиотеки. Поместите ее рядышком с кнопкой паузы:



5. Выделив кнопку «play button», откройте панель Действия (Actions). Вставьте следующий код:

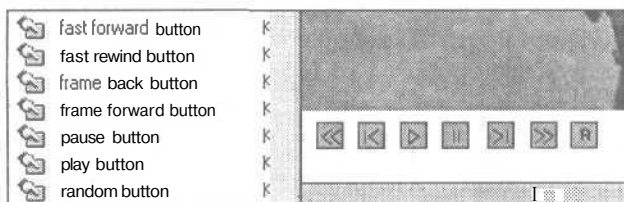
```
on (release) {
    _root.vid.play ();
}
```

Похоже на код для кнопки «пауза», но есть маленькое отличие. Вместо stop здесь play. Все просто, правда?

6. Протестируйте фильм и обе кнопки. Все в порядке? Идем дальше. Нужно добавить остальные кнопки — на кадр вперед («frame forward») и назад («frame back»), быстрая перемотка вперед («fast forward») и назад («fast rewind»), а также случайный кадр («random»).

7. Самостоятельно создайте эти кнопки тем же способом, что и кнопку воспроизведения:
 - сдублируйте кнопку паузы;
 - дайте соответствующее имя;
 - двойным щелчком откройте новую кнопку для редактирования;
 - измените вид кнопки в слое «detail»;
 - перетащите копию кнопки на основную сцену в слой кнопок («buttons»).

Вот как выглядят мои кнопки:



Если с этим покончено, нужно добавить скрипты, чтобы кнопки заработали. Давайте начнем с кнопок перехода на кадр вперед и назад.

9. Выделите на основной сцене кнопку перехода на кадр вперед и откройте панель Действия (Actions). Введите следующий код:

```
on (release) {  
    _root.vid.nextFrame();  
}
```

Та же самая схема, что и в предыдущих образчиках кода, только использован другой способ манипулирования видеоклипом. Метод nextFrame не нуждается в разъяснениях — он указывает нашему видео (vid) перейти к следующему кадру. Таким образом, при каждом нажатии на эту кнопку видео будет продвигаться вперед на один кадр.

10. Выделите кнопку перехода на кадр назад и в панели Действия введите:

```
on (release) {  
    _root.vid.prevFrame();  
}
```

Снова знакомый код — за исключением метода prevFrame: это всего лишь сокращение от «previous frame» — предыдущий кадр. Совершается, естественно, переход на один кадр назад.

11. Теперь у нас имеются кнопки покадрового перехода. Просто, а? Протестируйте фильм, и продолжим. Только не забудьте, что на данный момент работают только четыре кнопки.
12. Последняя из наших кнопок — «randombutton» — иллюстрирует возможность перехода к любому кадру видеоклипа. Выделите в сцене кнопку перехода к случайному кадру и введите следующий код в панели Действия:

```
on (release) {  
    _root.vid.gotoAndStop (Math.round (Math.random ()*116));  
}
```

Нуда, это выглядит немного посложнее, но это просто из-за ужасного количества скобок. Этот код попросту говорит: «Когда пользователь нажмет на кнопку, переместить курсор воспроизведения к случайно выбранному кадру с номером от 1 до 116».

116 — это не магическое число, это количество кадров в использованном мною видеоклипе! Функция `Math.round` возвращает округленное до целого значение числа — в данном случае случайного числа от 0 до 1, умноженного на 116.

13. Еще раз протестируйте фильм. Нажав на кнопку «random», вы волшебным образом переместитесь к случайно взятому кадру видеоклипа.
А теперь возьмемся за две оставшиеся кнопки быстрой перемотки.

Кнопки перемотки вперед и назад

Многие владельцы старых и дешевых видеомагнитофонов знают, какво это — подолгу, до боли в пальце, удерживать нажатой кнопку перемотки, чтобы найти нужное место на кассете.

Может, кому-то это покажется тривиальным, но я скажу: создавая любое приложение, очень важно продумать, каким образом должна работать та или иная функция. Так, я решил, что здесь уместен как раз метод «большого пальца», требующий, чтобы для быстрой перемотки пользователь удерживал нажатой кнопку мыши, — ведь клип-то короткий, и если использовать обычный щелчок мышью, то весь клип может быть перемотан прежде, чем пользователь успеет остановить его.

1. Выделите кнопку перемотки вперед и откройте ее Действия (Actions). Введите следующий скрипт:

```
on (press) {  
    _root.ffwd = true;  
}
```

```
on (release) {  
    _root.ffwd = false;  
}
```

Данный код состоит из двух частей, соответствующих двум действиям: `press` (нажать) и `release` (отпустить). Действие `press` присваивает флагу `ffwd` значение «истина» (`true`), разрешая быструю перемотку, а действие `release` (когда пользователь отпускает кнопку мыши) отключает перемотку. А где же проверяется флаг? Эта проверка будет происходить в самом видеоклипе.

2. Выделите видео и добавьте следующие действия:

```
onClipEvent (enterFrame) {  
    if (_root.ffwd == true) {  
        frame = this._currentframe +5;  
        this.gotoAndStop (frame);  
    }  
}
```

На сей раз мы вставили код непосредственно в клип. Если вам показалось, что смысл его трудно постигнуть, не паникуйте. Давайте разберем его по шагам. Если перевести этот код на простой язык, то он будет выглядеть так:

- в каждом кадре фильма Flash
- проверять флаг, и если его значение — «истина», то:
- запомнить сумму номера текущего кадра видео + 5
- переместить указатель воспроизведения к кадру, номер которого равен запомненному числу.

Если же флаг — не истина, все действия, заключенные в фигурные скобки после оператора `if`, игнорируются. Здесь `frame` — переменная, контейнер для хранения информации, которая впоследствии может быть использована во Flash. Вероятно, вы заметили, что моя перемотка несколько быстрее, чем обычно, ведь проходится 5 кадров за раз.

3. Проверьте фильм. При нажатии кнопки перемотки фильм должен ускоряться, но только пока вы не отпустите кнопку мыши. Если хотите снизить скорость перемотки, достаточно просто заменить в предыдущем скрипте «5» на другое, меньшее, число.
4. Теперь так же обойдемся с кнопкой перемотки назад («`rewind button`»). Выделите ее и введите следующий код:

```

on (press) {
    _root.frwd = true;
}
on (release) {
    joot.frwd = false;
}

```

Как видите, действия те же самые, только имя флага другое.

5. Выделите в сцене видеоклип и вставьте перед последней фигурной скобкой следующий код:

```

if (joot.frwd == true) {
    frame = this._currentframe -5;
    this.gotoAndStop (frame);
}

```

Таким образом, полностью код для видеоклипа должен выглядеть так:

```

onClipEvent (enterFrame) {
    if (_root.ffwd == true) (
        frame = this._currentframe +5;
        this.gotoAndStop (frame);
    )
    if (joot.frwd == true) {
        frame = this._currentframe -5;
        this.gotoAndStop (frame);
    }
}

```

В каждом кадре проверяются значения каждого из флагов, то есть, не нажаты ли кнопки перемотки. Само собой, сразу обе кнопки нажать невозможно, тем не менее, проверять необходимо оба условия.

6. Прежде, чем закрыть панель Действия (Actions), добавьте в начало предыдущего кода еще пару строк:

```

onClipEvent (load) {
    this.stop();
}

```

Это предотвратит проигрывание клипа сразу после его загрузки во Flash. То есть, как только Flash-фильм начнется, vid будет поставлен на паузу на кадре 1, после чего пользователь сможет управлять им по своему усмотрению.

7. Теперь проверьте фильм. Как видим, кнопка перемотки вперед работает отлично, а вот с обратной перемоткой — проблемка. Перемотка происходит не так быстро, как вперед. Дело в том, что при воспроизведении видео производится буферизация в память Flash Player, поэтому с переходом вперед проблем никаких, а вот назад...

Итак, наш видео-контроллер закончен, все, что осталось — это небольшая отделка. Например, окончательный вид моего контроллера (controller_final fla) таков:



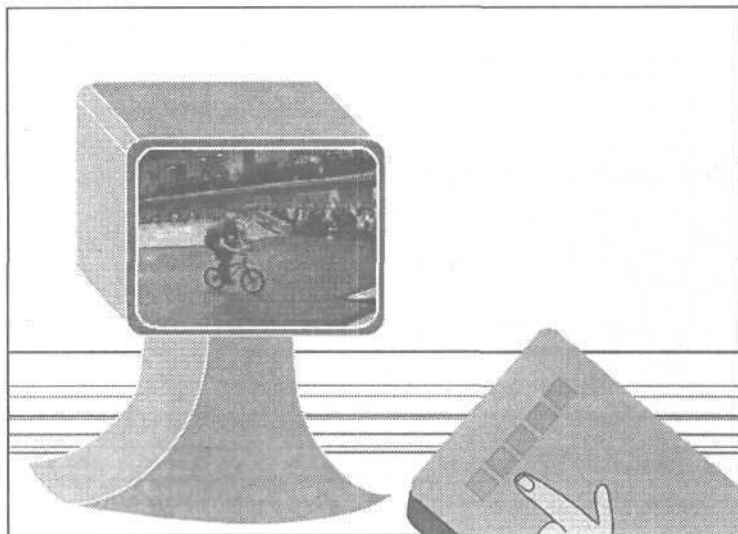
В этом упражнении мы узнали много полезных вещей о монтажных столах видеоклипов и способах манипулирования ими во Flash. Как вы видели, для этого нужно лишь немного ActionScript, ничего особенно сложного, тем более что все эти скрипты вы можете использовать повторно.

Главное, о чем не следует забывать, — необходимо дать вашему клипу-символу ссылочное имя в инспекторе Параметры (Property), а также указать ссылку на место размещения вашего клипа (в данном упражнении это `_root` — основной, или «корневой», монтажный стол).

Управление несколькими видеоклипами

Мы умеем размещать видеоклипы, а также управлять ими при помощи основных кнопок. А значит, мы сможем создать для доступа к нашим

клипам интерфейс в виде телевизора (для тех, у кого черно-белая картинка, подскажу, что телевизор у меня — желто-оранжевый). Первым делом мы рассортируем видеоклипы, затем добавим кнопки управления, а в конце — звук.

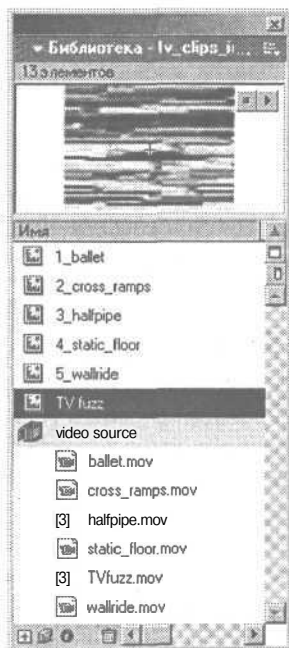


Хочу Flash-телевизор!

Главный монтажный стол для нашего телевизора будет состоять из шести кадров — по одному для каждого клипа, плюс один для «снега» — помех при переключении «каналов». Кнопки послужат для навигации, а также для глушения звука предыдущего клипа — каждому «каналу» соответствует отдельная звукозапись, воспроизводимая в том же темпе, что и клип.

1. Скачайте с сайта и откройте файл `tv_clips fla`. Он содержит импортированные клипы и звукозаписи, используемые в этом упражнении. Как и прежде, вы вольны использовать свои собственные файлы — импортировать видео вы умеете, о звуке подробно рассказывается в главах 6 и 7.
2. Первым делом надо поместить видеоклипы в клипы-символы. Используя команду Вставить > Новый символ (`Insert > New Symbol`), создайте новый клип-символ под названием «`1_ballet`».
3. На монтажный стол символа `1_ballet` перетащите из Библиотеки копию клипа `ballet.mov` (любители этого вида спорта тут же заявят, что этот трюк называется «Flatlands», а никакой не балет — ну, приношу извинения!).

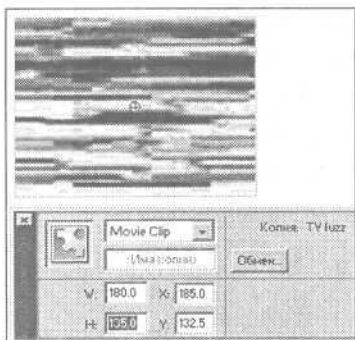
4. На запрос об увеличении числа кадров в появившемся диалоговом окне ответьте «да» и отцентрируйте видео в сцене.
5. Создайте клипы-символы для всех оставшихся видеоклипов (включая TVfuzz.mov), не забывая расширять монтажный стол каждого клипа. Дайте символам имена — такие, как на снимке:



6. Покончив с этим, создайте в Библиотеке новую папку под названием «video movie clips» и сложите в нее все созданные клипы-символы, чтобы не создавать беспорядка и путаницы в нашей Библиотеке.
7. Вернувшись на основной монтажный стол, дайте имеющемуся слою новое имя — «videos». Клавишей F7 вставьте пустые ключевые кадры в каждый кадр до 6 включительно:

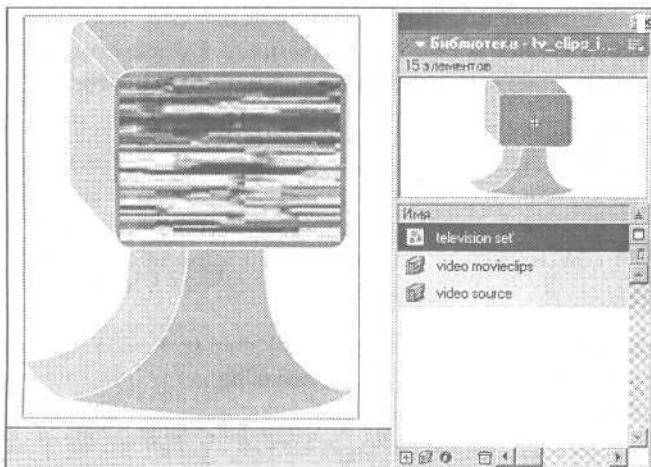


8. Перетащите копию символа TV_fuzz на сцену — в кадр 1 слоя videos. В инспекторе Параметры (Property) измените размер фильма на 180x135 точек, затем отцентрируйте клип.



Теперь клип со «снегом» такого же размера, как остальные импортированные клипы. Этот клип символизирует помехи, мы можем спокойно изменять его размер — чем больше искажений, тем лучше эффект.

9. Теперь давайте нарисуем сам телевизор. Создайте новый слой под названием «TV» и поместите его ниже слоя videos.
10. Заблокируйте слой videos, и используйте клип TV fuzz как шаблон для изображения телевизора в слое TV. Если нужно закруглить углы «ящика», в инструменте Прямоугольник (Rectangle) используйте опцию Радиус скругления прямоугольника (Round Rectangle Radius). В общем, подойдите к этому творчески и создайте свой идеальный телевизор, а хотите — импортируйте картинку с изображением реального телевизора.
11. Закончив рисовать, выделите весь свой телевизор и преобразуйте его в графический символ под названием «television set»:

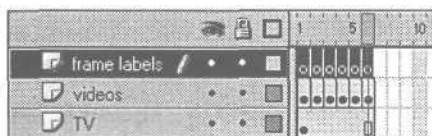


12. Заблокируйте слой TV и разблокируйте слой videos. Перетащите символ 1_ballet в кадр 2 слоя videos и отцентрируйте.

13. Перетащите последовательно оставшиеся клипы в кадры с 3 по 6, последним будет 5_wallride в кадре 6. Все их отцентрируйте.

Теперь нужно сделать кое-что особо важное. Во-первых, надо установить метки кадров, чтобы связать их с нашими кнопками.

14. Создайте новый слой под названием «frame labels» (метки кадров). Поместите его поверх двух других слоев. Как вы уже делали недавно, вставьте ключевые кадры в каждый кадр слоя frame labels до 6 включительно:



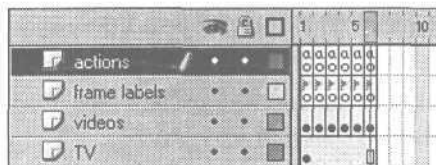
15. Выделите кадр 1 и назначьте ему в инспекторе Параметры (Property) метку кадра (frame label) «fuzz». Мы будем использовать ее для того, чтобы возвращать воспроизведение «помех» после того, как клип будет проигран до конца.

16. Выделите кадр 2 и назначьте метку 1. Наверное, вы уже поняли, что она соответствует цифре-приставке в имени клипа. Пометьте кадры 3 — 6 цифрами 2 — 5 (то есть номер кадра минус 1). И вот теперь мы видим на монтажном столе множество красных флажков. Да здравствует День Победы! Ошиб... ну, успокойтесь. Это просто метки кадров загружены, извините.

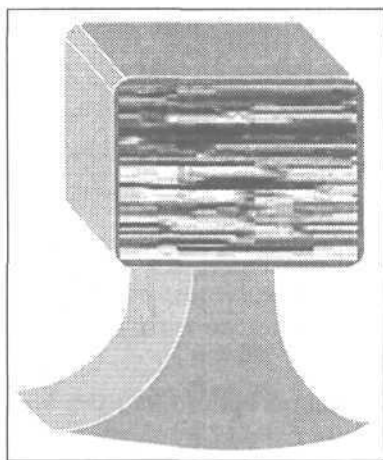
Эти метки будут служить ссылками для кнопок каналов, которые мы создадим чуть позже. Но сначала мы остановим кадры на монтажном столе.

17. Создайте еще один слой, назовем его «actions», то есть «действия». Расположите его поверх остальных слоев и, как прежде, вставьте ключевые кадры до 6 включительно. Помещать слой actions выше всех остальных — общая практика, это упрощает расшифровку фильмов Flash.

18. Выделите кадр 1 слоя actions и вставьте действие stop. То же самое сделайте с остальными кадрами в этом слое, и монтажный стол должен приобрести такой вид:



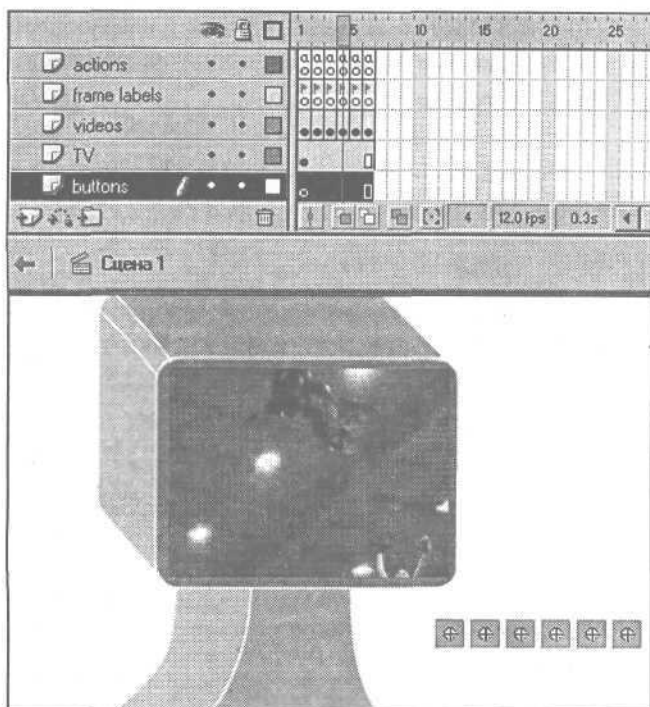
Если сейчас вы протестируете фильм, то увидите сплошные помехи и больше ничего. Монтажный стол остановлен на кадре 1 и клип fuzz прокручивается снова и снова:



Кнопки управления

Как же нам прекратить эти помехи и посмотреть что-нибудь поинтереснее? Пора заняться кнопками...

1. Создайте новый символ-кнопку под именем «change button».
2. На монтажном столе символа нарисуйте небольшой квадратик, поместив его в центре.
3. Вернувшись на основной монтажный стол, создайте новый слой под названием «buttons» (кнопки), поместив его ниже всех прочих слоев. Перетащите в слой buttons пять копий символа change button:



4. Каждую кнопку нужно запрограммировать на включение одного из каналов, поэтому выделите крайнюю слева кнопку и откройте панель Действия (Actions). Введите следующий код:

```
on (release) {  
    if (_currentframe !=2){  
        gotoAndPlay ("1");  
    }  
}
```

Данный код указывает Flash, что следует переместить указатель воспроизведения основного монтажного стола к кадру с меткой «1» (это у нас второй кадр).

Вместо метки можно было бы использовать просто номер кадра, но метки зачастую бывают удобнее (например, если материал на монтажном столе постоянно сдвигается, пришлось бы обновлять номер кадра). Мы могли бы написать вторую строку в виде `_root.gotoAndPlay («1»)`, но, поскольку кнопки расположены на `_root` (главном монтажном столе), дополнительно ссылаться на него нет смысла.

5. Выделяем вторую кнопку и вводим в панели Действия следующий код. Как и в случае с предыдущей кнопкой, индикатор воспроизведения отсылается к помеченному кадру:

```
on (release) {
    if (_currentframe !=3) {
        gotoAndPlay ("2");
    }
}
```

6. Повторите то же самое с остальными кнопками, каждый раз прибавляя единицу к номеру метки кадра; для последней кнопки код должен выглядеть так:

```
on (release) {
    if (_currentframe !=6) {
        gotoAndPlay ("5");
    }
}
```

7. Покончив с этим делом, протестируйте фильм, теперь вы можете, нажимая на кнопки, переключать каналы. Разве не классный у нас получился пульт управления?

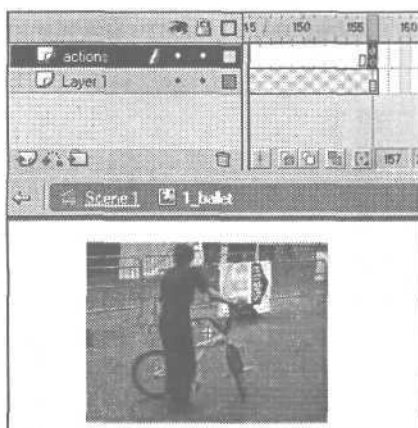
Проблема в том, что «помехи» мы видим только в самом начале, а клипы зациклены, ведь мы им никаких дополнительных указаний не давали. В каждый видеоклип следует вставить строчку ActionScript, чтобы по окончании воспроизведения клипа индикатор воспроизведения отсылался обратно к помехам.

8. Двойным щелчком на символе 1_ballet в Библиотеке откройте его для редактирования. Создайте новый слой и назовите его «actions».
9. Отыщите последний кадр видеоклипа и в параллельный кадр слоя actions вставьте ключевой кадр, как показано на рисунке вверху следующей страницы.
10. Откройте окно Действия (Actions) и вставьте следующий код:

```
_root.gotoAndStop ("fuzz");
```

Это просто указание Flash перейти к метке fuzz (кадру 1) на основном монтажном столе, когда воспроизведение клипа закончится.

11. Повторите шаги 8 — 10 для всех видеоклипов. Не забывайте вставлять ключевой кадр параллельно последнему кадру видео.

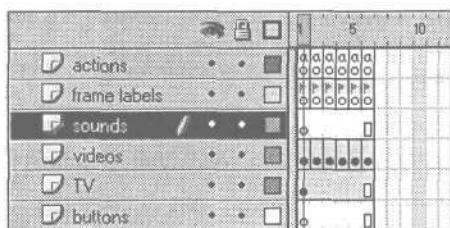


12. Проверьте фильм — клип воспроизводится, а по окончании переходит обратно к помехам. В телевизоре, конечно, не совсем так, но все-таки лучше, чем реклама.

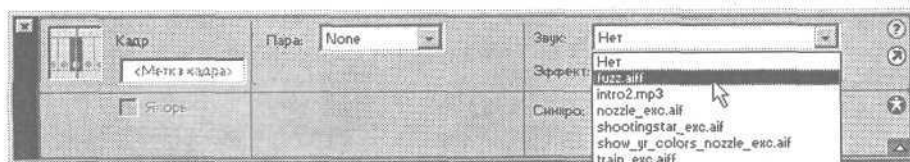
Добавляем звук

Наш домашний театр почти готов. Осталось только озвучить его и добавить маску, чтобы скруглить углы телеэкрана.

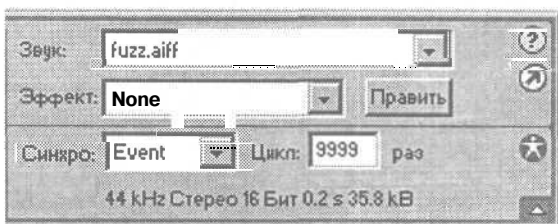
1. Вставьте на основной монтажный стол новый слой под названием «sounds» (звук). Поместите его над слоем videos. Как и в других слоях, вставьте ключевые кадры в каждый кадр слоя до 6 включительно:



2. Выделите кадр 1 слоя sounds и откройте список Звук (Sound) в инспекторе Параметры (Property):



3. Из списка выберите `fuzz.aiff`. В опции Цикл (Loop) задайте значение «9999»:



Максимальное число повторов во Flash — 9999, и нам остается надеяться, что наши пользователи не станут вглядываться в помехи так долго, что успеют прослушать все эти повторы.

4. Для кадра 2 этого слоя выберите из списка Звук (Sound) `train_exc.aiff`. В поле Цикл (Loop) оставьте стандартный 0, поскольку он должен быть проигран только один раз (продолжительность звукозаписи примерно такая же, как видеоклипа).
5. Остальные звуки у нас будут такими:
- Кадр 3: `intro2.mp3`
 - Кадр 4: `show_yr_colors_nozzle.aif`
 - Кадр 5: `nozzle_exc.aiff`
 - Кадр 6: `shootingstar_exc.aif`
6. Сделав это, проверьте фильм. Если вы будете быстро нажимать на кнопки, на вас обрушится стена шума. Вслушавшись в этот гул, вы обнаружите непрекращающийся шум помех.
7. К счастью, с помощью ActionScript это можно устранить. Выделите крайнюю слева кнопку и откройте окно Действия (Actions). Вставьте строку кода после первой:

```
stopAllSounds ();
```

Полностью код для этой кнопки будет выглядеть так:

```
on (release) {
    if (_currentframe !=2) {
        stopAllSounds ();
        gotoAndPlay ("1");
    }
}
```

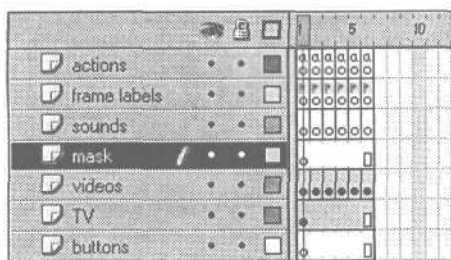
Конечно, это средство весьма радикально, но нам нужно, чтобы в каждый момент времени воспроизводилась только одна звукозапись, поэтому такое решение подойдет как нельзя лучше.

8. Теперь нужно добавить тот же код для каждой кнопки.
9. Проверьте фильм, и теперь переходы между звуковыми дорожками должны быть безупречными. Единственная проблема, которую можно отметить — то, что звуковая дорожка клипа иногда продолжается уже после того, как клип закончился, и начались «помехи».
10. Для устранения этого явления добавьте строчку кода `stopAllSounds()`; в начале того кода, который мы вставили в конец каждого видеоклипа (`_root.gotoAndPlay («fuzz»);`).

Маскирование

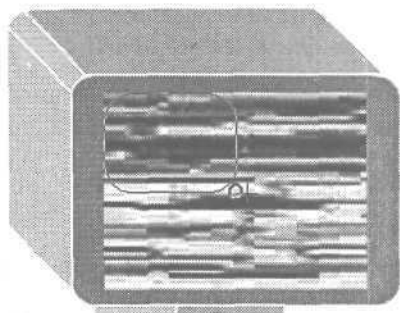
Последнее, что мы сделаем — наложение на видео маски с закругленными углами. Для Flash это обычный эффект, наверняка вы видели подобное тысячи раз. В том и прелесть видео во Flash, что оно становится как бы единым целым с прочим содержимым Flash-фильма.

1. Вставьте новый слой под названием «mask» (маска). Поместите его непосредственно над слоем `videos`. Нам нужно нарисовать прямоугольник со скругленными углами по размеру нашего телеэкрана.

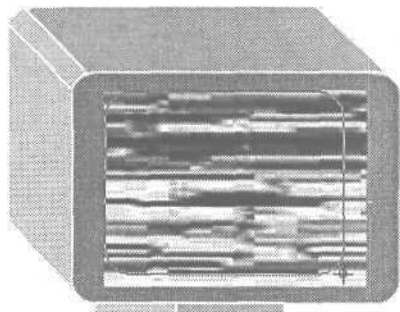


2. Для облегчения этой работы нужно включить опцию Привязать к объектам (Snap to Objects). Раскрыв меню Вид (View), проверьте, включена ли она.
3. Выберите инструмент Прямоугольник (Rectangle), в качестве цвета заполнения и обводки выберите отвратительный ядовито-зеленый цвет. На самом деле это не обязательно должен быть зеленый, просто такой цвет, который вы вряд ли станете использовать в своем творчестве. Этот прямоугольник станет маской, и цвет будет обозначать прозрачное «окно», сквозь которое будет смотреть пользователь, а нашего цветного прямоугольника он не увидит никогда.

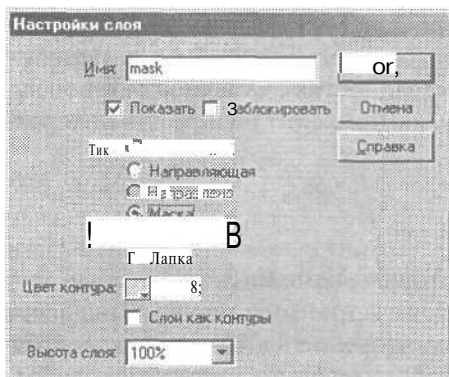
4. Выберите опцию Радиус скругления прямоугольника (Round Rectangle Radius) и установите то же значение, какое вы использовали, рисуя телевизор — в моем случае 10, — чтобы у нашей маски были закругленные углы.
5. В кадре 1 слоя mask подведите курсор к левому верхнему углу видеоклипа fuzz.



6. Нажмите кнопку мыши и растягивайте фигуру по размеру изображения видеоклипа. Вы заметите, что прямоугольник автоматически «прицепился» к верхнему углу.



7. Растянув фигуру до нижнего правого угла, отпустите кнопку мыши. Теперь ярко-зеленый прямоугольник со скругленными углами закрывает весь экран нашего телевизора.
8. Двойным щелчком на значке (в виде листа бумаги) откройте Настройки слоя (Layer Properties), в списке типов выберите Маска (Mask). Нажмите кнопку ОК, и значок рядом с именем слоя превратится в квадратик с овалом внутри.

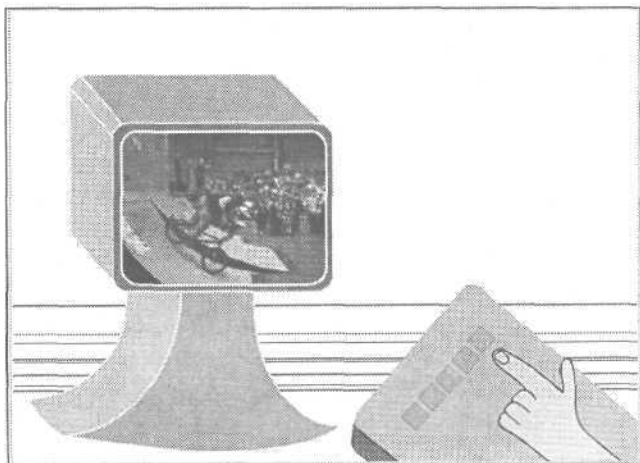


9. Нажмите и тащите слой `videos` под букву «m» слоя `mask`:



10. Отпустите кнопку мыши — значок слоя `videos` изменится, а само имя слоя сдвинется. Это значит, что он теперь маскирован слоем `mask`. Проверьте фильм, и вы увидите, что теперь форма нашего телеэкрана стала более реалистичной.

Итак, наш телевизор готов! Вы можете продолжить самостоятельно, выбрав собственные клипы, звуковые дорожки, улучшив графику. Я вот еще добавил пульт и руку, которая переключает каналы, а также рамку вокруг видеоклипов, чтобы резче выделить их. Если хотите посмотреть на то, что у меня получилось, скачайте файл `tv_final fla`:



В этом упражнении мы показали, как можно **дать** возможность пользователю самому выбирать **порядок** просмотра ваших клипов, не ограничивая его линейной последовательностью. Обычное сетевое видео попросту не позволяет этого, и здесь приходит на помощь интерактивность Flash.

Заключение

В данной главе мы рассмотрели некоторые возможности, которые дает нам использование видео во Flash. Понимание того, что во Flash все, в том числе видео, базируется на идеологии монтажных столов, состоящих из кадров, открывает перед нами возможности достаточно свободного манипулирования материалом, — того, на что просто неспособны иные проигрыватели.

Мы увидели, насколько просто организовать подачу видео в сочетании с графическими и другими материалами. Таким образом, можно создавать целые веб-страницы с внедренным видео. Скажите «до свидания» всяким дополнительным окнам и внешним программам, встречайте веб-сайты, созданные целиком во Flash!

Если вас в какой-то степени напугало частое использование ActionScript в данной главе, спешу утешить — в следующей главе мы рассмотрим эффекты, которые можно создавать без всяких скриптов.

Применение эффектов для Flash видео

Краткое содержание главы:

- различные *способы маскирования видео материала*;
- *управление параметрами клипов — такими, как вращение, альфа-фактор, а также применение необычных эффектов к видео материалу*;
- *использование этих возможностей для повышения четкости изображения*.

Flash позволяет не только управлять видео материалом, но и изменять его. Для видео, вставленного в клип-символ, мы можем изменить такие параметры, как тонирование, размер, контрастность, яркость, можем повернуть кадр и так далее. В этой главе мы рассмотрим эти свойства (без применения ActionScript), покажем, как можно создать с их помощью всякие классные эффекты, как интегрировать видеоматериал в общий дизайн веб-страницы.

Прежде чем приступить к изучению этих параметров, мы сделаем краткий обзор несложных приемов маскирования — вы увидите, что творческий подход к применению масок может дать потрясающий эффект.

Нестандартные приемы маскирования

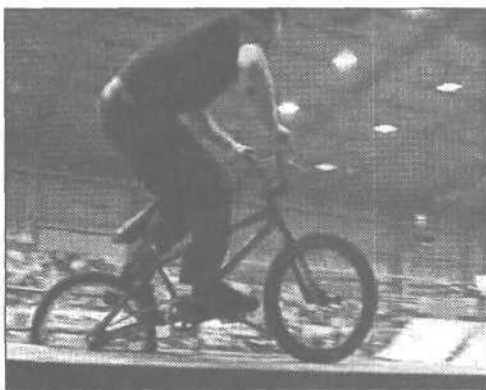
Теперь, с Macromedia Flash MX, мы более не обязаны отображать видео в отдельном окне традиционной прямоугольной формы. Видео может

быть встроено в интерфейс, уютно устроившись среди других элементов оформления веб-страницы. Существует два способа такой интеграции видео: при помощи продуманного применения **слоев** и при помощи **масок**.

Прежде, чем мы рассмотрим примеры, хочу немного порассуждать о том, что необходимо принимать во внимание при использовании любого из этих методов. Прежде всего, не закрывайте слишком большую часть изображения вашего видеоролика! Ну вот, скажете вы, сначала было столько разговоров о том, что маски — это хорошо, а теперь... Но подумайте сами: какой смысл маскировать изображение там, где его можно просто обрезать — еще до импортирования? Заранее обрезав «ненужную» часть кадра, мы уменьшим размер конечного файла!

Здесь важно упомянуть, что использование во Flash масок (особенно анимированных масок) отнимает не так уж мало вычислительных ресурсов компьютера, — сами понимаете, при показе видео разбрасываться нежелательно. Лучше всего было бы протестировать эффект маски на нескольких машинах, дабы убедиться в том, что при воспроизведении не возникает заметных задержек.

Для примера вообразим, что нам захотелось при помощи наложения слов создать имитацию широкоэкрannого кино (ну, вдруг кто-то из вас, как я, испытывает подсознательную любовь к этим черным полосам). Если мы импортируем во Flash вот такой клип:



...а затем имитируем «широкий экран», поместив два черных прямоугольника в слой, расположенном на монтажном столе выше слоя видео:



...то мы заплатим за это — размером файла и неоправданной загрузкой вычислительных мощностей пользователя, ведь те области видео изображения, которые мы закрыли, тем не менее, содержатся в файле и обрабатываются процессором. Если вам нужен эффект широкого экрана или нечто подобное, гораздо лучше будет обрезать изображение еще до импортирования, в видео редакторе (например, опция маскирования в QuickTime — хорошо и быстро). Другой вариант — экспортирование в фильм — с широкоэкранным соотношением сторон (16:9). А уже после обрезки или изменения соотношения можно сделать имитацию во Flash, просто добавив два черных прямоугольника.

Ну, а теперь давайте рассмотрим несколько примеров нестандартного, творческого использования масок для видео. Все исходные FLA-файлы для этой главы имеются на сайте www.friendsofed.com.

Текст в качестве маски

Используя текст как маску, можно, например, быстро «оживить» скучный логотип, но эта техника требует продуманного подхода. Текст занимает на экране очень мало реального пространства, поэтому вполне возможно, что большая часть маскированного видео изображения будет «растрочена» впустую. Это изображение будет содержаться в файле и обрабатываться, но пользователь его не увидит. Чтобы применение этой техники стало оправданным, необходимо найти сбалансированное решение, такое, чтобы видео достаточно хорошо просматривалось сквозь маскирующий текст.

Давайте рассмотрим пример (если хотите детально рассмотреть этот эффект, исходный файл называется `masking_with_text fla`).

Неплохой баланс, на экране много видео, сквозь текст достаточно хорошо просматриваются детали:



Маска состоит из центрального прямоугольника и текста над и под ним; текст я преобразовал в графические примитивы:

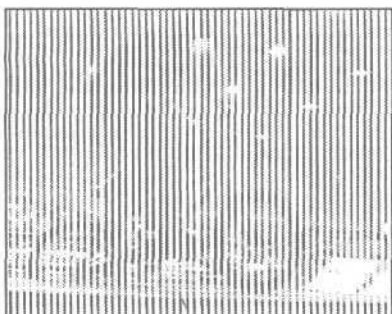


Как видите, создать такой эффект несложно, а выглядит очень здорово, объединяя видео и логотип. Ценнее всего то, что видео-элемент действительно очень оживит веб-страницу — статичная JPEG-картинка вряд ли создала бы столь же приятное впечатление.

Текстуры

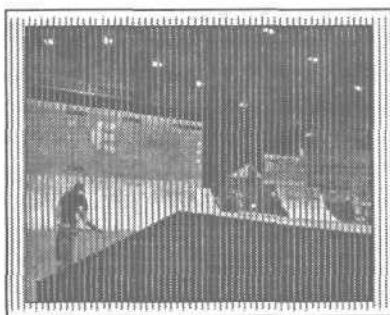
Не существует каких-либо правил и ограничений, которым нужно следовать при создании масок. Мой совет — выбрав инструмент Кисть (Brush), нарисуйте что-нибудь через всю сцену, а затем протестируйте фильм. Порой результаты небольшой импровизации бывают совершенно неожиданными.

В следующем примере (`masking_texture.fl`) для создания маски была нарисована одна-единственная линия, которая затем была преобразована в графический символ и многократно продублирована — в итоге линии покрыли весь видеоклип:



Нечего и говорить, что во Flash вы не создадите маску, используя штриховые изображения (вроде линий, начерченных с помощью инструментов Карандаш (Pencil) и Линия (Line), или штриховых очертаний кругов и прямоугольников). Однако это ограничение можно обойти, преобразовав штриховые фигуры в «заполненные» — при помощи команды Изменить > Форма > Преобразовать линии в заполнения (Modify > Shape > Convert Lines to Fills). Если нужно создать маску из линий, нарисованных от руки (свободным перемещением указателя), можете воспользоваться этим советом, хотя проще рисовать с помощью инструмента Кисть (Brush), поскольку он рисует заполнения, а не штрихи.

Вот как выглядит маска на сцене Flash в моем примере:



В файле `masking_texture_tween fla` я попробовал усилить эффект из предыдущего примера, заставив маску двигаться. Маска появляется сверху, из-за края сцены, и затем спускается вертикально вниз, пока не исчезнет за нижним краем сцены.

В этом примере наглядно проявляется проблема, которую мы обсуждали ранее: неэффективный размер файла. В каждый момент времени **видна** лишь небольшая часть видео изображения, и хотя эффект смотрится интересно, он не пригоден для того, чтобы донести до зрителя какую-либо ценную информацию. Возникает несколько важных вопросов:

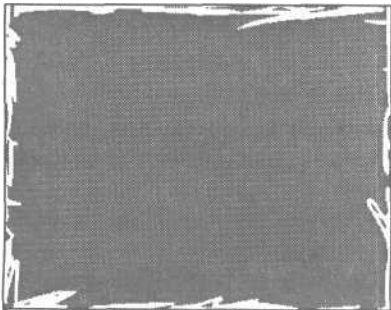
- для чего вы используете видео?
- является ли видео значимой составляющей вашего сайта или вы используете его только ради интересного эффекта?
- какое впечатление должно создавать видео у пользователя?

Следующий эффект маски (`masking_texture_edges fla`) слегка напоминает вид необрезанной фотографии. Устраняя отвратительно ровные края, он придает клипу такую «шероховатость», что превосходно подходит для моего ВМХ-материала:



Эту маску можно анимировать, создав несколько кадров с различными каракулями по краям; тогда маска будет «работать» вместе с материалом, усиливая характерность эффекта. Очевидное отличие от предыдущего примера в том, что здесь маска — лишь способ подачи материала, он тонко подчеркивает содержание, а не скрывает его, не заглушает эффектом самим по себе.

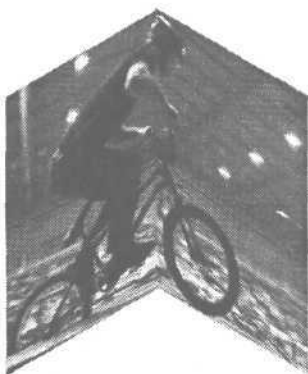
Эта маска сделана просто: я изобразил прямоугольник, полностью закрывающий видеоклип, а затем как попало «повозил» по его краям ластиком — инструментом Стирание (Eraser):



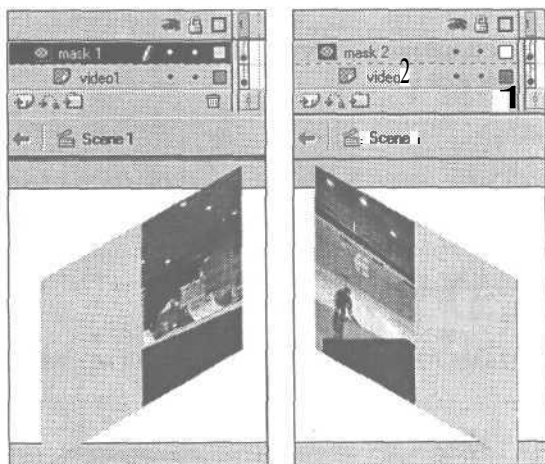
Множественные маски

ЕСЛИ у вас достанет терпения, вы можете комбинировать различные маски и видеоклипы — вариантов здесь бесконечное множество. Правда, такие комбинации применяются гораздо реже, чем маски из предыдущих примеров; тем не менее, их стоит рассмотреть.

В первом примере (`multiple_mask_skew.fla`) использованы два одинаковых видеоклипа, скошенные в разные стороны. Две маски придают изображению подобие раскрытой книги:



Здесь нужно потрудиться побольше, чем в предыдущих примерах, однако ничего сложного нет. Оба видеоклипа сначала отцентрированы, а затем скошены на 30 и -30 градусов соответственно. Маски сформированы из прямоугольников, которые уменьшены наполовину по ширине, и затем скошены. Вот эти два клипа и соответствующие им маски:



Помните: вы можете проверить, как выглядит эффект маски, еще до тестирования фильма — просто заблокировав маску и маскируемые слои.

Если вы заинтересовались масками и созданием различных эффектов на их основе, вы можете почерпнуть некоторые идеи у Lifaros'a — www.friendsofed.com/fmc/lifaros.

Итак, мы бегло познакомились с эффектами маскирования, а теперь давайте обратимся к свойствам видеоклипа-символа, при помощи которых можно манипулировать видео во Flash.

Параметры видеоклипа

Напомню, что, изменяя свойства видеоклипа, можно совершать над видео много забавных манипуляций. Эти свойства позволяют добиться лучшей интеграции видео в ваш сайт, а также открывают поле для всяческих сумасбродных экспериментов.

Сейчас мы рассмотрим по отдельности каждый из параметров и для каждого выполним мини-упражнение. Наверняка с большинством этих параметров вы разобрались бы и без пояснений, но, поскольку в данном случае речь идет о видео, возможно, вы все же узнаете что-то новое.

В качестве отправной точки для всех последующих упражнений используется файл `properties_base fla`; если вы хотите следовать нашим примерам в точности, можете, как обычно, взять его на сайте www.friendsofed.com.

Рассматриваемые здесь параметры не требуют привлечения скриптов (расслабимся после предыдущей главы). Но, если вы заинтересовались использованием `ActionScript` для управления видео и фильмами Flash вообще, об этом будет подробно рассказано в главе 11.

Давайте начнем с физических параметров (что, кстати, как раз подходит ко всем этим трюкам с велосипедами).

Различные вращения

Наверное, каждому когда-нибудь приходилось смотреть телевизор вверх ногами. Правда, как следует насладиться новизной не удастся — кровь приливает к голове. Так вот, если вам захотелось вспомнить молодость и посмотреть видео вверх тормашками, вы можете сделать это во Flash (и голова при этом не заболит).

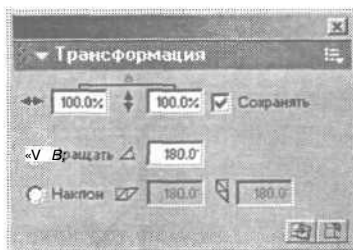
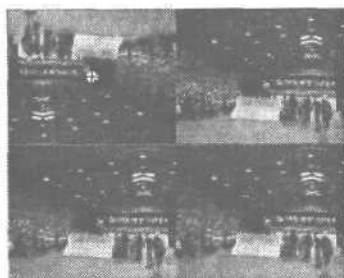
1. Импортируйте видео во Flash или скачайте и откройте файл `properties_base fla`.

- Создайте новый фильм-символ с названием «video» и перетащите копию вашего вставленного видеоклипа из Библиотеки на монтажный стол. Задайте фильму центральную регистрационную точку и отцентрируйте видеоклип в точке (0, 0).
- Вернувшись на основной монтажный стол, перетащите в рабочую область четыре копии вашего фильма-символа. Расположите их рядом, вот так:



Сейчас мы перевернем два из них на 180 градусов, чтобы получить некий таинственный эффект.

- Выберите левый верхний клип и откройте панель Трансформация (Transform) (команда Окно > Трансформация (Window > Transform)). В поле Вращать (Rotate) введите значение «180» и нажмите Enter.



- Повторите то же самое с правым нижним клипом — поворот на 180 градусов:
 - Качество (Quality): здесь нет особой детализации.
 - Интервал ключевого кадра (Keyframe interval): подвижность умеренная.

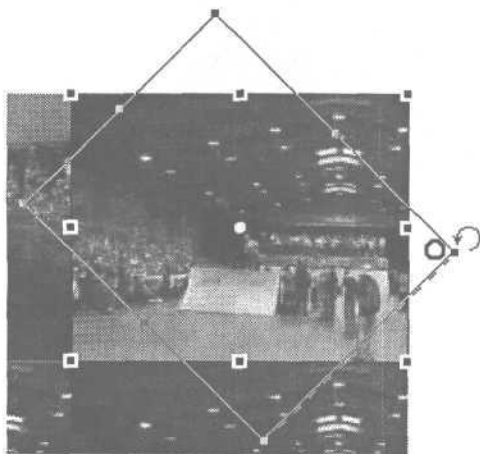
- Масштаб (Scale): оригинал слишком велик для модема 56к.
- Соотношение кадров: мне совсем не обязательно оставлять все кадры.
- Импорт аудио (Import audio): в моем клипе нет звука.

В конечном итоге у меня получился фильм размером 85 килобайт — из 22,2 мегабайтного несжатого DV-PAL фильма QuickTime. Такой результат достигнут, главным образом, благодаря пропуску каждого второго кадра, компрессии, а также уменьшению размера изображения.

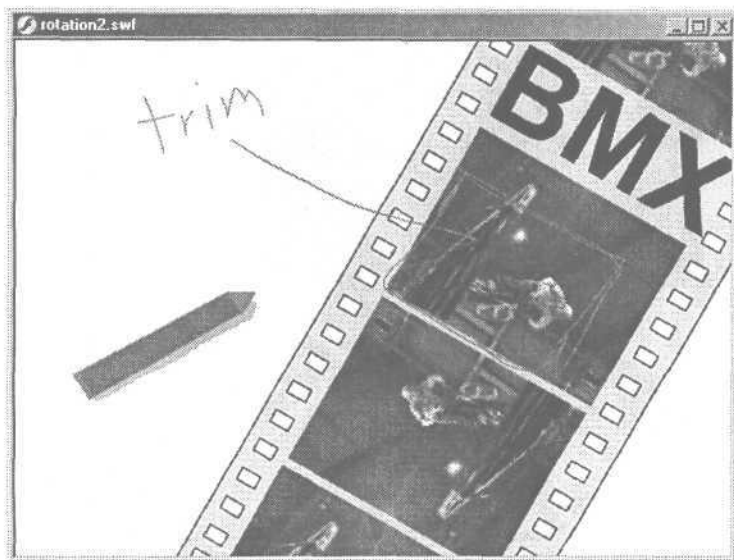
Напоминаю, что вы можете переимпортировать клип или просто посмотреть его параметры, дважды щелкнув на клипе в Библиотеке (Library):

6. Запустите тест фильма и, откинувшись на спинку, понаблюдайте. Мы всего лишь перевернули два из четырех клипов, а впечатление такое, будто все гораздо сложнее.

Чтобы повернуть клип, не обязательно вводить числовые значения, можно вращать вручную, при помощи инструмента Свободная трансформация (Free Transform). Также можно щелкнуть на клипе правой кнопкой и выбрать из контекстного меню команду Вращение и скос (Rotate and Skew). В обеих функциях, если вы будете при вращении удерживать нажатой клавишу SHIFT, угол поворота будет увеличиваться строго на 45 градусов.



Вот еще маленький пример того, как при помощи вращения можно уютно устроить видео материал на веб-странице, здесь я использовал пару видеоклипов, повернутых на 30 и -150 градусов (смотри rotation2 fla).



Скос

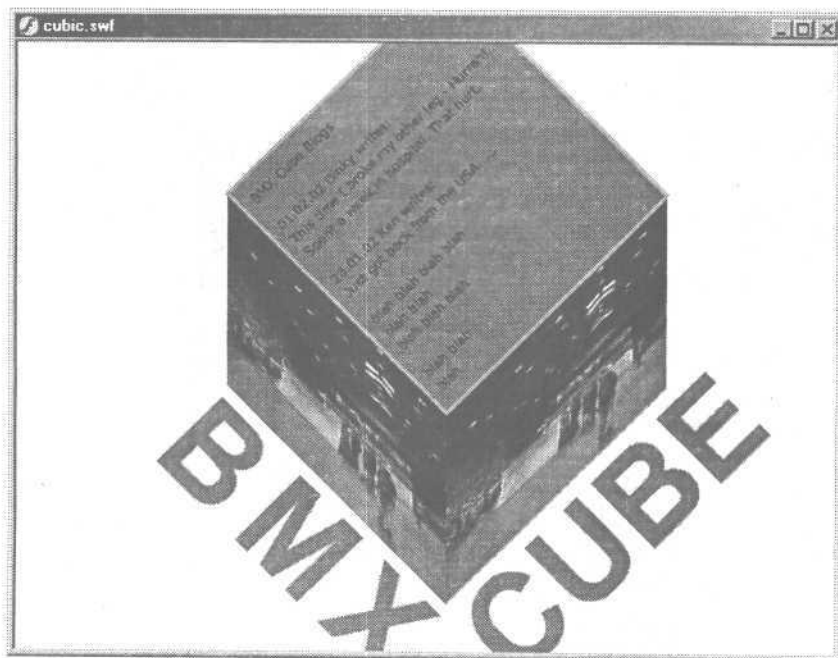
Как и вращение, скос бывает полезен нечасто. Пример такой ситуации — интерфейс типа «галереи», в которой видеоклипы «развешаны на стене». Для создания впечатления перспективы нужно будет не просто повернуть, а перекосить изображение. При скачивании могут возникать довольно причудливые искажения; стоит «поиграть» с этой опцией, посмотреть, что можно из нее выжать.

Кубик

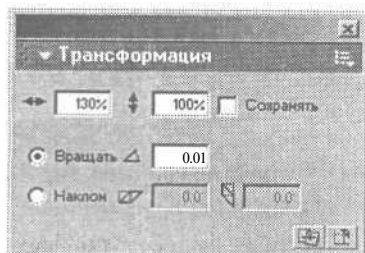
Страница всемирно известного (кхм) сайта BMX Cube (для скачивания — файл cubic fla) выглядит так, как показано вверху следующей страницы.

Как видите, боковые грани куба — это скошенные видеоклипы. Давайте посмотрим, как это делается.

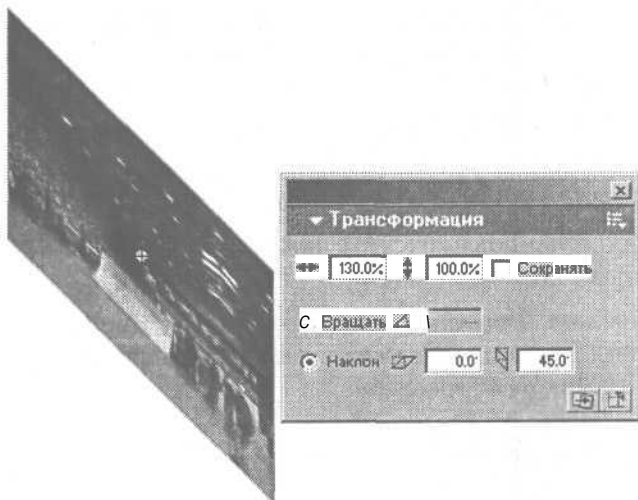
1. Откройте файл properties_base fla или импортируйте собственные видеоклипы в новый фильм.
2. Как всегда, создайте новый фильм-символ и перетащите в него из Библиотеки копию вставленного видео. Отцентрируйте видео в точке фильма-символа (0,0).



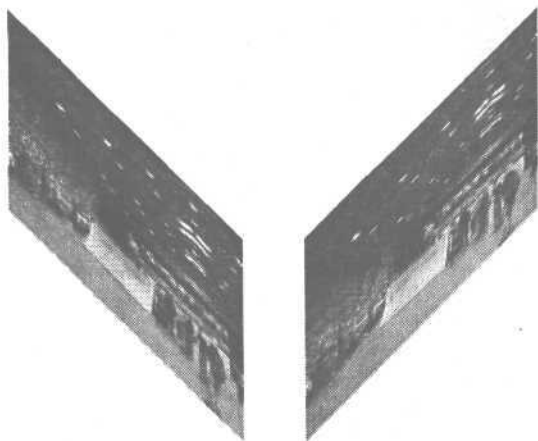
3. На основной монтажный стол перетащите две копии фильма-символа. Чтобы дать больше места для VMX-блогов на верхней грани куба, увеличьте оба клипа по ширине до 130% — при помощи панели Трансформация (Transform), при этом нужно будет снять флажок Сохранять пропорции (Constrain).



4. Выделите левый клип и в той же панели Трансформация установите скос по вертикали на 45 градусов. Для этого сначала нужно включить опцию Наклон (Skew).

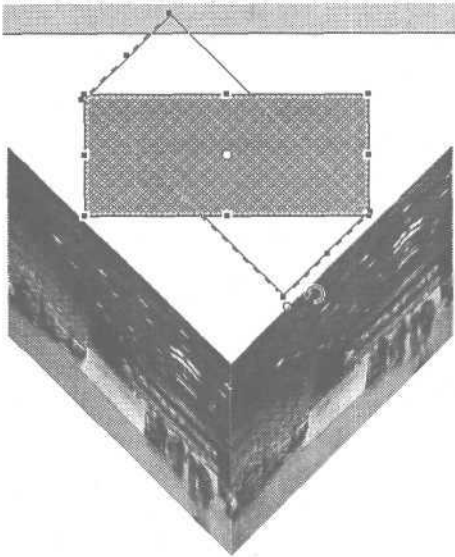


5. Выделив правый клип, установите для него наклон по вертикали на -45 градусов. У вас должно получиться два клипа, скошенных к центру:

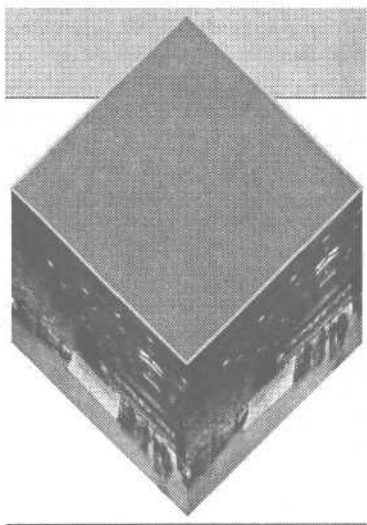


6. С помощью панели Выровнять (Align) подравняйте клипы по нижней кромке. Затем сдвиньте их так, чтобы они соприкасались вертикальными кромками. Двигая один клип к другому, удерживайте нажатой клавишу Shift, чтобы он не смещался по горизонтали. Придвинув их достаточно близко, воспользуйтесь увеличением, а затем — стрелками на клавиатуре для перемещения по пикселям.
7. Сдвинув клипы вместе, выберите оба и сгруппируйте их командой Изменить > Группировать (Modify > Group).

8. Отцентрируйте сгруппированные клипы по горизонтали с помощью панели Выровнять (Align). Теперь боковины нашего куба готовы, нужно изобразить верх.
9. Создайте новый слой под названием «cube» и выберите инструмент Прямоугольник (Rectangle). Подберите подходящий цвет, в инспекторе Параметры (Property) установите толщину штриха 2.
10. Выше сгруппированных клипов нарисуйте прямоугольник. Пока что не заботьтесь о его размере и положении.
11. Выделите прямоугольник и щелкните на инструменте Свободная трансформация (Free Transform). Удерживая нажатой клавишу Shift, поверните прямоугольник на 45 градусов.



12. Не снимая выделения с прямоугольника, клавишами-стрелками выровняйте его по верхним краям видеоклипов. Наконец, с помощью инструмента свободной трансформации растяните прямоугольник, чтобы получился ровный квадрат, формирующий верхнюю грань куба:



Все, что осталось сделать — добавить, если угодно, несколько текстовых элементов и протестировать фильм.

При желании можете заменить «крышку» еще одним видеоклипком (повернутым и растянутым) — я это пробовал, но решил оставить цветной ромб... иногда хорошего все же бывает слишком много!

Зеркальное отражение

ЕСЛИ вы столкнулись с изображением, которое плохо вписывается в общую композицию, делу может помочь обычный переворот, или, вернее, зеркальное отражение картинку по горизонтали (даже если при этом возникнут некоторые несообразности). То же самое и с видео — если плохо вписывается, попробуйте перевернуть. Сложности могут возникнуть лишь в тех случаях, когда видео изображение содержит какие-то важные знаки или ориентиры, либо, наоборот, изъяны, которые делают зеркальное отражение невозможным.

С обратной стороны

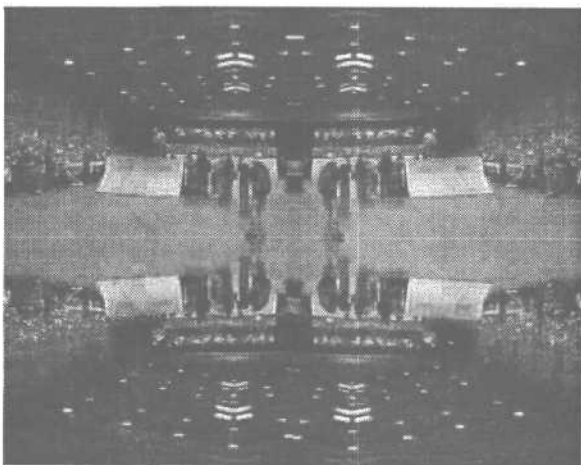
1. Откройте файл `properties_base fla` или импортируйте собственный видеоклип в новый фильм. Как и в предыдущих упражнениях, создайте новый фильм-символ и перетащите в него из Библиотеки копию видео.
2. На основной монтажный стол перетащите четыре копии фильма-символа. Расположите их четким прямоугольником, так, чтобы они соприкасались краями, и между ними не выло заметных промежутков:



3. Поочередно выбирая клипы, примените к ним — с помощью команды Изменить > Трансформация > Отразить по горизонтали (Modify > Transform > Flip Horizontal) или Отразить по вертикали (Flip Vertical) — соответствующее зеркальное отражение, как показано здесь:

Без отражений

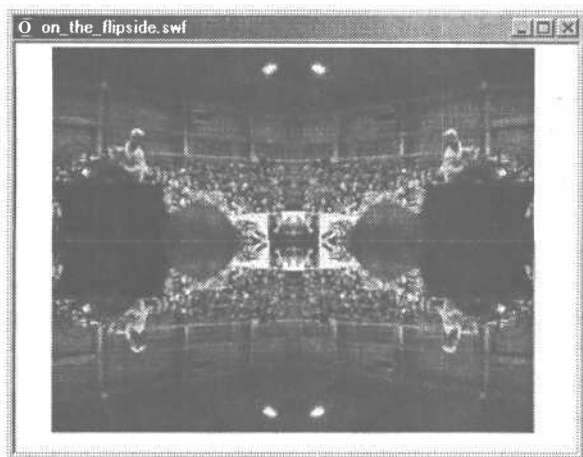
Отражение
по горизонтали



Отражение
по вертикали

Отражение по вертикали
и горизонтали

4. Покончив с этим делом, протестируйте фильм — и совершите путешествие в прошлое, во времена расцвета подобных видеоэффектов, примерно 1972 год.



Комбинации поворотов и отражений могут порой приводить к совершенно потрясающим результатам.

Яркость

Если материал, с которым вы работаете, оказался темным или тусклым, мы рады предложить вам возможность сделать его ярче во Flash. Опция **Brightness** (яркость) позволяет осветлить или затемнить видео. По умолчанию значение равно нулю, его можно увеличить или уменьшить на 100%.

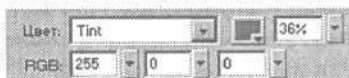
Чтобы изменить параметр яркости фильма или символа, выделите символ в рабочей области, а затем из списка Цвет (Color) в инспекторе Параметры (Property) выберите **Brightness**:



Тонирование

Если предыдущие параметры вам показались **прикольными**, приготовьтесь выпасть в осадок от опции **Tint** (тонирование)! Если как следует поработать с этим параметром, можно сделать из клипа такую цветомузыку!

Свойство **Tint** состоит из двух изменяемых параметров: цвет (значения RGB) и альфа-фактор. Любой цветовой оттенок представляет собой смесь в определенной пропорции красного, зеленого и синего (red, green, blue — RGB). На иллюстрации представлен чистый красный цвет при значении альфа-фактора 36%:



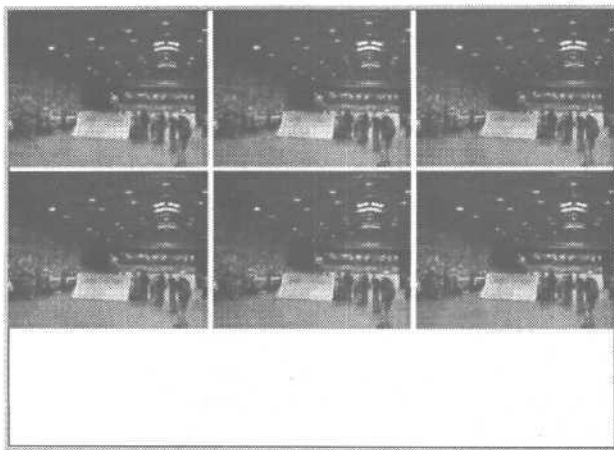
Выбрав нужный цветовой оттенок, вы, изменяя значение альфа, увеличиваете или уменьшаете общую интенсивность цвета. Например, в случае установки значения альфа 100% символ будет залит чистым, непрозрачным цветом.

Опция Tint предназначена отнюдь не только для экспериментов, она позволяет исправить цветовую гамму вашего видео. Предположим, в вашем видео излишне много красного, недостаточно желтого — можно добавить желтого; или, к примеру, видеоклип выдержан в синих тонах, и вы хотите усилить этот эффект, добавив еще синевы. В общем, смысл этого процесса близок видеосъемке через цветные светофильтры — идея в том, чтобы усилить отдельные цветовые оттенки.

Убийственный танцпол

Давайте проведем испытание, выполнив упражнение.

1. Откройте файл `properties_base fla` или, создав новый фильм, импортируйте в него собственный видеоклип. Создайте новый фильм-символ и перетащите в него видео из Библиотеки.
2. Вернувшись к основному монтажному столу, перетащите в рабочую область шесть копий вашего фильма-символа. Расположите их в виде сетки 3x2:



3. Выделив верхний левый клип, выберите Tint из списка Цвет (Color) в инспекторе Параметры (Property).

Мы не станем подбирать вычурные цветовые оттенки, достаточно цветов радуги: красный, оранжевый, желтый, зеленый, синий и фиолетовый (голубой цвет придется вычеркнуть из списка!).

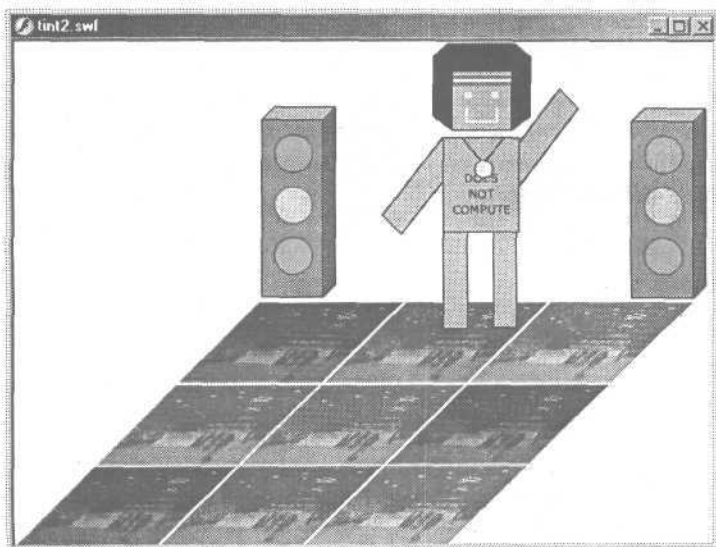
4. Выбрав красный цвет, установите значение альфа-фактора 40%.
5. Двигаясь по часовой стрелке, поочередно выбирайте остальные клипы и применяйте к ним тонирование оранжевым, желтым, зеленым, синим и фиолетовым цветами. Альфа во всех случаях — 40%

Старайтесь при этом выбирать оттенки цветов так, чтобы клип стал ярче, а не бледнее. Мы ведь хотим изобразить радугу, чтобы порадовать народ!

6. Получив в конце концов достаточно привлекательный результат, сохраните фильм под именем `tint1 fla` и протестируйте его. Как видите, установленное нами значение альфа-фактора вполне достаточно — хотя цвет тонирования преобладает, оригинальные цвета также различимы. Впрочем, если вам кажется, что тонирующих цветов чересчур много (или мало), подрегулируйте альфа-фактор самостоятельно.

Если вы установите слишком большое значение, то получится нечто в стиле альбомов «The Smits» («*The Queen is Dead*» — хороший пример); впрочем, возможно, такой эффект вам и нужен.

Что ж, это, конечно, не совсем дискотечный танцпол, который я обещал, но нечто вроде шоу Лайонела Ричи году так в 1986, если смотреть с потолка. Впрочем, при помощи скоса вы можете получить изометрическую проекцию того самого танцпола. Черт побери! Наш Pixel Воу сбежал с соревнований и заявился собственной персоной на видео-дискотеку (скачайте-ка файл `tint2 fla` и полюбуйтеесь):



Альфа-фактор

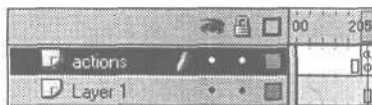
Параметр Alpha — одно из наиболее часто используемых свойств во всех эффектах Flash, от простого перемещения по направляющей до анимаций под управлением ActionScript. Причина проста — этот параметр позволяет создать эффект появления «из ничего» какого-либо объекта на вашем сайте или в фильме. На самом деле, трюк с проявляющимся и увеличивающимся текстом стал уже банальным, но мы можем применить этот фокус к видео и тем самым дать ему новую жизнь.

Как и маскирование, альфа-эффект требует дополнительных вычислительных мощностей, поэтому подумайте как следует, стоит ли использовать его в своем фильме.

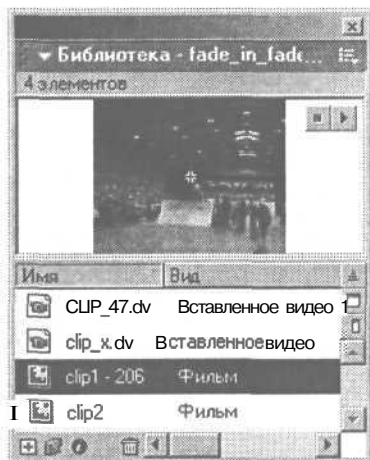
Проявление и растворение

Сейчас мы воспользуемся параметром Alpha для создания во Flash перехода с взаимным растворением. Мы создадим последовательность из двух разных видеоклипов, а затем, манипулируя альфа-фактором каждого клипа, создадим переход между ними. Если захотите сравнить свои результаты с нашими, найдете их в файле `fade_in_fade_out fla`.

1. Скачайте и откройте файл `alpha_base fla` либо подберите два собственных видеоклипа и импортируйте их в новый Flash-фильм.
2. Создайте новый фильм-символ под названием «clip 1». Перетащите в него копию одного из видеоклипов. Получив запрос об увеличении числа кадров, запомните длину клипа, затем нажмите ОК. Отцентрируйте видеоклип в символе:
3. Вставьте новый слой с именем «actions» на монтажный стол клипа-символа. Найдите конец видеоклипа и вставьте над ним, в слое actions, ключевой кадр (F6). Добавьте в этот последний кадр (в нашем примере — кадр 206) действие stop:

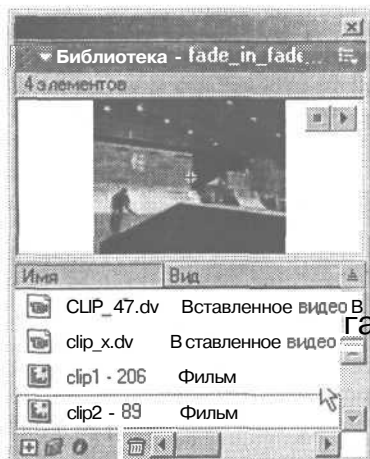


4. Дважды щелкните на символе `clip1` в Библиотеке и измените его имя - добавьте к нему число кадров (продолжительность видео):

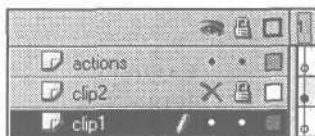


- Повторите три последних шага для второго видеоклипа из Библиотеки, сначала поместив его в символ под названием «clip2», затем добавив действие stop в последний кадр на монтажном столе, и переименовав символ.

Теперь у вас в Библиотеке должно быть два фильма-символа, к именам которых добавлено число кадров в клипе:



- Вернитесь на основной монтажный стол и вставьте два новых слоя. Переименуйте слои, сверху вниз: «actions», «clip2», «clip1»:

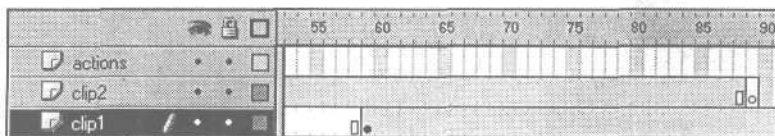


7. Теперь нужно расположить наши клипы в рабочей области. Перетащите копию символа `clip2` в слой `clip2`, копию символа `clip1` в слой `clip1`. Расположите оба по центру сцены.
8. Мы должны увеличить длину монтажного стола, чтобы она соответствовала продолжительности клипов. Так, если вы используете наш исходный файл, в слое `clip1` должно быть 206 кадров, а в слое `clip2` – 89 кадров. Если вы используете свои видеоклипы, эти значения будут другими.
9. Вставьте ключевые кадры (F6) в последний кадр каждого клипа.

Поскольку мой `clip2` гораздо короче другого клипа (89 кадров против 206), я решил поместить его первым в последовательности. Более короткому клипу мы позволим запускаться сразу, а другой клип нужно будет сместить дальше по монтажному столу.

Приняв во внимание длину моих клипов и немного поэкспериментировав, я решил использовать 30-кадровое наложение, что соответствует (при частоте 25 fps) примерно одной секунде.

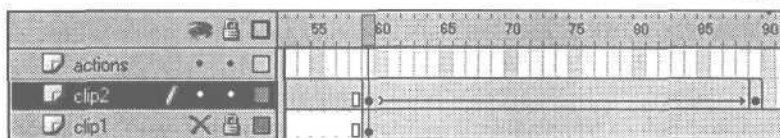
10. Выделите клип, который вы хотите поместить вторым в последовательности (в моем файле — `clip1`), и перетащите его по монтажному столу так, чтобы его начало находилось за 30 кадров до конца другого клипа (в моем файле — в кадре 59).



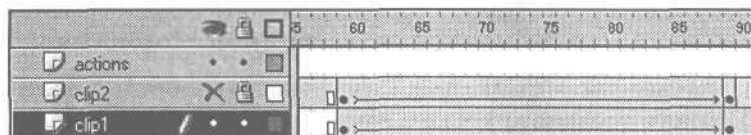
Теперь нужно создать эффекты фединга — «проявления» и «исчезновения». Давайте начнем с исчезновения.

11. Вставьте ключевой кадр (F6) в кадр 59 слоя `clip2` (или того слоя, который содержит первый клип в вашей последовательности).
12. В том же слое выберите ключевой кадр 89, а затем выделите видеоклип в рабочей области. В инспекторе Параметры (Property) выберите Alpha из списка Цвет (Color). Установите значение 0%.

13. Выберите любой кадр между двумя ключевыми и в инспекторе Параметры укажите направляющую движения — Motion из списка Пара (Tween). Ваш монтажный стол должен выглядеть примерно так:



- Если теперь вы протащите указатель воспроизведения по кадрам, содержащим направляющую, вы увидите, как первый клип постепенно растворяется и из-под него становится виден клип из нижнего слоя.
14. Чтобы получить правильный переход, нужно добавить фединг в clip 1. Скройте и заблокируйте слой clip2, чтобы нечаянно не изменить его. Вставьте ключевой кадр в кадр 89 слоя clip 1 (или в кадр, соответствующий окончанию направляющей фединга другого клипа).
15. Выделите видео в ключевом кадре 59 слоя clip1 и установите альфа-фактор на 0%. Теперь выберите любой кадр между 59 и 88 и вставьте направляющую движения. Теперь наш clip1 постепенно проявляется между кадрами 59 и 89:



16. Итак, визуальные элементы фильма закончены. Последнее, что осталось сделать — предотвратить заикливание последовательности. Добавьте ключевой кадр в слой actions, в последний кадр последовательности (в исходном файле — кадр 264). Добавьте в этот кадр действие stop.
17. Теперь протестируйте фильм, и вы увидите аккуратный переход.

И все это благодаря параметру Alpha. Безусловно, эффект альфа-фединга не является чем-то сверх-оригинальным, он используется очень часто, благодаря простоте создания и отличным результатам. Не обязательно всегда комбинировать его с направляющей, как мы делали только что; его можно использовать, например, для создания неяркого видео фона или для наложения двух различных видео объектов.

Расширенные эффекты

Последняя опция в списке Цвет (Color) — это Advanced Effect (расширенные эффекты). Это, по существу, комбинация опций Tint и Alpha, представленная отдельным диалоговым окном, дающим возможность точно настроить значения параметров.

Помимо обычного тонирования и расцветивания, расширенные эффекты позволяют тонко подправить вашу картинку или видео. В следующем упражнении мы исследуем творческие возможности, которые открывают перед нами опции расширенных эффектов.

Повышение резкости видео изображения

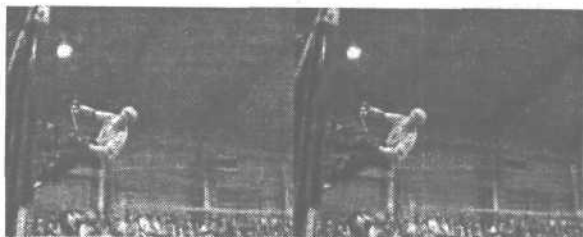
1. Откройте файл `properties_base fla`. Как обычно, поместите видео в фильм-символ, позволив добавить кадры, и отцентрируйте в точке (0,0). На основном монтажном столе переименуйте имеющийся слой в «test» и добавьте новый слой под названием «control».
2. Перетащите две копии фильма-символа `video` из Библиотеки, поместив по одной в каждый слой. Поместите клип, находящийся в слое `control` справа от центра рабочей области, а клип в слое `test` — слева. Тот видеоклип, что справа, мы будем использовать как контрольный, для сравнения с левым, к которому будем применять эффект. Заблокируйте слой `control`, чтобы случайно не изменить его содержимое.

Первое, что мы сделаем — слегка повысим резкость изображения видеоклипа, немного добавив яркости и изменив контрастность.

3. Выделив левый клип, выберите Advanced из списка Цвет (Color) в инспекторе Параметры (Property). Нажмите кнопку Настройки (Settings) и измените значения в диалоговом окне Расширенные эффекты (Advanced Effect) в соответствии с иллюстрацией:



4. Нажмите ОК, чтобы применить настройки. Сохраните фильм под именем `advanced_sharpen fla`, а затем протестируйте. Как видите, изображение стало более четким и чистым, чем в контрольном клипе:



Единственный негативный эффект — то, что тестовый клип требует больше ресурсов из-за повышения яркости, и об этом следует помнить, применяя этот метод.

Закройте тест, а основной фильм оставьте открытым для следующего упражнения.

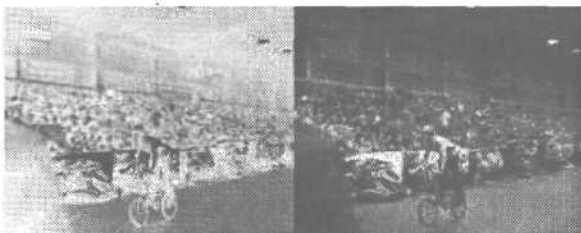
Инверсия цвета

Поскольку цветовые переменные в окне Расширенных эффектов могут принимать и отрицательные значения, мы можем полностью «выкачать» какой-либо цвет из изображения (снизив до 0%), а затем и инвертировать, сделав меньше нуля.

1. Фильм, сохранившийся от предыдущего упражнения, сохраните под именем `advanced_color_inversion1 fla`. Как и в прошлом упражнении, этот файл содержит единственный видеоклип и ничего больше.
2. На основном монтажном столе выделите тестовый клип и примените к нему следующие настройки в диалоговом окне Расширенные эффекты (Advanced effect):



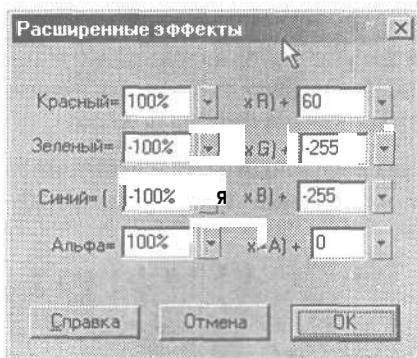
3. Протестируйте фильм, и вы увидите, что черное стало белым, и наоборот:



Напоминает древний аналоговый эффект, применявшийся в мало-бюджетных фильмах, чтобы показать сны или бред сумасшедшего. Такой эффект может пригодиться для того, чтобы подчеркнуть какие-то напряженные моменты, внезапные события и тому подобное.

Что ж, давайте попробуем еще один эффект и на этом закончим. На этот раз мы «вырежем» все цвета, кроме черного и красного.

4. Закройте окно проверки фильма; вернитесь обратно на основной монтажный стол и сохраните фильм как `advanced_color_inversion fla`. Выделите тестовый клип и измените настройки в диалоговом окне Расширенные эффекты (Advanced effect) следующим образом:



5. Протестируйте фильм, и увидите, что он стал одноцветным, красно-черным. Все потому, что мы полностью убрали зеленый и синий. Если в значении параметра зеленого цвета (-255) изменить знак обратно на плюс, вы увидите, что вместо черного цвета появился зеленый — не очень-то приятная картинка! Зато теперь вы увидели, как сильна тварь по имени Расширенные эффекты, и, наверное, поняли, что с ней стоит серьезно поэкспериментировать.

Конечно же, настройки Расширенных эффектов можно изменять плавно, при помощи направляющей, — как и прочие параметры фильма-символа. Можно, например, плавно затемнить фильм до черноты (отключив все, кроме альфы) или создать переход от реальности к психоделическим грезам.

Заключение

Хотя описание всех этих параметров происходило в несерьезной манере, думаю, вы поняли, что возможности они открывают очень серьезные, позволяя очень тонко и аккуратно встроить видео в интерфейс веб-сайта. Например, функции Расширенных эффектов пригодятся для того, чтобы затонировать видео в соответствии с цветами торговой марки компании или цветом общего фона интерфейса. Вращение и скос помогут **надлежащим** образом сориентировать видео в интерфейсе, отличающемся от стандартного, прямоугольно-перпендикулярного.

Главное преимущество манипулирования материалом внутри Flash — в том, что эффект можно немедленно протестировать (не требуется предварительной обработки, рендеринга) и не нужно нести дополнительные издержки, приобретая специальную программу для редактирования (подобную Adobe After Effects). В то же время возможность управлять материалом во Flash, например, цветовой гаммой видео, оставляет возможность импортировать несколько различающихся видеоклипов для специфических случаев.

До сего момента, говоря о видео во Flash, мы уделяли львиную долю времени визуальным аспектам. Пришла пора обратить внимание на другую, не менее важную составляющую — звук...

Звук

Краткое содержание главы:

- *использование программ редактирования звука для манипуляций со звуком вне Flash;*
- *импортирование звуковых файлов во Flash;*
- *различные способы внедрения звука во Flash-фильмы;*
- *настройка параметров звука в интерфейсе Flash.*

До сих пор мы игнорировали все, что связано со звуковым сопровождением видео, чтобы сосредоточиться на том, как работает видео и не увязнуть в болоте звуковых проблем. Следующие две главы мы посвятим рассказу о том, как работать со звуком в Macromedia Flash MX, как комбинировать звук и видео.

В данной главе мы как раз и обратимся к звуку, выясним, в каких случаях лучше оставить видео файл со звуком, а в каких лучше импортировать звуковой файл отдельно.

В этой книге мы не уставали повторять, что качество *исходного* видео-файла, предназначенного для переноса во Flash, имеет важнейшее, решающее значение для того, насколько хорош будет результат. То же самое касается и работы со звуком.

Для начала давайте поговорим о том, как подготовить звуковые файлы к импорту во Flash, пользуясь сторонним программным обеспечением для **редактирования звукозаписей**.

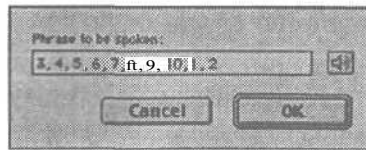
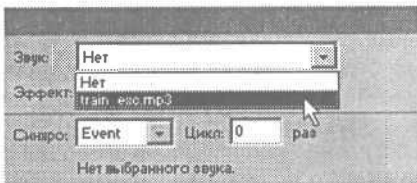
Звуковые редакторы

Звуковые редакторы, предназначенные для работы с дискретным цифровым аудио, позволяют вырезать и вставлять фрагменты звукозаписей, применять к ним всевозможные фильтры и эффекты. Они придут вам на помощь, если нужно обрезать звукозапись, усилить ее, устранить посторонние шумы.

Рассмотрим две программы для редактирования звука: Sonic Foundry Sound Forge 5.0 для Windows, а для пользователей Mac — Sound Studio 2.0 от Felt Tip Software. Наше описание не является всеобъемлющим, поскольку каждая из этих программ достойна отдельной книги! Мы сосредоточимся на основных навыках, которые понадобятся вам, чтобы подготовить звуковые файлы для Flash.

Для следующего упражнения пользователям PC придется скачать довольно много (23,1 мегабайта пробной версии Sound Forge 5), пользователям Mac — гораздо меньше (Sound Studio 2 — всего лишь 1,2 MB). Если вам уже приходилось пользоваться какими-либо звуковыми редакторами, и вы чувствуете себя в этом вопросе уверенно, можете сэкономить время и сразу перейти к разделу, озаглавленному «Звук во Flash». Если же хотите получить сведения и об этих приложениях, читайте дальше!

Если вас это интересует, звуковой файл, используемый в данном упражнении — это запись старого доброго голоса «Макинтоша» из панели управления Speech:



Голос был записан при помощи Snapz Pro (от Ambrosia Software), при включенной опции Record MacAudio. Сохранен с CD-качеством (44 кГц).
Прежде всего — об очистке и фильтрации звука.

Фильтрация звука

Обе программы, о которых пойдет речь, содержат множество инструментов для исправления недостатков звукозаписи, но сотворить чудо они не могут! Если вы попытаетесь записать интервью с человеком, стоящим посреди автострады, немного вы сможете сохранить из его драгоценных слов — даже в цифровой записи. К сожалению, «Photoshop для звука» пока не существует, но все же звуковые приложения прошли длинный путь, стараясь искоренить всевозможные посторонние хлопки, гудение и свист. Вот наиболее распространенные инструменты, которые помогут вам в этом деле.

- Нормализация применяется для повышения общего уровня звука в аудио файле. Принцип работы таков: выделенная порция аудиозаписи сканируется для определения наивысшего уровня, затем в соот-

ветствии с полученным результатом повышается уровень всего файла. Полагаю, вы будете часто применять нормализацию к самостоятельно записанным вами звуковым файлам.

- **Коррекция** — это то, что делает знакомый всем эквалайзер: изменение уровня звука определенной частоты. Пользователь по своему выбору может повысить или понизить уровень отдельных частот. Например, может понадобиться убрать лишние басы (снизить низкие частоты), или устранить резкое, металлическое звучание. Кроме того, при помощи эквалайзера можно выделить и устранить звуковой фон, нередко возникающий в процессе записи.
- **Компрессия** применяется для снижения громкости отдельных звуков, превышающей заданный уровень, так что общий уровень громкости становится более «гладким», без резких скачков. Будет полезно, например, в ситуации: вы ведете запись при помощи неподвижного микрофона, при этом говорящий то приближается к микрофону, то удаляется (или говорит то громче, то тише). Компрессия приведет к выравниванию общего уровня громкости.

А теперь приступим к работе с аудио редактором Sound Forge 5 для Windows. Пользователи Mac могут перелистать несколько страниц — до следующего раздела, озаглавленного «Sound Studio 2 для Macintosh».

Sound Forge для Windows

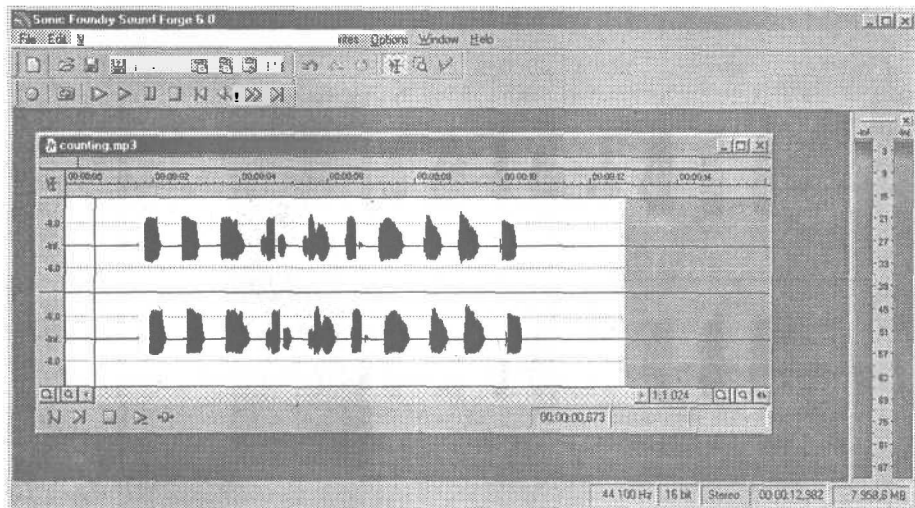
Sonic Foundry's Sound Forge — пожалуй, наиболее популярная программа для редактирования звука под Windows. Это практически всеобъемлющее приложение, возможности которого удовлетворят любого любителя звукотехники. Помимо полной версии, существует более дешевая урезанная версия Studio — всего за \$60. Ну, ну, не впадайте в панику — вы можете взять бесплатную пробную версию (без возможности сохранения файлов) с www.soundforge.com/download, чтобы попользоваться ей в этом упражнении и решить, устраивает ли она вас!

Кроме этого, мы будем использовать исходный файл `counting.mp3` — его можно скачать на сайте www.friendsofed.com.

Работаем в «Кузнице звука»

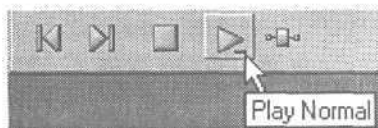
1. Откройте свою версию Sound Forge и выполните команду `File > Open`. Выберите исходный файл `counting.mp3`.

Вы увидите следующее:



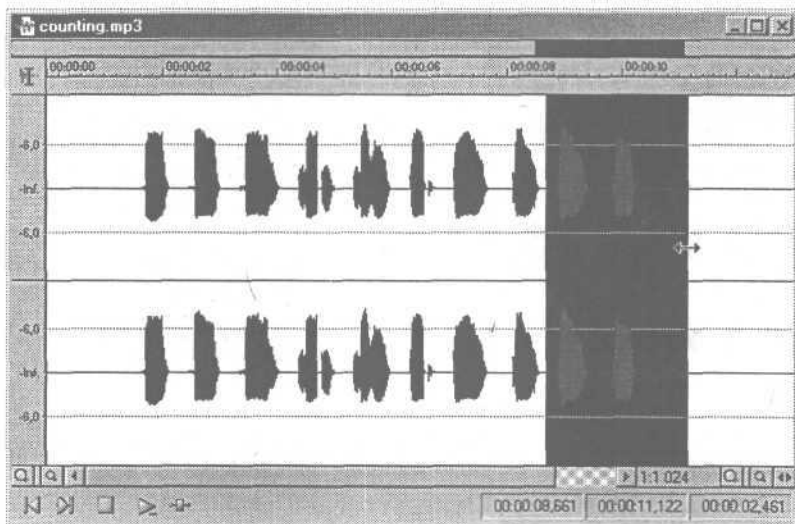
Два слоя представляют два аудио канала (левый и правый), темные фигуры — звуковые волны: визуальное представление звука. Каждая отдельная группа представляет произносимое число, от 1 до 10, четкие промежутки между ними позволяют легко различить их визуально.

2. Нажмите кнопку Play Normal, расположенную в левом нижнем углу активного окна. Ваш файл будет воспроизведен один раз от начала до конца и остановлен:

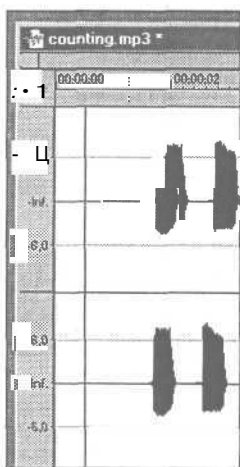


При прослушивании вы заметите, что отсчет несколько странный — начинается с 3, доходит до 10, и заканчивается на 2! Числа были перепутаны сознательно, чтобы мы смогли попрактиковаться и переставить звуки в правильном порядке. Итак, первое, что нам предстоит сделать — сдвинуть числа 1 и 2 в начало файла.

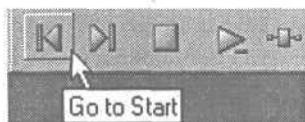
3. Щелкните слева от второй с конца волновой формы и тащите мышью вправо, выделяя два последних числа (две последние волновые формы), захватив немного свободного пространства в конце:



4. Теперь вырежьте эти числа командой **Edit > Cut**. Они исчезнут с временной диаграммы и будут помещены в буфер обмена (так же, как это происходит с текстом и изображениями).
5. Установите курсор временной диаграммы перед первым числом и вставьте вырезанный участок звука при помощи **Edit > Paste**.

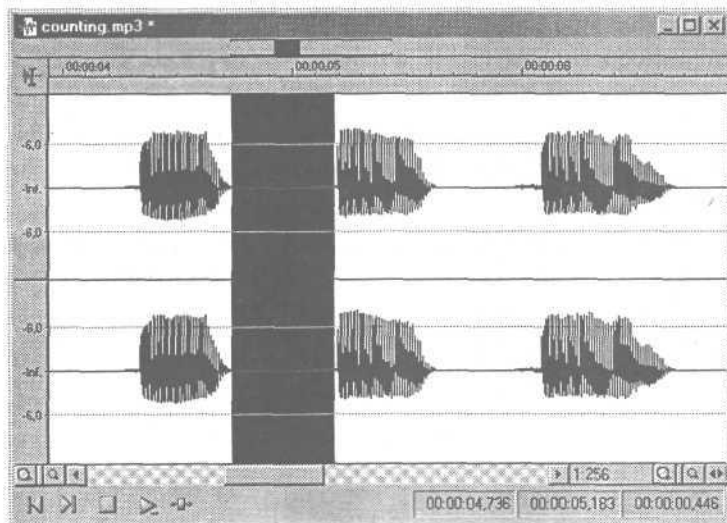


6. Хорошо, теперь нажмите кнопку перемотки к началу — **Go to Start**, и снова воспроизведите последовательность.



На этот раз наши числа произносятся в правильном порядке, однако между числами 2 и 3 слишком длинная пауза — это досадно. Давайте уберем ее и сделаем последовательность более равномерной. Первым делом, нам нужно измерить продолжительность типичной паузы, чтобы знать, какое расстояние оставлять между 2 и 3.

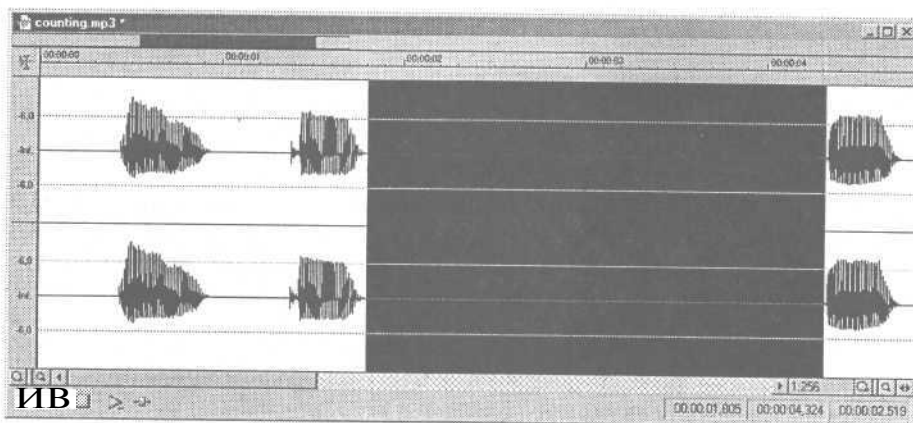
7. Выделите паузу между числами 3 и 4 (это будет проще сделать при небольшом увеличении) и взгляните на информационные поля в правом нижнем углу активного окна:



Два первых поля показывают время начала и конца нашего выделения, а третье — его продолжительность, — отлично, то, что нам нужно!

Как видим, между словами у нас приблизительно по 0,4 секунды, это значение мы и используем. Однако прежде надо вырезать теперьшнюю длинную паузу между числами 2 и 3.

8. Выделите тишину (пустое пространство) между 2 и 3:

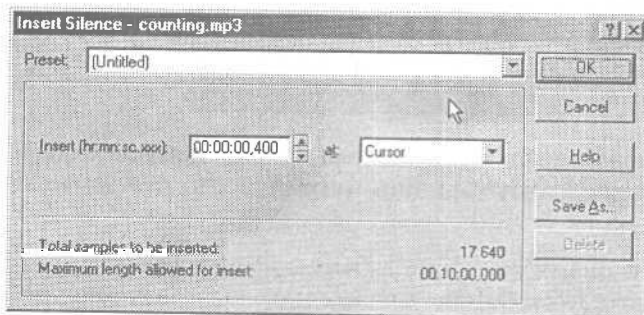


Если бы это был реальный проект, вам пришлось бы действовать *гораздо* внимательнее и использовать большее увеличение, чтобы выполнить эту операцию как можно аккуратнее, но на этот раз сойдет и грубое приближение.

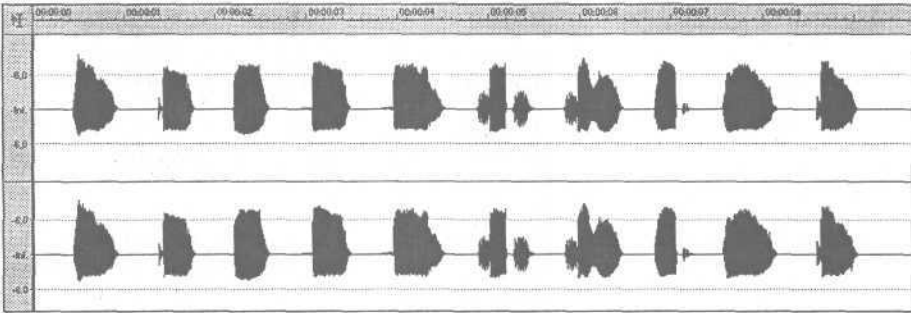
9. Установив выделение, вырежьте тишину с помощью Edit > Cut.

Теперь наши числа 2 и 3 оказались совсем рядом. Слишком близко! Теперь мы вставим между ними паузу — такую, как мы измерили. К счастью, курсор как раз находится в том месте, где мы только что резали, так что мы можем вставить сюда отрезок тишины.

10. Идем в меню Process > Insert Silence..., и в появившемся окне, в поле Insert устанавливаем продолжительность тишины «0.4»:

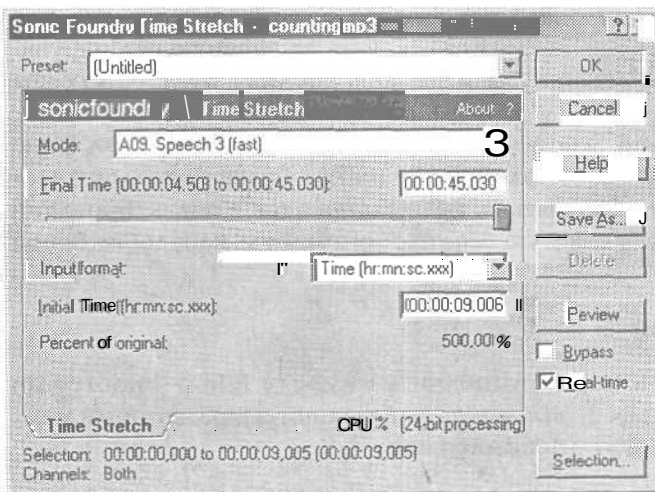


11. Теперь нажмите кнопку ОК. Вернувшись в основное окно, мы увидим, что теперь наша последовательность в полном порядке — отлично! Воспроизведите файл еще раз, послушайте, как он звучит.



Порядок, наша звукозапись готова. А теперь давайте немного повеселимся — используем один из сумасшедших фильтров Sound Forge, **Time Stretch** (растяжение времени). Наверняка вы его уже слышали — он применяется в *любой* электронной музыке. Практически всегда.

12. Первым делом, удалите все паузы от начала до конца, выделяя их и применяя команду Edit > Delete (Clear). Теперь начнем веселиться...
13. Выделите все звуки (Edit > Select All) и воспользуйтесь командой меню Process > Time Stretch.... Вы увидите еще одно унылое серое окно, но вы еще не знаете его волшебных способностей!
14. Выберите A09.Speech из списка Mode (это список предустановленных режимов, которые можно изменять) и тащите движок Final Time вправо. До предела! Ваши настройки должны выглядеть примерно так:



15. Нажмите кнопку Preview, чтобы тут же прослушать полученный эффект. Если он вам понравился, нажмите кнопку ОК, если нет — поэкспериментируйте еще с этим и другими фильтрами и эффектами.

В Sound Forge имеется несколько отличных эффектов и клише, которые так и напрашиваются, чтобы их применили. Но, конечно же, все зависит от того, как с ними обращаться.

Я настоятельно рекомендую вам «поиграть» со всем этим, посмотреть, что как работает. Кое-кто из вас может обнаружить у себя скрытый талант к музыке, ну, а остальные, по крайней мере, порадуются интересным эффектам!

Помимо звуковых эффектов, в Sound Forge имеется еще одна очень полезная вещь — возможность открывать AVI-файлы и визуально редактировать звук в них. Это очень пригодится вам при озвучивании ваших видео-файлов, ведь программы редактирования видео не способны на такую гибкость при работе со звуком.

Ну, вот мы и закончили наше сверхкраткое введение в редактирование звука с Sound Forge. Если хотите, можете пропустить приложение для Mac и перейти сразу к разделу, озаглавленному «Звук во Flash».

Sound Studio 2 для Macintosh

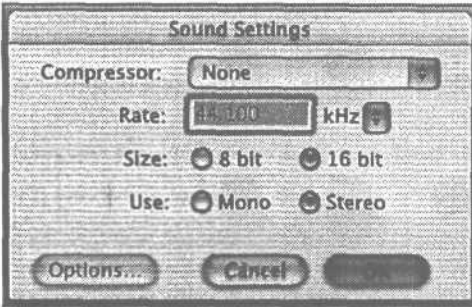
Sound Studio — условно-бесплатное приложение для редактирования звукозаписей, созданное компанией Felt Tip Software для платформы Mac. Версия 2 оптимизирована под Mac OS X, однако будет работать под Mac OS от 8.1 и выше. Вы можете скачать 14-дневную полнофункциональную пробную версию для этого упражнения с www.felttip.com/ss (1,2 мегабайта).

Поработав с Sound Studio, лично я остался верен приложению Macromedia SoundEdit. К сожалению, SoundEdit более не поддерживается, и вряд ли будет обновляться. Если вы новичок в редактировании звукозаписей на Mac, я советую начать с чего-нибудь наподобие Sound Studio.

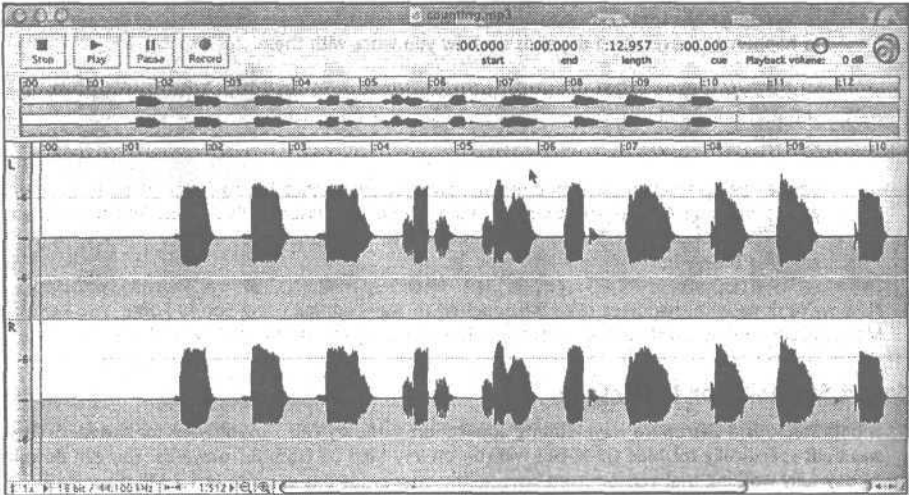
Работаем в «Студии»

1. Откройте Sound Studio и выполните команду File > Import with QuickTime. Мы будем использовать файл [counting.mp3](http://www.friendsofed.com), который можно скачать с сайта www.friendsofed.com.

В окне запроса я оставьте предложенные настройки:



2. Нажмите кнопку ОК. Вас приветствует основной интерфейс:

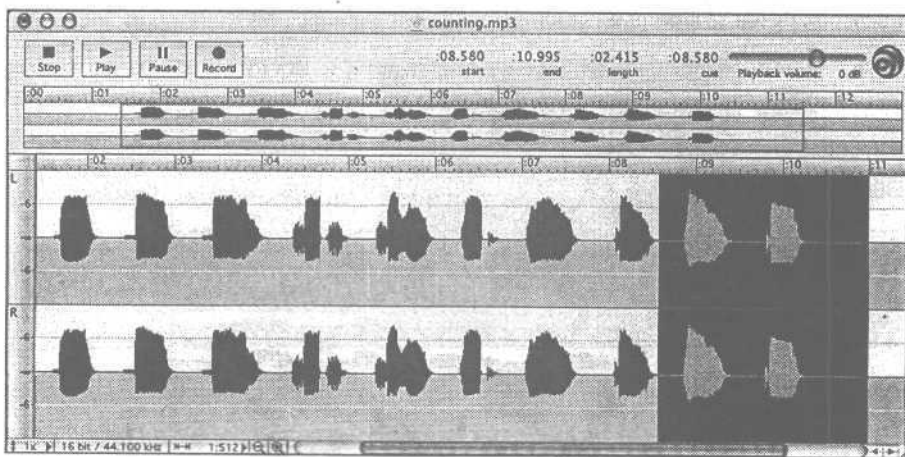


Два слоя представляют два аудио канала (левый и правый), темные фигуры — звуковые волны: визуальное представление звука. Каждая отдельная группа представляет произносимое число, от 1 до 10, четкие промежутки между ними позволяют легко различить их визуально.

3. Нажмите кнопку Play, чтобы прослушать импортированный файл.

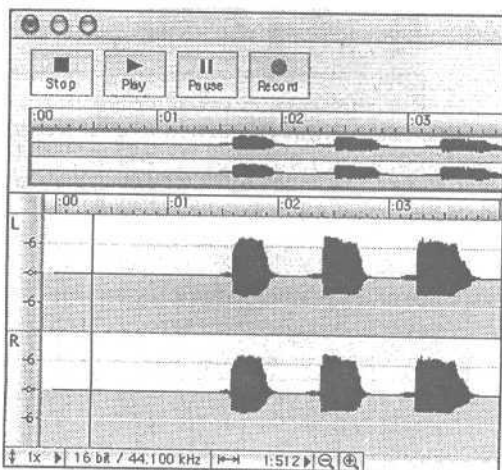
М-да, вы правы — либо у Фреда съехала крыша, либо он считать не умеет! Совершенно идиотская последовательность чисел — от 3 до 10, а потом 1 и 2. На самом деле, числа были перепутаны сознательно, чтобы мы смогли попрактиковаться и переставить звуки в правильном порядке.

- Выделите два последних числа (две последние волновые формы), щелкнув слева от второй с конца волновой формы и протаскив мышью вправо. Вырежьте эту порцию звука клавишами **APPLE + X** (или командой **Edit > Cut**).



Мы собираемся переставить 1 и 2 на правильную позицию, перед всеми остальными числами. Не бойтесь захватить слишком много тишины вместе с числами, чуть позже мы все обрежем поаккуратнее.

- Щелкните левее всех чисел — на временной диаграмме появится мигающий курсор:



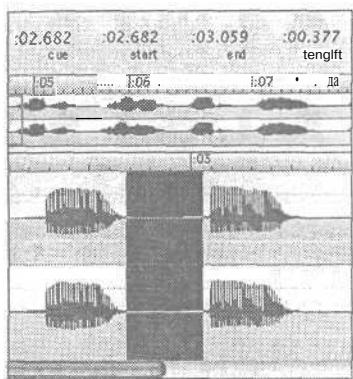
Сюда мы вставим наши 1 и 2.

6. Нажмите клавиши Apple + V (или воспользуйтесь командой Edit > Paste), чтобы вставить вырезанные числа. Они будут помещены в начале, где им и следует быть — по крайней мере, меня учили считать именно так!
7. Нажмите Play и прослушайте числа в правильной последовательности.
8. Однако меду числами 2 и 3 слишком длинная пауза, давайте попробуем ее исправить и сделаем последовательность более равномерной.

Два первых поля показывают время начала и конца нашего выделения, а третье — его продолжительность, — отлично, то, что нам нужно!

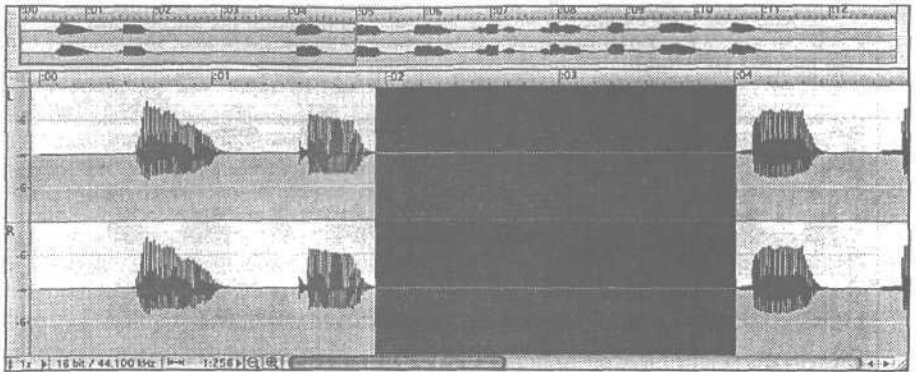
Как видим, между словами у нас приблизительно по 0,4 секунды, это значение мы и используем. Однако прежде надо вырезать теперешнюю длинную паузу между числами 2 и 3.

9. Выделите промежуток между числами 3 и 4 (это будет проще сделать при небольшом увеличении) и взгляните на информацию о выделении в верхней части окна:

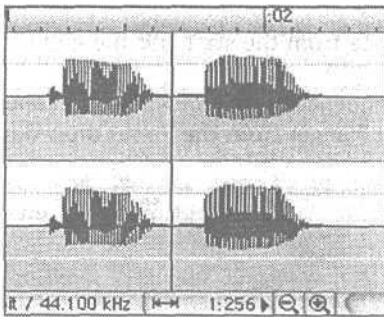


Четвертое информационное поле, length, показывает продолжительность (в секундах) нашего выделения — паузу между произносимыми числами. В среднем все паузы в нашей последовательности продолжаются примерно 0,4 секунды, такой промежуток мы и сделаем между 2 и 3. Однако сначала нужно устранить существующий промежуток.

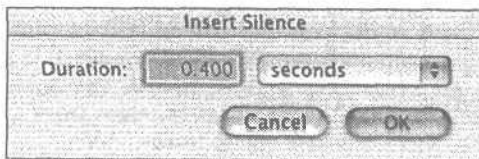
10. Выделив всю тишину между числами 2 и 3, вырежьте ее клавишей Backspace:



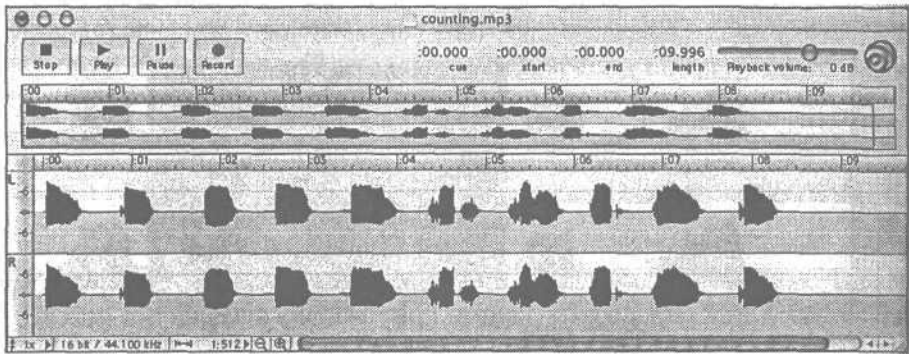
Теперь наши числа 2 и 3 стали близкими друзьями! Печально, но придется прекратить их маленькую вечеринку еще до того, как она начнется. Заметьте, мигающий курсор как раз находится в том самом месте, где мы только что вырезали тишину, и это отличная позиция для вставки нового, правильного, отрезка тишины:



11. Выполнив команду Insert > Silence, в поле Duration появившегося окна вводим «0.4» секунды:



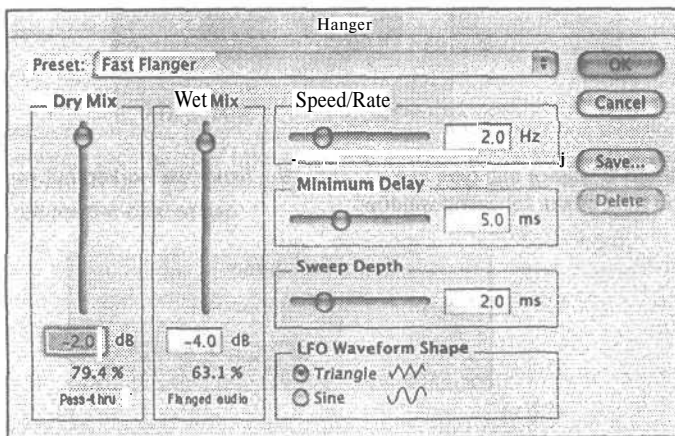
12. Теперь, нажав кнопку ОК и установив такой масштаб, чтобы видеть всю последовательность, мы увидим, что все числа расположены равномерно:



13. Нажмите кнопку Play, чтобы прослушать реорганизованный файл — видите, Фред снова научился считать.

В качестве завершающего штриха мы придадим его голосу особую окраску с помощью эффекта Flange.

14. Удалите излишнюю тишину в начале и в конце файла.
 15. Выделите всю последовательность с помощью Edit > Select All и выполните команду Filter > Flanger.... В появившемся окне из списка Preset выберите Fast Flanger и нажмите кнопку ОК:



По окончании недолгой обработки нажмите Play и прослушайте эффект. Ну как, впечатляет новый голос Фреда?

На это наше краткое введение в Sound Studio завершено. Как видите, программа умеет гораздо больше, чем мы рассказали, и мы предлагаем вам исследовать ее возможности самостоятельно. Здесь вы можете задержаться и поиграть с различными звуковыми эффектами, а можете сразу взяться за следующий раздел, в котором мы начнем работать со звуком во Flash.

Звук во Flash

Существует несколько методов работы со звуком во Flash, и реальность такова, что звук — вещь более гибкая в работе, чем видео. Эта гибкость достигается при управлении звуком при помощи ActionScript (наподобие того, как мы управляли видео в главе 4). Мы рассмотрим подробно способы манипуляции звуком при посредстве ActionScript в следующей главе.

Прежде, чем обратиться к проблемам звука в соединении с видео, давайте сделаем краткий обзор различных случаев использования звука.

Музыка

Музыка часто используется на Flash-страницах для усиления общего впечатления. Однако здесь нужен осторожный подход, фоновая музыка может быстро надоест и стать докучной, особенно если небольшой музыкальный фрагмент повторяется вновь и вновь. В лучшем случае пользователь выключит звук, а может и просто уйти с вашего сайта.

Музыка не только придает дополнительную окраску вашему сайту; стиль музыки может многое сказать как о самом сайте, так и о том, на какую аудиторию он ориентирован. Само собой, это дело вкуса, выбирать вам. Просто представьте, что некто, приодевшись в свой лучший бархатный костюмчик в стиле неоромантизма, нечаянно входит в хранилище записей ценителя кондового панк-рока — ошарашенный ударной волной трех аккордов, он тут же захлопнет дверь!

Прежде чем использовать на своем сайте фоновую музыку, вы должны четко представить себе, для чего вы это делаете. Задайте себе несколько вопросов.

- Какое настроение должна создавать музыка и насколько она соответствует содержанию сайта?
- Для загрузки музыки понадобится дополнительное время, стоит ли она того?

- Не ограничит ли выбранная музыка вашу аудиторию? Хотите ли вы этого?

Если подразумевается, что музыка будет аккомпанементом к видео, будет ли она полезна? Кинематограф доказал, что музыка значительно усиливает выразительность и эмоциональное наполнение визуального материала. Если из фильма убрать музыку, все его изъяны немедленно «вылезут» наружу (я сам убедился в этом в университете, когда преподаватели заставляли нас смотреть знаменитые хичкоковские сцены без звука!). Хотя в моем ВМХ-проекте нет какой-либо повествовательной структуры, и комментариев диктора в общем-то не нужен, без подходящей музыки клипы кажутся незавершенными, в них как бы не хватает жизни, энергетики.

Привнести музыку во Flash-фильм можно двумя способами: либо импортировать отдельный звуковой файл (например, MP3, WAV или AIFF), либо использовать звук, присоединенный к видеоклипу. Какой же способ лучше?

Если музыка синхронизирована с видео-содержанием или «главную роль» в клипе играет вокал, то лучше импортировать музыку вместе с видеоклипом. Тогда не придется мучаться, пытаясь синхронизировать видео и звук во Flash — чудовищно сложное дело! Если же музыка является лишь дополнением к видео и не требует точной синхронизации (как в случае с моим ВМХ материалом), стоит подумать об использовании отдельно импортированного звукового сопровождения. Это позволит управлять звуком независимо, потенциально пользователь может получить возможность изменять громкость, заглушить звук или пропустить очередной звуковой трек.

Если идея использования двух отдельных файлов вас не вдохновила, и вы считаете, что это — излишние сложности, вот вам еще одно важное преимущество такого метода: выигрыш в производительности. На не самых быстрых машинах, чтобы сохранить синхронизацию видео со звуком, будут пропускаться кадры. Если же аудио и видео находятся в отдельных файлах, видео будет воспроизводиться куда более гладко, ведь оно ни к чему не привязано.

Вспомним опять о музыке в кино — если полностью отбросить музыку, в фильме останутся только диалоги и звуковые эффекты, сопровождающие происходящие события и окружающую обстановку. При монтаже кинофильмов так и поступают, чтобы убедиться, что эти звуковые элементы находятся точно там, где следует. Если такая необходимость возникает во Flash, очень важно сохранить синхронизацию видео и аудио. Это тот случай, когда звук, импортированный вместе с видео, предпочтительнее отдельного звукового файла, несмотря на потери в производительности.

Бывают и другие ситуации, когда синхронизация звука с видео может нарушиться, главным образом при манипуляциях с видео посредством **ActionScript**. Например, вы создали интерактивный фильм, в котором пользователь сам выбирает, в какое место видеозаписи нужно переключиться. Имеется один видеоклип, содержащий все возможные «пункты назначения», для перемещения в них по запросу пользователя используется **ActionScript**. При таком сценарии, когда вы «перебрасываете» воспроизведение видео к определенному кадру, звук зачастую оставляет «хвост», и ваша драгоценная синхронизация нарушается. Способ преодоления этого очевиден — следует использовать несколько видеоклипов, вставленных в отдельные фильмы-символы, каждый из которых и будет представлять какое-то «место назначения». Из-за этого увеличится размер файла, зато сохранится синхронизация на всех отрезках.

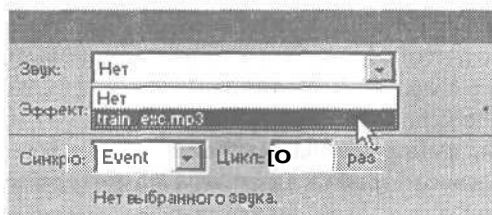
Работа без скриптов

Помимо программного управления при помощи скриптов, можно работать со звуком во Flash и вручную. Одно из преимуществ такого подхода — то, что вы получаете возможность потоковой передачи аудио. Давайте рассмотрим звуковые возможности Flash, выполнив небольшое упражнение.

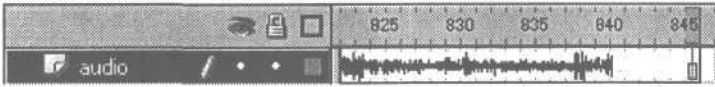
1. Откройте новый фильм Flash и переименуйте имеющийся слой в «audio».
2. Выполните команду **Файл > Импорт (File > Import)** и найдите достаточно длинный звуковой файл на своей машине либо воспользуйтесь файлом `train_exc.mp3` — он имеется среди файлов, иллюстрирующих данную главу, на www.friendsofed.com.
3. Нажав кнопку **Открыть (Open)**, импортируйте звуковой файл во Flash.

Помните, во Flash можно импортировать файлы нескольких форматов, включая стандартные MP3, WAV и AIFF.

4. Выделите на монтажном столе кадр 1, и в инспекторе **Параметры (Property)** появится опция **Звук (Sound)**. Выберите из этого списка свой звуковой файл:

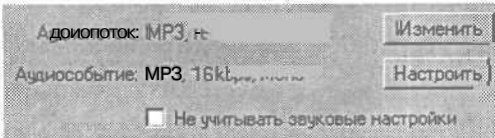


- В списке содержатся все звуковые файлы, присутствующие в Библиотеке (Library). Прочие настройки оставьте, как есть, только убедитесь, что в опции Синхро (Sync) выставлено Event. Скоро вы поймете, для чего.
- Сняв выделение с кадра 1, вы увидите в нем синий значок-тильду — это наш звуковой файл. Растяните монтажный стол с помощью F5, и вы увидите весь файл в виде волновой формы:



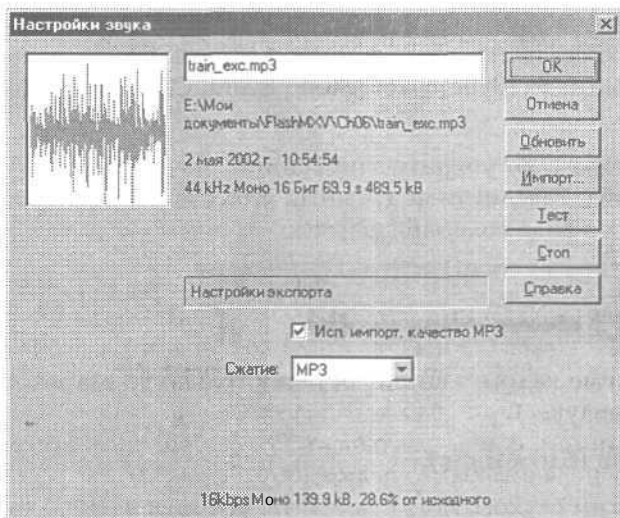
- Удалите лишние пустые кадры в конце, оставив только те кадры, в которых содержится звук.
- Протестируйте фильм (Ctrl + Enter).

Те, кто обладает тонким слухом, могут заметить, что звук несколько отличается от исходного. Это из-за того, что Flash применяет к любому звуку некое стандартное сжатие, если вы не укажете **ему** что-либо иное. Эти общие установки находятся в нижней части на вкладке Flash окна Настройки опубликования (Publish Settings) (команда меню Файл > Общие параметры (File > Publish Settings)):



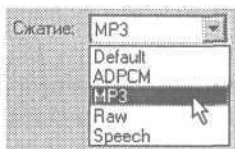
Если вы недовольны сжатием, которое Flash применил к вашей прекрасной музыке, можно выставить индивидуальный вид компрессии для каждого звукового файла. Давайте посмотрим...

- Откройте Библиотеку (Library) (F11) и двойным щелчком на значке вашего звукового файла откройте окно Настройки звука (Sound Properties):



В верхней части окна — информация об оригинальном файле, внизу — настройки экспорта, и целый набор кнопок справа. Впрочем, в данный момент сведений об экспорте информации почти нет — это потому, что используются стандартные настройки, указанные в окне Настройки опубликования (Publish Settings).

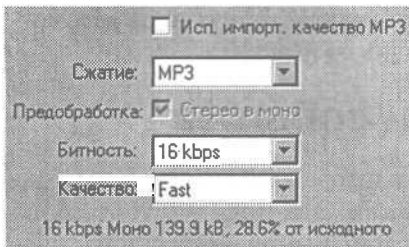
10. Раскройте список Сжатие (Compression), и вы увидите несколько типов компрессии:



Вкратце расскажем о двух наиболее значимых кодеках из списка:

- MP3: кодек, стремительным натиском покоривший интернет — так сильно он позволяет сжимать файлы. Наилучший выбор для музыки;
- Speech: этот кодек лучше всего подходит для речевых звукозаписей. Частотный диапазон человеческой речи ограничен, что позволяет ограничить максимальную частоту 5 кГц. Этот кодек полезен, например, для обучающих сообщений, комментариев.

11. Выберите кодек, наиболее подходящий для вашего аудио файла. У меня музыка, поэтому я выбираю MP3:



Теперь появилось несколько опций, показывающих настройки экспорта (чтобы их увидеть, вы должны снять флажок **Исп. импорт. качество MP3** (Use imported MP3 quality)), а также полезную информацию о размере экспортированного файла. Как видите, в этом примере размер экспортированного файла составляет 28,6% оригинального, это довольно хорошо, но все-таки это целых 139 КВ.

Давайте попробуем изменить настройки, и посмотрим, насколько меньше мы сможем его сделать... а заодно услышим разницу!

12. Раскройте список **Битность (Bit Rate)** и выберите значение 8 kbps. Сразу видно, как уменьшился размер итогового файла, однако хорошо бы сначала прослушать файл с такими настройками.
13. Нажмите кнопку **Тест (Test)**, чтобы прослушать файл с новыми настройками. Появится индикатор выполнения, и вам придется немного подождать, пока Flash перекодирует файл в формат MP3 с настройками, которые мы задали.

Как только кодирование завершится, мы услышим звук. Внимательно вслушивайтесь, чтобы уловить всякие бульканья и шуршания.

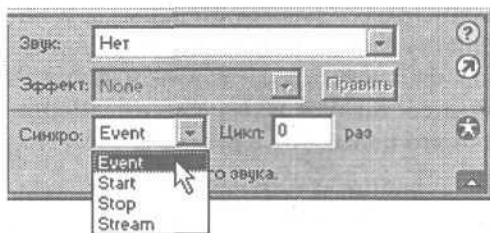
Чтобы добиться столь малого размера файлов, Flash конвертирует оригинальные стерео файлы в моно, а также снижает качество звука. Как вы слышали, музыка при этом получается довольно убогой. Как правило, при сжатии музыки не стоит устанавливать частоту дискретизации (bit rate) ниже 20 kbps, чтобы сохранить приемлемое качество; в то же время более низкие значения прекрасно подойдут для звуковых эффектов, таких, как щелчки кнопок и тому подобное.

14. Если качество вас устраивает, нажмите кнопку **Обновить (Update)**, а затем кнопку **ОК**. Если же оно для вас неприемлемо, повысьте частоту дискретизации, и затем нажмите **Обновить (Update)** и **ОК**.
15. Сохраните фильм под именем `audio_import fla` для дальнейшего использования в этой главе.

Ну, хорошо, со сжатием немного разобрались, теперь пора рассмотреть различные способы применения звука во Flash MX.

Потоковая обработка и события

Как и видео, звук во Flash может быть реализован несколькими методами; они называются типами синхронизации. Эти типы — Event (событие), Start (старт), Stop (стоп), и Stream (поток). Выбираются они из списка Синхро (Sync) в звуковых опциях инспектора Параметры (Property):



Поток

Потоковый звук трактуется таким же образом, что и потоковое видео, и тоже буферизуется перед воспроизведением. Однако, в отличие от видео, здесь буферизация встроенная, поэтому нет необходимости заполнять монтажный стол кадрами, чтобы «купить» время на загрузку — звуковой поток синхронизирован с монтажным столом.

Стандартное значение времени буферизации во Flash — 5 секунд, но его легко можно изменить при помощи ActionScript — для этого предназначен параметр `_soundbuftime`. Как и для видео, потоковая обработка звука возможна только на главном монтажном столе. Если вам понадобилось уменьшить буфер, следует вставить в кадр 1 вашего фильма такое действие:

```
_soundbuftime = 3;
```

Тем самым вы получите трехсекундный буфер, это значит, что Flash сделает паузу длительностью 3 секунды, прежде чем начнет воспроизводить вашу звукозапись. Напоминаю, что `_soundbuftime` — глобальный параметр, поэтому обычный префикс `_root` не требуется.

Опция Stream хороша тем, что в фильм загружается только необходимая часть файла звукозаписи. Так, если аудио файл немного длиннее, чем ваш монтажный стол, этот излишек будет проигнорирован, сэкономив драгоценный размер файла.

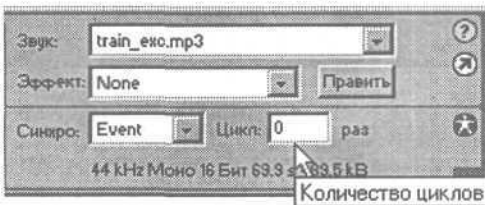
Протестируйте ваш поток — при помощи Ctrl + Enter запустите проверку фильма, а затем выполните команду меню Вид > Показать поток (View > Show Streaming).

Событие

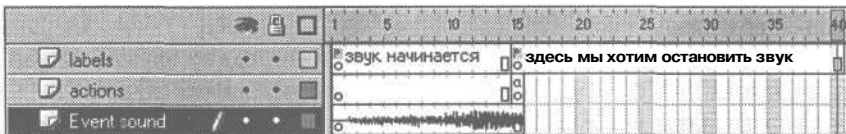
Как видно из названия, звук типа Event (Событие) воспроизводится тогда, когда происходит некоторое событие. Таким событием может быть нажатие на кнопку, окончание игры, музыкальное вступление и тому подобное — любые звуки, связанные с чем-либо.

Звуковые события не могут обрабатываться как поток; подобно фильму-символу, они должны быть полностью загружены в фильм, прежде чем будут воспроизведены.

Для звуковых событий можно задать определенное количество циклических повторов в секции звуковых опций инспектора Параметры (Property):



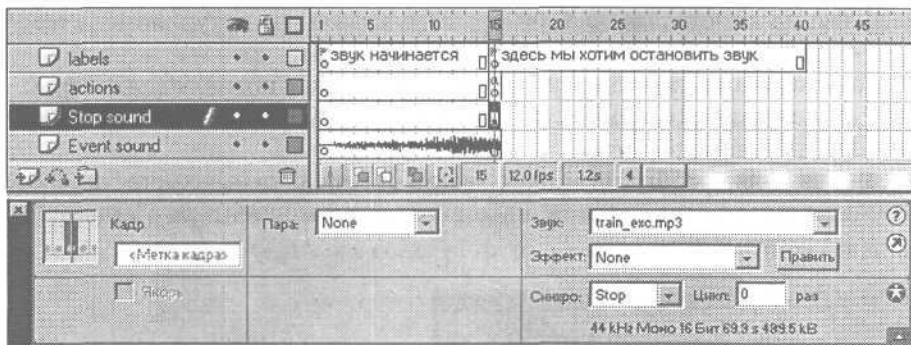
У звуковых событий есть одно важное свойство, о котором не следует забывать: они всегда воспроизводятся до конца звукозаписи. Воспроизведение не будет остановлено, даже если закончатся кадры на монтажном столе или встретится команда stop. На следующей иллюстрации полная длительность звукозаписи соответствует 840 кадрам, однако мы хотели бы остановить ее на 15-м кадре, поэтому лишние кадры мы обрезаем, и в кадр 15 вставили действие stop.



Хотите — верьте, хотите — проверьте, но звукозапись будет проиграна полностью, до конца, несмотря на всю нашу защиту! Но остановить ее все-таки можно — при помощи пустого ключевого кадра, помещенного в последний кадр слоя sound или путем использования звука типа **Stop...**

Start и Stop

Эти варианты являются просто расширениями типа Event, позволяющими контролировать момент начала воспроизведения звука (это эквивалент типа Event) и его остановки. Если в предыдущем примере мы добавим новый слой «Stop sound» и ключевой кадр в кадр 15, мы сможем остановить звукозапись событием Stop:



Честно говоря, хотя этот метод позволит сэкономить один кадр в вашем фильме, куда проще остановить звукозапись вставкой пустого ключевого кадра в тот же слой, где находится звуковое событие.

Из всех типов синхронизации варианты Stream и Event применяются чаще всего. Если звук используется в фильме периодически — как, например, звонок на старте игры — тогда ставьте тип Event.

Если же у вас длинный аудио файл, который должен воспроизводиться однократно и целиком, для вашей аудитории удобнее будет потоковая передача звука — тогда не придется ждать, пока весь файл загрузится. При тестировании вашего фильма во Flash не забывайте о том, какова пропускная способность канала у вашей аудитории и устанавливайте для опции Показать поток (Show Streaming) наименьшую возможную скорость загрузки.

Формат MP3 намного упрощает потоковую передачу звука, долгой буферизации не требуется и пользователю не приходится ждать.

Теперь, когда мы рассмотрели основные опции звука, пора узнать, какие эффекты мы можем применить во Flash к нашим аудио файлам.

Звуковые эффекты Flash

Давайте сразу определимся: здесь под «эффектами» подразумеваются отнюдь не те виды спецэффектов, о которых говорилось в начале этой

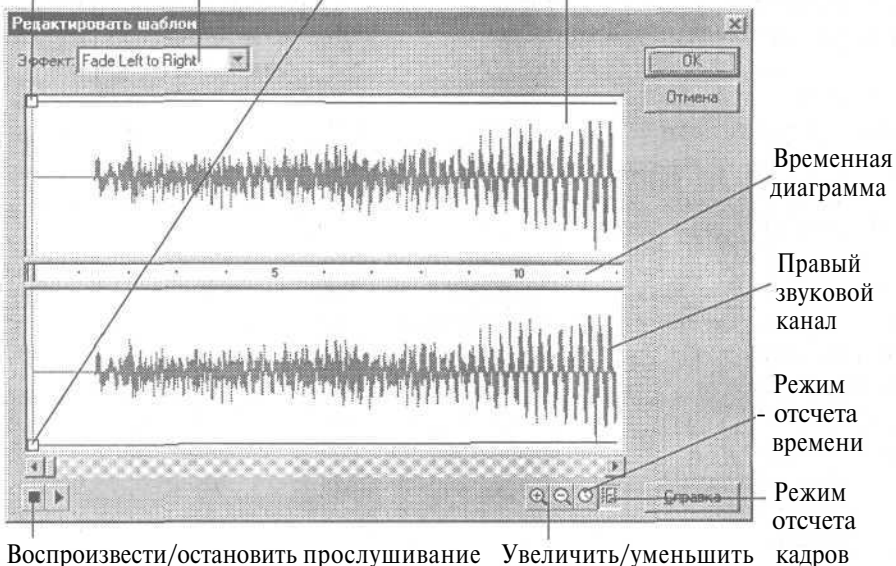
главы при описании приложений для редактирования звукозаписей. Flash умеет управлять громкостью и панорамой, и вы сможете получить контроль над этими параметрами.

Чтобы открыть окно Редактировать шаблон (Edit Envelope), следует выбрать нужную звукозапись в инспекторе Параметры (Property), а затем нажать кнопку Править (Edit...):



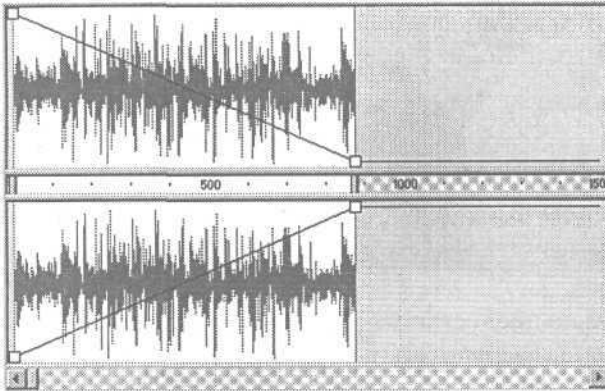
Здесь имеется несколько предустановленных эффектов, и они пригодятся нам, чтобы понять, как работать с этим диалоговым окном. Из списка Эффект (Effect) выберите Fade Left to Right, и мы рассмотрим все подробно:

Левая узловая точка Эффект Правая узловая точка Левый звуковой канал

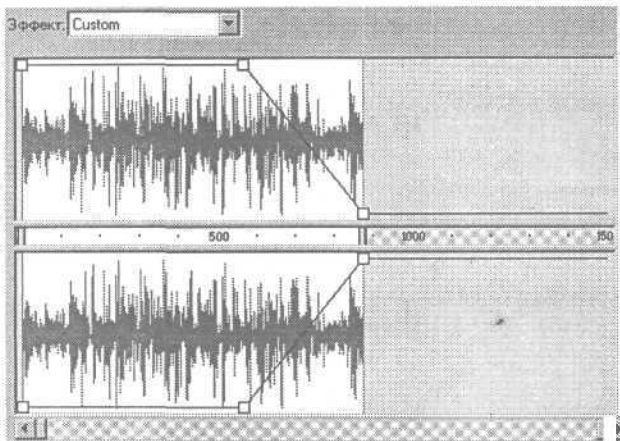


Посмотрев на левый звуковой канал, вы увидите, что узловая точка здесь находится у верхнего обреза окна, что означает полную громкость, тогда как громкость правого канала установлена на ноль. Выбранный

эффект задает постепенное перемещение звука из одного канала в другой на протяжении всей звукозаписи (чтобы увидеть эффект целиком, нужно уменьшить изображение звукозаписи):



Узловую точку можно перемещать в пределах окна, можно добавить новую узловую точку, щелкнув где-либо на линии. Таким путем, к примеру, можно сделать так, чтобы переход происходил в последних сценах фильма, а не с самого начала (обратите внимание, что в списке Эффект (Effect) теперь указано Custom — «заказной»):



Вы можете добавить в рабочую область до восьми узловых точек, обозначающих этапы изменения звука. Может статься, что предустановленные эффекты прекрасно вам подходят, но при необходимости вы можете пойти дальше и настроить их в соответствии с содержанием своего фильма. Например, быстрое затухание звука может пригодиться,

когда действие, происходящее в фильме, заканчивается для его участников неудачей.

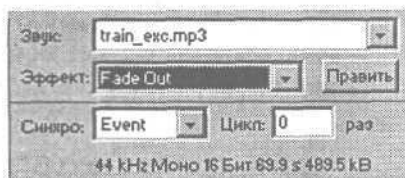
А теперь давайте применим то, что мы узнали, к нашему звуковому файлу.

Подправляем звукозапись

1. Откройте файл `audio_import fla`, с которым мы уже работали в этой главе.
2. Вставьте новый слой под названием «actions», и в последний кадр этого слоя (кадр 840 в нашем примере) вставьте ключевой кадр (F6). Поместите в этот кадр действие `stop`.

Сейчас моя звукозапись как-то очень неудачно обрывается в конце, и я намерен смягчить концовку при помощи небольшого затухания.

3. Выделите первый кадр слоя `audio` и выберите `Fade Out` из списка Эффект (Effect) в инспекторе Параметры (Property):



4. Нажав комбинацию клавиш `Ctrl + Enter`, протестируйте фильм.

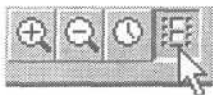
Затухание получилось слишком долгое, надо бы его укоротить. Для этого я должен выяснить, откуда должен начинаться фединг — определить стартовую точку. Это можно сделать двумя способами: используя Профиль полосы (Bandwidth profiler) или следя за курсором воспроизведения. Но можно и схитрить.

Хитрость простая: надо сменить тип синхронизации на `Stream`. Тогда Flash позволит прослушивать звук при перетаскивании курсора воспроизведения по монтажному столу.

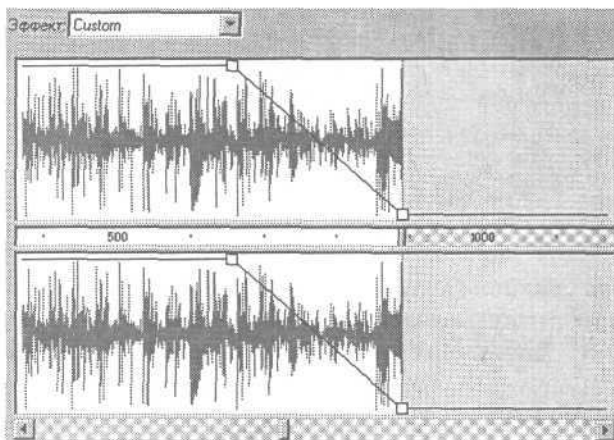
Давайте так и сделаем.

5. Не снимая выделения с кадра 1 слоя `audio`, отправляйтесь в инспектор Параметры (Property) и выберите `Stream` из списка Синхро (Sync).
6. Теперь протащите курсор воспроизведения вдоль монтажного стола и найдите точку, где должно начинаться затухание. После небольшого исследования я выбрал кадр 775, а вы вольны выбрать свою точку.

7. В секции звуковых опций инспектора Параметры (Property) смените тип Синхро (Sync) опять на Event и нажмите кнопку Править (Edit...). Мы отредактируем точку фединга в окне Редактировать шаблон (Edit Envelope).
8. Нажмите кнопку переключения в режим отсчета кадров — теперь на центральной шкале вместо времени будут отсчитываться кадры фильма Flash:



9. Перетащите узловую точку из текущей позиции примерно к кадру 775. Не имеет значения, точку какого канала вы будете перетаскивать; Flash всегда двигает их параллельно (если вы нечаянно добавили лишние узловые точки, не переживайте, вы всегда можете удалить их, просто утащив за пределы окна соответствующего канала):



10. Увеличив изображение, вы сможете поместить узловую точку точно в кадр 775. Нажмите ОК, чтобы подтвердить изменения.
11. Протестируйте фильм, убедитесь, что все правильно.

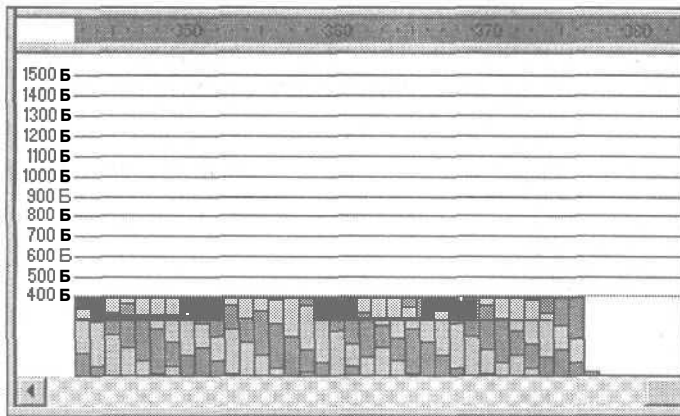
Итак, звукозапись отредактирована; давайте посмотрим, что у нас с потоком и буферизацией...

12. Закройте окно тестирования и измените тип синхронизации на Stream (да, опять!)

Как уже упоминалось, стандартный буфер во Flash — 5 секунд. Посмотрим, как проходит потоковая передача нашей музыки при такой буферизации.

13. Запустите тест фильма и в меню Отладка (Debug) выберите пункт 56К. Теперь выполните команду Вид > Показать поток (View > Show streaming). Зеленая полоса молнией пронесется по временной диаграмме, и через 5 секунд начинается воспроизведение звука.
14. Прокручивая временную диаграмму по ходу проигрывания, вы увидите, что загрузка звукозаписи завершается к кадру 374.

В результате мы получаем очень плавное воспроизведение, без задержек:



Что же это означает? Да то, что мы можем себе позволить буфер менее 5 секунд! В принципе, 5 секунд — не такое уж долгое ожидание для зрителя, но мы отважно дадим даже меньше! Давайте попробуем магическое число 3.

15. Закройте окно тестирования и вернитесь на основной монтажный стол.
16. Выделите кадр 1 слоя actions и откройте панель Действия (Actions) (F9). Выбрав Экспертный режим (Expert Mode) (Ctrl+ Shift+ E), введите следующую строку кода:

```
_soundbuftime = 3;
```

17. И все. Снова запустите тест и выполните Вид > Показать поток (View > Show streaming).

Проследив за зеленой полосой, вы заметите, что загрузка звука заканчивается задолго до того, как индикатор воспроизведения догонит зеленую полосу. В результате, мы вновь получаем плавное воспроизведение. Можно попытаться счастья еще, но лучше остановиться, пока мы в выигрыше!

18. Сохраните ваш фильм и закройте его.

Заключение

Работая со звуком во Flash MX (как, впрочем, и с другими видами информации), важно найти правильный баланс между качеством и размером файла; об этом следует помнить как на этапе предварительной подготовки, так и в процессе импортирования.

Не исключено, что те возможности управления звуком, которые доступны непосредственно из интерфейса Flash MX и инспектора Параметры, вы сочтете недостаточными. В следующей главе мы пойдем дальше и научимся более сложным манипуляциям со звуком, рассматривая звукозапись как программный объект и используя для управления ActionScript.

Объект типа Sound

Краткое содержание главы:

- *свойства и методы, присущие объекту Sound;*
- *как управлять объектом Sound посредством ActionScript;*
- *создание при помощи ActionScript динамического MP3-проигрывателя.*

В предыдущей главе мы научились помещать звукозаписи на монтажный стол, организовывать их потоковую передачу, немного управлять ими при помощи диалогового окна Редактировать шаблон (Edit Envelope). Все то же самое можно сделать с **объектом** типа **Sound**, и при этом мы получим еще ценное преимущество — возможность управлять его свойствами «на лету» при помощи ActionScript.

ActionScript является ныне полноценным языком **объектно-ориентированного программирования** (object-oriented programming, **ООП**) и дает возможность не только работать с различными объектами, но и создавать их в динамике. Основные типы объектов во Flash — это фильмы-символы, текстовые поля, компоненты и, конечно же, звук. В предыдущей главе звук был представлен в совершенно сыром виде (как бы сам по себе) и, соответственно, не мог контролироваться посредством ActionScript. Чтобы получить возможность такого контроля, звук должен быть представлен как объект.

Знакомство с объектом Sound

В следующем скрипте объявляется новый объект типа Sound, в него помещается звуковой файл из Библиотеки под названием «rockSong», затем происходит обращение к объекту — его воспроизведение:

```
mySound = new Sound ();  
mySound.attachSound ("rockSong");  
mySound.start ();
```

Если что-то здесь не понятно, не волнуйтесь — мы будем часто использовать этот скрипт, и у вас будет возможность поближе с ним познакомиться.

Объекту Sound присущи несколько встроенных методов, в том числе:

start

Начинает воспроизведение объекта типа Sound с указанной отметки. Например, команда

```
mySound.start (5);
```

начнет воспроизведение объекта mySound с отметки 5 секунд. Если параметр стартовой отметки опущен, то mySound будет воспроизведен немедленно, с самого начала. Этот метод позволяет также указать количество циклов воспроизведения после параметра стартовой отметки. Например,

```
mySound.start(5, 2);
```

воспроизведет звукозапись дважды со стартовой отметки 5 секунд. По умолчанию звукозапись воспроизводится один раз.

stop

Останавливает воспроизведение звука. Если вы в ActionScript не укажете, какой именно звук следует остановить, Flash остановит воспроизведение *всех* звукозаписей, воспроизводимых в данный момент.

setPan

Этот метод позволяет смещать баланс между левым и правым каналами объекта Sound. Значения могут быть от -100 (звук полностью смещен влево) до 100 (полностью вправо). По умолчанию setPan установлен на 0, то есть звуковая панорама расположена по центру, без смещения.

setVolume

Позволяет изменять громкость объекта Sound от 0 (тишина) до 100 (полная громкость). Значение по умолчанию равно 100.

loadSound

Теперь, с помощью этого нового метода, Flash MX позволяет импортировать файлы MP3... и даже более того! Flash может не только импортировать MP3-файл, но и организовать его потоковую передачу. В следующем примере определяется файл MP3 и указывается, что его передача в фильм должна вестись потоком:

```
mySound.loadSound (http://www.yourhomepage.com/discopop.mp3, true);
```

Если значение параметра `isStreaming` равно `true`, идет потоковая передача звука. Если же `false` — Flash импортирует MP3 как звуковое событие, то есть загружает полностью, прежде чем воспроизвести.

Таким образом, теперь во Flash MX можно создать динамический MP3 проигрыватель.

Дополнительные методы

Существует несколько методов, позволяющих получить текущие значения объекта `Sound`, в том числе `getPan` и `getVolume`. Что более важно, Flash имеет возможность получить значения двух таких элементов объекта, как:

- **duration:** продолжительность звукозаписи;
- **position:** позиция указателя воспроизводимого в данный момент клипа.

Если вы мало что поняли во всех этих терминах из области программирования, не стоит пугаться — большинство описанных методов мы используем в следующем упражнении, вы увидите, как они работают. К концу главы вы будете общаться с ними, как с близкими друзьями!

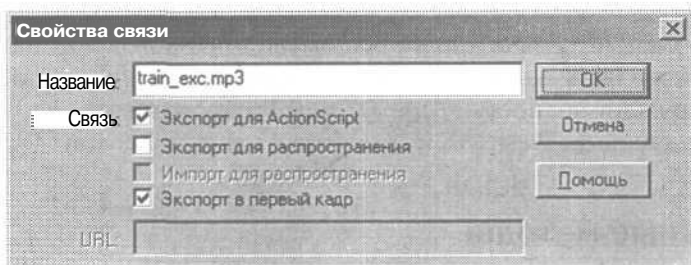
Динамический звук

В этом упражнении мы создадим звуковое мини-приложение, чтобы показать функциональные возможности, которые дает использование объекта `Sound` в связке с объектно-ориентированным `ActionScript`. Начнем мы с двух важнейших базовых элементов — громкости и баланса.

1. Откройте новый фильм Flash и импортируйте не слишком длинный звуковой файл (`Ctrl + R`). Я в этом упражнении буду использовать все тот же файл `train_exc.mp3`, с которым мы работали в предыдущей главе (его можно взять на www.friendsofed.com), а вы, если хотите, можете взять какой-либо свой файл.
2. Чтобы управлять нашим MP3 при помощи `ActionScript`, нужно дать ему имя связи. Откройте Библиотеку (Library) (`F11`) и щелкните пра-

вой кнопкой мыши на имени звукового файла. В меню выберите пункт Связи (Linkage...).

3. Перед вами появится окно Свойства связи (Linkage Properties). Установите флажок Экспорт для ActionScript (Export for ActionScript), после чего произойдут два события: автоматически установится флажок Экспорт в первый кадр (Export to First Frame), и станет доступным неактивное до этого поле Название (Identifier):



4. Измените Название (Identifier) на «music» и нажмите кнопку ОК.
5. Теперь давайте подключим наш звук. Переименуйте имеющийся на монтажном столе слой в «actions» и, выделив кадр 1, откройте панель Действия (Actions) (F9). Введите следующий код:

```
mySound = new Sound ();
mySound.attachSound ("music");
mySound.start ();
```

Скрипт практически тот же, что и в начале главы. Единственное различие состоит в том, что во второй строке кода указано другое имя связи. Давайте разберем этот код строка за строкой. Мы указываем Flash, что он должен:

- создать новый объект типа Sound под названием mySound;
- взять из Библиотеки аудио файл с именем связи music и поместить его в mySound;
- начать воспроизведение mySound с самого начала, без повторов.

Яв этой главе частенько использую имя mySound, но вы можете назвать ваш объект типа Sound как угодно. Впрочем, постарайтесь, чтобы имя не совпадало с именами переменных, командами ActionScript и другими зарезервированными словами Flash, иначе в вашем скрипте могут возникнуть проблемы.

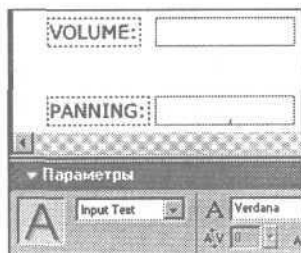
6. А теперь протестируйте фильм, и вы услышите вашу запись. Помните, мы сделали это всего лишь при помощи нескольких строчек ActionScript и нового объекта типа Sound, безо всяких длиннющих монтажных столов.

Но не стоит слишком обольщаться — в конце концов, это только однократно проигранная звукозапись. Мы пойдем дальше и начнем манипулировать звуком в динамике. Давайте попробуем создать простые поля для ввода информации — в них пользователь будет вводить значения громкости и баланса для воспроизводимого звука.

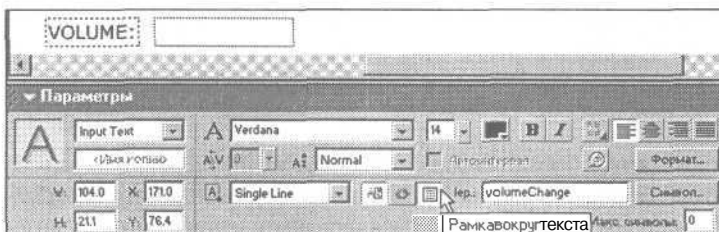
7. Закройте ваш тестовый SWF. На монтажном столе создайте новый слой «text». В кадре 1, с помощью инструмента Текст (Text), изобразите два статичных текстовых поля (Static Text) с надписями «VOLUME:» (громкость) и «PANNING:» (баланс):



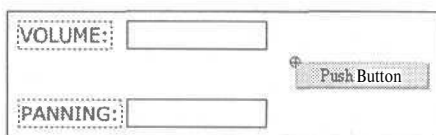
8. Рядом с каждым изобразите примерно такие же текстовые поля для ввода (Input Text):



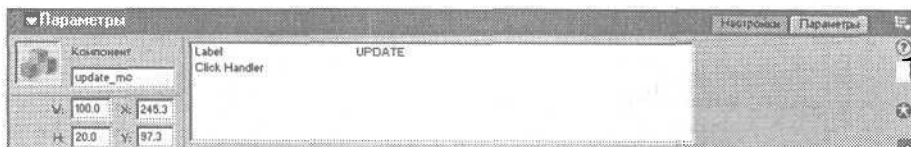
9. Выделите поле для ввода рядом с надписью VOLUME. Нажмите кнопку Рамка вокруг текста (Show Border Around Text) в инспекторе Параметры (Property), в поле Пер. (Var) (переменная, variable) введите «volumeChange»:



10. Туже процедуру повторите для поля рядом с надписью PANNING, только переменная будет называться panChange.
11. Текстовые поля готовы, теперь нам нужна кнопка, при нажатии на которую Flash должен изменить параметры звука в соответствии с данными в текстовых полях. Вставьте новый слой под названием «updateButton».
12. Выделите кадр 1 слоя updateButton. Чтобы не рисовать кнопку и сэкономить время, откройте панель Компоненты (Components) (Ctrl + F7) и перетащите копию компонента PushButton в рабочую область, поместив рядом с полями ввода:



13. Выделив PushButton в рабочей области, отправляйтесь в инспектор Параметры и дайте копии имя «update_mc». Затем щелкните в поле Label и введите «UPDATE»:



14. Теперь, чтобы эта кнопка могла управлять объектом Sound, нужно добавить еще ActionScript. Выделив кадр 1 слоя actions, откройте панель Действия (Actions) и введите под предыдущими тремя строчками следующий код:

```

this.update_mc.onRelease = function() {
    mySound.setPan(panChange);
    mySound.setVolume(volumeChange);
};

```

Здесь используются уже рассматривавшиеся в этой главе методы `setPan` и `setVolume` для обновления параметров объекта `mySound` (или звукового файла) после нажатия на компонент `PushButton`.

15. Запустив тест фильма, попробуйте вводить различные значения и нажимать кнопку. Помните, что значение громкости изменяется в пределах от 0 до 100, а значение баланса — от -100 (только левый канал) до 100 (только правый).
16. Сохраните фильм как `sound_app_1.flai` и оставьте открытым для следующего упражнения.

На данный момент наше приложение не слишком «умно», не знает пределов громкости и баланса, и пользователь может задрать уровни в область искажений. В следующем упражнении мы установим ограничения, чтобы уровни баланса и громкости не могли выйти за пределы надлежащих значений.

Добавляем графический интерфейс

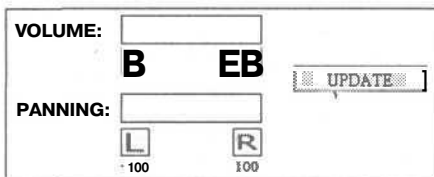
Мы можем усовершенствовать наше маленькое приложение, добавив кнопки «плюс» и «минус»; тем самым мы дадим пользователю нечто более похожее на графический интерфейс и вернем ему возможность комфортно управлять звуком при помощи мыши.

1. Если вы еще не закрыли `sound_app_1.flai`, созданный в последнем упражнении, сохраните его как `sound_app_2.flai`, и мы начнем построение нашего интерфейса.
2. Вставьте новый слой под названием «small buttons». В кадре 1 нарисуйте маленький квадратик и преобразуйте его в графический символ (F8) под названием «small square». Это будет базовая форма для четырех кнопок.
3. Теперь выделите этот графический символ и преобразуйте его в символ-кнопку под названием «plus volume button». На монтажный стол `plus volume button` добавьте новый слой и в нем, в состоянии Вверх (Up) нарисуйте значок плюс:



4. Создайте остальные три кнопки, дублируя «plus volume button» и заменяя знак плюса на минус, «L» и «R». Назовите их «minus volume button», «left button» и «right button» соответственно.

5. Также можете добавить маленькие подписи значений баланса, чтобы подсказать пользователю, какие числа можно вводить. Вот так выглядит весь интерфейс:



Теперь каждой кнопке следует дать инструкции. Я решил сделать так, чтобы при каждом нажатии на кнопку соответствующее значение увеличивалось или уменьшалось на 5, иначе пользователю придется щелкать мышью слишком много раз.

6. Выделите кнопку minus volume button и присоедините к ней следующий код:

```
on (release) {
    volumeChange -= 5;
}
```

При этом из значения переменной volumeChange вычитается 5 (эквивалент выражения `volumeChange = volumeChange - 5`).

7. Для plus volume button введите следующий код:

```
on (release) {
    volumeChange += 5;
}
```

8. Выделив left button, добавьте такой скрипт:

```
on (release) {
    panChange -= 5;
}
```

9. Наконец, для right button код будет такой:

```
on (release) {
    panChange += 5;
}
```

Ну вот, наши маленькие кнопки закончены. Но помните, что я говорил об **предельных значениях**? Если сейчас вы протестируете фильм, то обнаружите, что можете повышать громкость не только до 100, но и **далее!** Нам нужно установить два набора ограничений:

- громкость не может быть больше 100 и меньше 0;
- баланс не может быть больше 100 и меньше -100.

10. Выделите кадр 1 слоя actions и вставьте код, выделенный жирным шрифтом, чтобы ActionScript для кадра 1 выглядел так:

```
mySound = newSound();  
mySound.attachSound ("music");  
mySound.start();  
this.update_mc.onRelease = function() {  
    if (volumeChange>100) {  
        volumeChange = 100;  
    }  
    if (volumeChange<0) {  
        volumeChange = 0;  
    }  
    if (panChange>100) {  
        panChange = 100;  
    }  
    if (panChange<-100) {  
        panChange = -100;  
    }  
    mySound.setPan(panChange);  
    mySound.setVolume(volumeChange);  
};
```

Смысл нового кода простой: «если пользователь вводит значения громкости или баланса, выходящие за установленные пределы, устанавливать соответствующее предельное значение».

11. Проверьте фильм, попробуйте изменять параметры объекта Sound, пользуясь кнопками.

Возможно, теперь вы захотите пользоваться только кнопками — что ж, это вполне нормально, хотя с цифрами и быстрее, и точнее. В идеале это приложение надо бы снабдить движковыми регуляторами, но это я оставляю вам — интегрируйте их в свое приложение самостоятельно.

12. Сохраните фильм и закройте его.

Звук и видео в «свободном плавании»

Последнее упражнение показало некоторые возможности манипулирования звуком при помощи ActionScript. Однако хотя мы определили наше маленькое приложение как динамическое, в одном отношении оно было

нединамическим — оно зависело от ввода информации пользователем. В следующем упражнении мы покажем, как создать более «самостоятельное» приложение.

Использование `setVolume`

Мы собираемся использовать метод `setVolume` и создать звукозапись, громкость которой будет «плавать» во время воспроизведения.

1. Откройте новый фильм Flash и импортируйте не слишком длинный звуковой файл. Я йо-прежнему буду использовать `train_exc.mp3`, а вы можете взять все, что угодно.
2. Щелкните правой кнопкой на звуковом файле в Библиотеке (Library), выберите в меню команду Связь (Linkage...) — откроется окно Свойства связи (Linkage Properties).
3. Установите флажок Экспорт для ActionScript (Export for ActionScript), в поле Название (Identifier) укажите имя связи «fluxer».
4. Для управления этим звуком понадобится ActionScript. Переименуйте имеющийся на монтажном столе слой в «actions», в кадр 1 добавьте следующий код:

```
speed = 1;
volumeChange = 100;
volumeAdd = speed;
mySound = new Sound ();
mySound.attachSound ("fluxer");
mySound.start ();
```

В первых трех строчках инициализируются переменные для фильма:

- `speed` — число, на которое изменяется значение громкости за один шаг;
- `volumeChange` — на старте фильма устанавливает громкость на 100. В дальнейшем по ходу фильма в этой переменной содержится значение уровня громкости;
- `volumeAdd` — реальное значение, добавляемое (или вычитаемое) к громкости на каждом шаге.

Остальная часть кода вам знакома. Создается объект типа `Sound`, берется звуковой файл с именем связи `fluxer` и начинается его воспроизведение.

5. Теперь добавьте такую строку:

```
createEmptyMovieClip ("mc", 1);
```

В рабочей области создается пустой фильм-символ, ему присваивается имя копии (идентификатор) `mc`. Это то же самое, что создать обычный фильм-символ во `Flash`, только здесь мы это сделали при помощи `ActionScript`.

6. Продолжаем и добавляем следующий код:

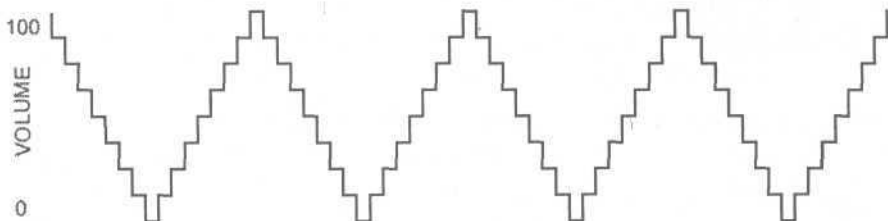
```
mc.onEnterFrame = function () {  
    if (volumeChange >= 100) {  
        volumeAdd = -speed;  
    }  
    if (volumeChange <= 0) {  
        volumeAdd = speed;  
    }  
    volumeChange += volumeAdd;  
    mySound.setVolume (volumeChange);  
};
```

Первая строка: пустой фильм-символ с именем `mc` используется, чтобы создать событие, происходящее в каждом кадре. Код, который обрабатывается в каждом кадре, заключен в фигурные скобки.

Оператор `if` проверяет, не достигла ли громкость максимума или минимума. Если это так, то знак переменной `volumeAdd` изменяется на противоположный, в результате чего громкость постоянно то взлетает, то падает («плавает»).

В последней секции, после оператора `if`, происходят две вещи: значение переменной `volumeChange` обновляется, затем она используется для установления соответствующей громкости звука.

7. Что ж, не знаю, все ли вы поняли; в любом случае, проверьте фильм. Возможно, все станет понятнее, когда вы это услышите. Громкость непрерывно изменяется от максимума до минимума и обратно:



Если вам кажется, что «волны» слишком медленные, попробуйте увеличить значение переменной скорости, и вы сразу почувствуете результат. Из этого вы легко сможете сделать эффект плавающей панорамы, когда звук «уходит» то в левый канал, то в правый. Все, что для этого нужно — изменить значение минимума в операторе `if`: `-100` вместо `0`, а также заменить последнюю строку на `mySound.setPan (volumeChange);`.

Кроме того, неплохо было бы дать переменным более точные имена, такие, как `panChange` и `panAdd`, это я оставляю на ваше усмотрение. Если будете тестировать этот эффект плавающей панорамы, не забудьте расставить пошире колонки.

А теперь пойдём ещё дальше и оживим наш плавающий звук визуальным элементом — видеоклипом.

Видео «плавает» вместе со звуком

Мы собираемся сделать так, чтобы альфа-фактор нашего видео «плавал» так же, как звук в предыдущем упражнении, от 0% до 100%. При этом мы можем использовать одни и те же переменные для управления и звуком, и видео.

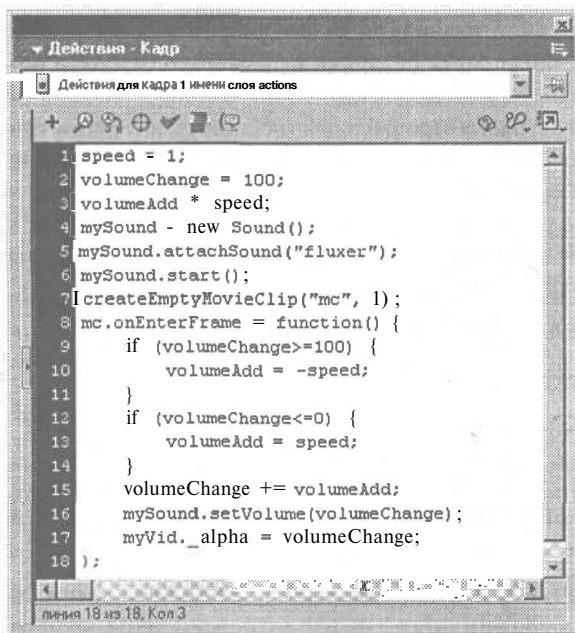
1. Продолжая работать с тем же самым фильмом, создайте новый фильм-символ (`Ctrl + F8`) под названием «video».
2. Импортируйте видео файл в фильм-символ `video`. Я использовал `halfpipe.mov`, если хотите, скачайте его и вы.
3. Вернувшись на основной монтажный стол, вставьте новый слой под названием «video».
4. Выделив кадр 1 этого слоя, перетащите копию фильма-символа `video` на основную сцену и присвойте копии имя «myVid»:



5. Теперь вновь откройте панель Действия (Actions) для кадра 1 слоя actions. После последней строки, но внутри фигурных скобок вставьте следующий код:

```
myVid._alpha = volumeChange;
```

Окончательный вид кода должен быть таким:



```
1 speed = 1;
2 volumeChange = 100;
3 volumeAdd * speed;
4 mySound = new Sound();
5 mySound.attachSound("flucker");
6 mySound.start();
7 createEmptyMovieClip("mc", 1);
8 mc.onEnterFrame = function() {
9     if (volumeChange >= 100) {
10        volumeAdd = -speed;
11    }
12    if (volumeChange <= 0) {
13        volumeAdd = speed;
14    }
15    volumeChange += volumeAdd;
16    mySound.setVolume(volumeChange);
17    myVid._alpha = volumeChange;
18 };
```

6. Конец! Можете протестировать фильм. Теперь ActionScript управляет одновременно двумя объектами: видео фильмом-символом и объектом Sound.
7. Сохраните ваш фильм и закройте его.

Мы увидели, какие преимущества дает использование объекта типа Sound, тем не менее, полностью игнорировать традиционный способ размещения звука на монтажном столе тоже нельзя. Давайте посмотрим, как синхронизировать звук с видео.

Параметр position

Если нужно синхронизировать видео со звуком, то лучше всего перенести видео во Flash вместе со звуковой дорожкой, в процессе импорта. Если же в какие-то моменты синхронизация теряется, и возникает не-

обходимость оптимизировать и обрабатывать звук отдельно, это можно сделать, используя параметр **position** объекта типа **Sound**.

Параметр **position** постоянно обновляется в ходе воспроизведения объекта **Sound**, получить его значение можно таким образом:

```
whereltsAt = mySound.position;
```

Переменная **whereltsAt** запрашивает объект **mySound**, где в данный момент находится указатель воспроизведения. Результат возвращается в миллисекундах, поэтому, например, если указатель находится на отметке 6 секунд, то будет возвращено значение 6000.

В следующем упражнении мы собираемся использовать это свойство в связке с методом **start** объекта типа **Sound**, передавая ему значение параметра. Неудобство в том, что метод **start** воспринимает параметр в секундах, поэтому, если мы укажем значение 6000, он поймет это как 1 час 40 минут. Чтобы обойти это и получить значение в секундах, нудно просто разделить число, возвращаемое параметром **position**, на 1000:

```
whereltsAt = mySound.position / 1000;
```

Проблема решена. Переходим к упражнению...

Я и мой звук \

В этом упражнении мы используем отдельные файлы видео и звука, но будем останавливать и запускать их одновременно при помощи **ActionScript**.

1. Откройте новый фильм **Flash** и импортируйте достаточно длинный видео клип (я использовал **halfpipe.mov** — тот же самый файл, что и в предыдущем упражнении). Как обычно, импортируйте видео в новый фильм-символ, дайте символу подходящее название.
2. Перетащите копию фильма-символа в рабочую область и дайте копии имя «**myVid**».
3. Импортируйте достаточно продолжительный звуковой файл (я опять использую **train_exc.mp3**) и дайте ему имя связи «**music**».
4. Вставьте новый слой под названием «**buttons**» и создайте в рабочей области три простые кнопки для воспроизведения, паузы и остановки. Можете нарисовать кнопки самостоятельно, а можете сэкономить время, используя встроенные кнопки **Flash**, которые можно найти в папке **Playback**, выполнив команду **Окно > Основные библиотеки > Buttons (Window > Common Libraries > Buttons)**.

Я перетащил себе кнопки **gel Stop**, **gel Right** и **gel Pause**:



Давайте продумаем действие кнопок. Мы в этом упражнении собираемся использовать переменную для проверки, воспроизводится ли в данный момент видео и звук. Эту переменную мы используем для кнопок воспроизведения и паузы, чтобы в каждый момент времени могла функционировать только одна из них.

5. Создайте новый слой под названием «actions» и откройте панель Действия (Actions) для кадра 1. Введите следующий код:

```
playing = true;  
mySound = new Sound ();  
mySound.attachSound ("music");  
mySound.start ();  
myVid.play ();
```

Все это мы уже видели, стандартная инициализация объекта типа Sound, и я не буду повторяться. Единственное новое добавление — в первой строке: это та самая переменная, о которой мы только что говорили. Последняя строка кода — указание начать воспроизведение фильма-символа video одновременно со звуком.

6. Выделите кнопку паузы и дайте копии имя «`pause_mc`». Выделите кадр 1 слоя actions и введите следующий код после имеющегося:

```
pause_mc.onPress = function () {
    if (playing==true) {
        pausePos = (mySound.position)/1000;
        mySound.stop();
        myVid.stop();
        playing = false;
    }
};
```

Давайте переведем этот скрипт на обычный русский язык:

```
Когда на меня, кнопку паузы, нажимают {
    если (видео и звук воспроизводятся
        ↳ (проверяется переменная)), нужно выполнить {
        поместить в переменную pausePos текущую позицию
        ↳ звука, деленную на 1000
        остановить воспроизведение звука
        остановить видео
        Изменить значение переменной playing на «видео
        ↳ не воспроизводится»
    }
};
```

Вообще-то это никакая не пауза, а чистейшее надувательство. На самом деле кнопка останавливает звук и сбрасывает на начало, тогда как видео действительно только приостанавливается (командой `stop!`). Прежде, чем сбросить звук, запоминается текущая позиция, делится на 1000, чтобы получить значение в секундах.

7. Возможно, вас удивляет, почему мы так хлопочем об этом, но сейчас вы все поймете. Выделите кнопку воспроизведения и присвойте копии имя «`play_mc`». Добавьте к предыдущему коду следующий (в кадре 1 слоя actions):

```
play_mc.onPress = function () {
    if (playing==false) {
        mySound.start (pausePos);
        myVid.play();
        playing = true;
    }
};
```

Это переводится так:

```

Когда на меня нажмут {
    если (видео не воспроизводятся), нужно выполнить {
        Запустить звук с сохраненной позиции
        Возобновить воспроизведение видео
        Изменить значение переменной playing на «видео
        ↳ воспроизводится»
    }
};

```

На предыдущем шаге мы сохранили позицию, в которой мы остановили музыку, в переменной `pausePos`. Теперь, когда пользователь захотел продолжить просмотр, мы можем запустить звук с того же места, где он был остановлен.

- Выделите кнопку остановки и присвойте копии имя «`stop_mc`». Добавьте к предыдущему коду следующий:

```

stop_mc.onPress = function () {
    mySound.stop();
    myVid.gotoAndStop (1);
    pausePos = 0;
    playing = true;
};

```

Скрипт полностью останавливает звук, но на сей раз позиция *не* сохраняется. Затем отправляет видео на кадр 1 и там останавливает. Далее переменной `pausePos` присваивается 0. Это сделано потому, что кнопка воспроизведения всегда запускает звук с сохраненной позиции. Наконец, скрипт изменяет значение переменной `playing` на `false`. Если пользователь нажмет кнопку воспроизведения теперь, и видео, и звук будут запущены с начала.

- Проверьте фильм. Единственной проблемой может оказаться то, что ваш видео файл длиннее, чем звукозапись. Чтобы исправить это, звуковой файл можно зациклить. Для этого просто добавьте две строчки в последний кадр вашего фильма-символа `video`:

```

_root.mySound.stop();
gotoAndStop(1);

```

Тем самым и звук, и видео будут остановлены по достижении конца видео файла. Но что, если раньше закончится звуковой файл? У нас есть в запасе такой трюк...

Не прекращать до onSoundComplete

Событие onSoundComplete — еще один образчик ActionScript, который поможет синхронизировать звук и видео. Это событие — новинка Flash MX. Как видно из его имени, событие происходит, когда воспроизведение объекта типа Sound завершено. Код самую малость посложнее, чем мы видели до сих пор:

```
mySound.onSoundComplete = function () {  
}
```

Проверяется, закончено ли воспроизведение mySound, и если это так, Flash выполняет то, что находится в фигурных скобках. Это событие может пригодиться, если, например, вам нужно сделать музыкальное вступление перед видео файлом.

В следующем упражнении мы создадим потоковый MP3-проигрыватель, показывающий различные видеоролики для каждого аудио трека. При этом мы используем событие onSoundComplete как переключатель, меняющий видеоролики.

Просто, как MP3

В этом упражнении мы динамически вызываем в наш Flash-фильм два музыкальных файла MP3. Если хотите следовать моему примеру в точности, эти два трека можно скачать на сайте www.friendsofed.com. Они называются `recordsback.mp3` и `nowewon.mp3`. «Весят» они 2 и 2,5 МВ соответственно, так что, если соединение у вас медленное, используйте свои MP3-файлы.

Здесь будет довольно много ActionScript, но мы все рассмотрим отдельными блоками.

1. Откройте новый фильм и импортируйте два видеоклипа, звук вместе с ними не импортируйте. Я использовал `cross_ramps.mov` и `halfpipe.mov`. Как обычно, не забудьте поместить каждый видеоролик в отдельный фильм-символ.
2. На основной монтажный стол вставьте два новых слоя — «video1» и «video2», и перетащите каждый фильм-символ в свой собственный слой.
3. Выделите кадр 1 слоя video 1 и перетащите его в кадр 2, затем выделите кадр 1 слоя video2 и перетащите в кадр 3. Ваш монтажный стол теперь должен выглядеть так:



Сейчас мы добавим ActionScript для динамического вызова в наш фильм MP3-файлов. Кроме того, мы зациклим каждый видеоклип до конца первого аудио трека, а затем, когда стартует следующий MP3, начнем воспроизводить второй видеоклип.

4. Переименуйте Слой 1 (Layer 1) в «actions», выделите кадр 1 и откройте панель Действия (Actions). Внимательно введите следующий код:

```
i=1;
songArray = new Array ();
songArray [0] = "recordback.mp3";
songArray [1] = "nowewon.mp3";
songCount = songArray.lenght;
```

Первым делом мы инициализируем переменную *i*. В ней будет храниться номер песни, проигрываемой в данный момент.

Далее мы создаем массив из названий песен. Скрипт во второй строке создает новый массив, две следующие строки заполняют ячейки массива названиями песен. Если вас удивляет, что наш список начинается с 0, а не с 1, ну... привыкайте, нумерация массивов всегда начинается с 0.

Массивы в чем-то схожи с переменными, они тоже хранят информацию, но не один элемент данных, а несколько. Можете сравнить массив со списком покупок, вы приходите в магазин и смотрите: «что у нас первое по списку?», «что у нас дальше по списку?»

А переменные хранят всего один элемент информации, как если бы вместо общего списка вы пришли в магазин с кучей отдельных бумажек, на каждой из которых записана одна покупка: что у меня там значится на желтоватой бумажке — нет, не на этой, на другой желтой бумажке!»

Вернемся к массиву в нашем скрипте. Вы можете добавить MP3-файлы, просто продолжая список. Если вам понадобился третий трек, просто вставьте строку:

```
songArray [2] = "myThirdTrack.mp3";
```

.. и так далее. Последняя строка кода сохраняет количество заполненных ячеек массива. В данном случае переменная `songCount` содержит число 2.

Как видите, в скрипте указаны имена файлов, используемых в нашем примере. Если вы используете собственные файлы, замените названия в скрипте именами своих файлов, заключенными в кавычки.

5. Добавьте в скрипт следующие две строки:

```
mySound = new Sound ();  
mySound.loadSound (songArray [i-1], true);
```

Первая строка нам знакома: инициализация объекта типа `Sound`. Во второй строке с помощью метода `loadSound` привлекается звуковой MP3-файл. Прежде мы использовали только `attachSound`, но, как ранее упоминалось, мы можем динамически вызывать внешние MP3 файлы для использования во Flash MX. Метод `loadSound` имеет следующие параметры:

```
mySound.loadSound (<URL /имя файла>, <флаг потоковой передачи>);
```

Если флаг потока имеет значение `true`, происходит потоковая передача MP3-файла в фильм, если же `false` — MP3 импортируется как звуковое событие. В данном упражнении мы будем использовать поток, поэтому в нашем коде установлен флаг `true`.

6. Следующая секция кода. Введите следующее:

```
mySound.onSoundComplete = function () {  
    i++;  
    if (i>songCount) {  
        i=1;  
    }  
}
```

Первую строку мы уже видели, здесь проверяется, завершено ли воспроизведение звука, при помощи события `onSoundComplete`. Если звукозапись закончилась, Flash выполняет инструкции, заключенные в фигурные скобки. Инструкции у нас таковы:

По окончании песни нам нужно увеличить нашу переменную-счетчик, `i`. Вторая строка кода — это сокращенный эквивалент выражения `i = i + 1`.

В третьей строке значение счетчика `i` проверяется и, если оно превышает реальное количество песен (хранящееся в `songCount`), то счетчик сбрасывается опять на 1, — таким образом, наш список MP3 закликивается, возвращаясь к первой песне.

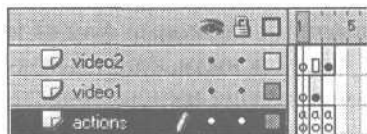
7. Теперь введите последнюю порцию кода:

```
mySound.loadSound (songArray [i-1], true);  
gotoAndPlay (i+1);  
};
```

Первая строчка — уже знакомая нам загрузка MP3-файла, используя информацию из списка `songArray`. Не удивляйтесь выражению `[i-1]`, — компьютер считает с 0, помните?

Вторая строка отсылает основной монтажный стол к следующему кадру, чтобы начать показ следующего видеоклипа. Здесь мы используем `i + 1`, поскольку наши клипы находятся в кадрах 2 и 3.

8. Последнее, что осталось здесь сделать — вставить действие `stop` в кадры 2 и 3 слоя `actions`:



9. Прежде, чем тестировать фильм, нужно сделать еще кое-что. Отйдите у себя на жестком диске MP3-файлы, используемые в этом упражнении, и поместите в ту папку, где находится FLA. Теперь сохраните фильм Flash как `mp3_player.fla` в той же папке, где находятся ваши MP3. Это важно — если они окажутся в разных папках, Flash не сможет их найти, когда вы будете проверять фильм.

Постарайтесь избежать употребления в именах файлов служебных символов (таких, как пробелы и слэши). При необходимости используйте знак подчеркивания или дефис, это поможет избежать ошибок в ActionScript.

10. А теперь протестируйте фильм.

Первый видеоклип воспроизводится циклически, пока проигрывается первый трек, а затем Flash одновременно запускает следующий трек и следующий видеоклип. Смена не самая плавная, но это из-за загрузки MP3-файлов. Если вы хотите синхронизировать звук с видео (то есть, сделать не просто MP3-проигрыватель), тогда лучше импортировать звуковой файл в Библиотеку и использовать метод `attachSound`, чтобы минимизировать паузу между треками.

Конечно, наше MP3 приложение не самое изощренное, но его относительно просто создать и очень легко содержать: не требуется много

труда, чтобы добавить в него больше звуковых треков и видеоклипов, и наше приложение может совершенно спокойно воспроизводить целую папку MP3-файлов. Если хотите обогатить его видео элементами, можно создать отдельные Flash-фильмы для каждого видеоклипа и, при помощи ActionScript загружать эти внешние SWF-файлы в основной фильм по мере надобности.

Еще можно использовать возможности ActionScript для расцвечивания или искажения видео изображений. Можно добавить несколько кнопок, чтобы пользователь мог «пролистать» треки или видеоклипы и найти то, что ему нужно. В общем, полная свобода творчества, экспериментируйте.

Заключение

В этой главе мы рассмотрели различные способы использования звука во Flash, узнали, как можно использовать внешние звуковые файлы и связать их с видео при помощи ActionScript. Мы выяснили, когда выгоднее динамически вызывать MP3-файлы, а когда лучше импортировать звуковое сопровождение вместе с видео. Мы узнали также, что импортирование звука во Flash отдельно от видео дает нам более полный контроль над звукозаписью; как в этом случае посредством ActionScript синхронизировать звук с монтажным столом видео.

Все это показывает многообразие творческих возможностей, открывающихся при использовании объекта Sound и управлении им посредством ActionScript, этот объект может быть использован в связке с видео, анимацией, или просто сам по себе.

Эта глава завершает первый и самый важный раздел нашей книги. Теперь, если вы внимательно изучили его, вы обладаете всеми навыками, необходимыми для создания Flash-презентаций. Однако не спешите закрыть книгу, следующий раздел — о том, как применить все эти навыки вместе и создать настоящий сетевой или локальный проект...

Раздел 2

Конкретные примеры

Конкретные примеры: планирование, изготовление и компоновка

Пора заняться «реальными» вещами. Итак, попробуем создать презентацию Flash, которую можно было бы представить в качестве небольшого сайта в сети, и, более того, поместить на CD-ROM. В следующих трех разделах мы рассмотрим, как спланировать, создать и экспортировать ролик Flash MX для парка Livingston Skatepark в солнечной Шотландии.



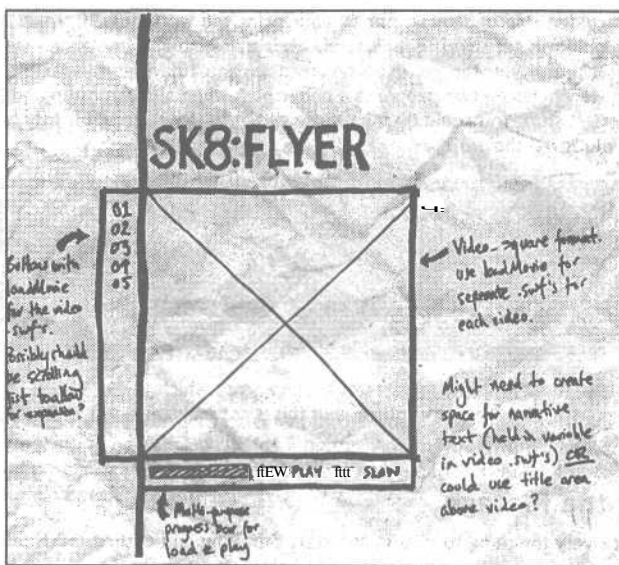
Сначала я создаю приблизительный набросок того, как ролик может выглядеть. При этом я не стремлюсь принять какие-либо окончательные решения, а решаю задачу, что в основном должен содержать ролик, чтобы заказчик был доволен. Я определяю, какие интерактивные элементы можно включить, и составляю грубый план создания ролика.

Разработанная таким образом стратегия дальнейшей работы может быть описана следующим набором правил:

- минимум дизайна, главную роль играет ролик;
- он должен содержать видеозапись катания на скэйте в парке;
- очень простенький текст, оформленный обычным полужирным шрифтом;
- парк представляет собой группу бетонированных дорожек, доставляющих людям немало неприятностей. Этот незначительный минус должен быть упомянут, не став при этом ложкой дегтя;
- это довольно оживленное место, в котором, как правило, много местного народа — это тоже должно быть упомянуто;
- звук очень важен, но нужно учесть, что обычный домашний компьютер обладает очень небольшим набором средств (вообще-то совсем не обладает) для создания звуковых элементов, аналогичных по качеству профессионально записанным;
- несмотря на то, что в ролике используется видео, мы хотим, чтобы он не «тормозил» даже при небольшой скорости передачи. Можно предложить пользователю в начале выбрать скорость передачи;
- видео, как правило, представляет собой линейную последовательность элементов. Это открывает новые возможности в отношении интерактивности;
- ролик должен полностью отображаться на экране в режиме 1024x768. Кроме того, было бы неплохо, если бы в CD-ROM версии он также «умещался» в режиме 800x600.

Некоторые из данных правил могут показаться очень грубыми, но они дают нам основу для дальнейшей работы.

Теперь нужно дождаться хорошего солнечного дня и направиться в этот парк, прихватив с собой цифровую видеокамеру и штатив. Штатив нужен для получения устойчивого изображения — это очень важно, т.к. позволит в дальнейшем без потерь существенно сократить размер файла.



Неудачный план или планирование неудач

Вообще-то перед походом в парк я бы порекомендовал набросать для себя список того, что вы собираетесь снимать, а затем просто пометать то, что уже сняли. Поверьте, вы будете не в восторге, если позже придется еще раз тащиться в парк с камерой, чтобы запечатлеть забытые, но обязательные вещи. Можно поступить иначе — определить, какого плана ролик вам требуется и затем просто ждать, пока он не поступит в продажу.

Со стилистической точки зрения я решил не вести съемку с самого скэйта, когда камера часто ныряет и делает крутые виражи. Не то, чтобы я против этого — совсем нет — просто быстро движущиеся объекты представляют «попотеть» модемы. Запомните — чем более статична картинка, тем выше качество видеозаписи, а значит и меньше размер файла.

В итоге я отснял материала примерно на час, вроде бы много, тем не менее, после отбора собираюсь собрать всего лишь пятиминутный ролик из кадров с действительно хорошим качеством. К тому же я поговорил с несколькими завсегдатаями о парке и запомнившихся им эффектных падениях скэйтбордистов. По разным причинам ни один из них не согласился на съемку нашей беседы — только катание. Так, мои планы запечатлеть заядлых мастеров, рассказывающих выразительные истории из своей практики, пришлось отложить до лучших времен.

В основе лежит золотое правило оператора — отнюдь не все, что вы сняли, будет выглядеть так, как вы задумывали. В кадр попадет куча абсолютно ненужных вещей, могут возникнуть проблемы с оборудованием, кто-то из людей будет стесняться. Но, с другой стороны, вам может попасться совсем незапланированный материал, неожиданные объекты и какие-нибудь сумасшедшие события, которые позволят заметно обогатить содержание ролика. (Представьте, вы снимаете диалог двух людей в парке. В этот момент в кадр незаметно попадает собака и внезапно бросается на одного из беседующих. Великолепно, не правда ли? Как минимум, пятьдесят баксов в любом из телевизионных шоу на такие темы вам обеспечены. А сколько еще разных сюрпризов можно встретить!)



Возникает проблема — из шестидесятиминутного материала я хочу сделать 10-15 клипов. В идеальном случае они бы следовали один за другим, но в действительности они перемешаны со скучными ненужными кадрами.

Просмотр отснятого материала

Сразу после возвращения из парка очень хочется тут же сесть за компьютер и заняться отбором и вырезанием кадров из имеющегося материала, т.к. у меня есть еще пара часов до намеченной встречи с друзьями. Казалось бы, времени хватит, и, тем не менее, придется остудить свой пыл — это не лучшая идея. На самом деле, файлы видео, как правило, очень велики по размеру, и работа с ними отнимает много времени. Тем более, если вы не применили сжатие файлов. Мы все-таки будем их сжимать при импортировании во Flash, стараясь при этом не понизить качество изображения. К примеру, 10-минутное цифровое видео, снятое при наилучшем разрешении и сжатии в формате DV (очень незначительное сжатие), займет на жестком диске около 1.6 гигабайт. Даже если у вас стоит винчестер громадной емкости, такие большие файлы сильно снизят производительность любого компьютера, за исключением самых «быстрых».



Так что же тогда я стану делать? Планировать. Я забываю о компьютере на какое-то время. Устраиваюсь поудобнее на диване с чашечкой хорошего кофе или чего-нибудь еще, положив рядом с собой бумагу, ручку и свою видеокамеру. Теперь я собираюсь прокрутить свой часовой шедевр и отметить начало и конец всех удачных кусков (на этой стадии большая точность не нужна). Для этого мне придется просмотреть запись несколько раз, после чего я уже достаточно хорошо буду знать отобранный материал. Советую заниматься этим до тех пор, пока запись не начнет надоедать, и вы не почувствуете, что знаете ее содержание наизусть.

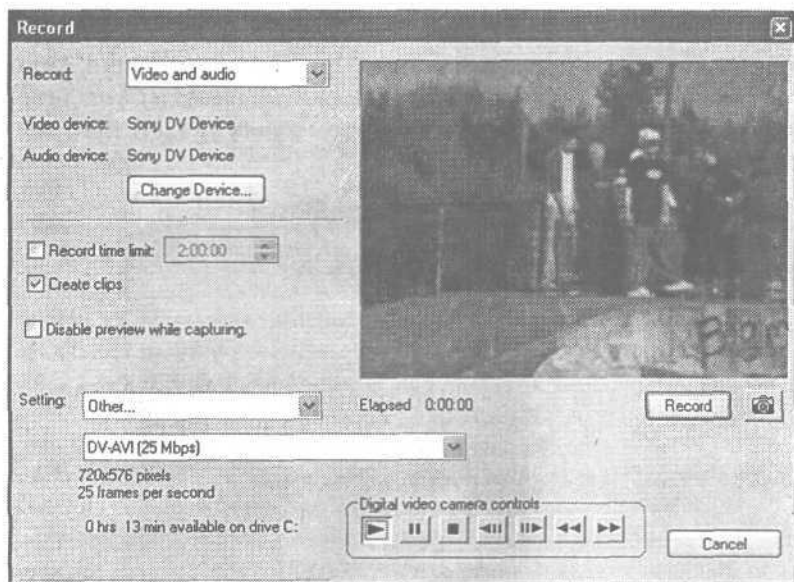


В итоге я отобрал девять клипов, которые достаточно полно отражают то, что парк может предложить своим посетителям. При этом появилась идея добавить интерактивные элементы. Я думаю, было бы неплохо включить возможность воспроизведения части ролика в замедленном действии, скажем, при нажатии кнопки. И, наверное, громкость звука во время такого показа нужно уменьшить. Однако вернемся к этому позже.

Скачивание видео

Теперь нужно ролик перенести с камеры на компьютер. Есть множество способов сделать это в зависимости от используемого приложения

и вида компьютера (Mac или PC). Я обычно для этого использую Moto DV, а затем обрабатываю запись в Cleaner 5. Это не самые дешевые программы, поэтому здесь я собираюсь использовать Windows Movie Maker, программу, поставляемую вместе с Windows XP.



Для скачивания лучше всего, конечно, использовать кабель FireWire. Это обеспечит наилучшее качество переноса, да и стоит такой кабель не очень дорого (если, разумеется, он не входит в комплектацию компьютера).

К счастью, в звуковой плате моего компьютера есть порт FireWire, с его помощью я подключаю цифровую камеру и нажимаю на ней кнопку воспроизведения записи. Через пару секунд появляется диалоговое окно, в котором мне предлагается переписать видео с использованием Windows Movie Maker. При нажатии кнопки ОК открывается окно записи с несколькими параметрами. Я хочу получить изображение наилучшего качества из всех возможных, поэтому в меню Параметры (Settings) я из списка выбираю Другое (Other), после чего появляется еще один список. Из него я выбираю DV-AVI (25 Мб/с).

При таком выборе видео переносится с незначительным сжатием с использованием кодека DV — несжатый файл в данном формате будет иметь размер в 4 раза больше. Учитывая то, что вы используете домашний компьютер, графические возможности которого отнюдь не запредельны, с такими большими файлами работать очень сложно (читай, долго).

В нижней части окна Windows Maker есть несколько голубых кнопок, управляющих процессом воспроизведения ролика камерой. Это позволяет перематывать запись в начало каждого клипа, которые я отобрал, лежа на диване. Именно на этом этапе и становится понятно, что я не зря потратил время на многократный просмотр своего шедевра. Для уверенности можно захватывать несколько кадров до начала каждого куска и несколько кадров после его конца. При нажатии кнопки Записать (Record) скачивание начинается, а при нажатии кнопки Стоп (Stop) открывается окно сохранения файла, в котором я могу указать программе, куда сохранить файл видео.



Теперь, после того как весь файл перенесен на компьютер, я сталкиваюсь с редактором временной шкалы. Можно слегка обработать изображение, например, обрезать лишние кадры в начале и конце каждого куска или удалить неудавшиеся кадры внутри них. По умолчанию Windows Movie Maker разбивает весь ролик на «клипы» так, как будто между ними нет связи, но вас это не должно смущать. Это только условное разбиение, я все еще работаю с одним файлом. Я щелкаю на каждом клипе и проигрываю его в окне предварительного просмотра. При этом прихожу к вы-

воду, что первый из них и два последних можно удалить, а затем щелкнув на первом из отобранных клипов и, нажав на клавишу Shift, — на последнем. Теперь можно все выделенные таким образом клипы перенести на временную шкалу, расположенную в нижней части экрана.

Итак, для создания файла, готового для импорта во Flash, нужно всего лишь выполнить следующие шаги.

1. В меню Файл(File) выберите пункт Сохранить фильм (Save Movie).
2. Установите Качество воспроизведения (Playback Quality) на Другое (Other), выберите из списка 'DV-AVI PAL (25 Mbps)' (или NTSC, если в вашей стране используется этот формат).
3. Нажмите ОК.
4. Задайте название файлу с новым клипом и нажмите кнопку Сохранить (Save).

Я проделал такую процедуру со всеми девятью клипами. Можете не волноваться по поводу того, что видео в таком виде при воспроизведении на компьютере кажется нестабильным. Оно еще не преобразовано к виду, наиболее подходящему для воспроизведения, и ваша машина будет изо всех сил стараться проигрывать запись при наилучшем разрешении в формате DV, пытаясь в то же время не уменьшать частоту смены кадров. Во Flash все проблемы будут решены.

В соответствии с наброском, сделанным в предыдущих разделах, я решил задать изображению квадратный формат. Вообще-то размеры кадра определяются производительностью процессора, необходимым размером файла и доступной областью экрана. Поскольку в нашем случае видео должно быть в центре внимания, я решил выделить сравнительно большую часть экрана размером 360x360 пикселей.

Теперь исходное изображение в формате 720x576 (PAL) нужно преобразовать к выбранному виду. Для этого я использую Adobe After Effects, так как эта программа у меня уже есть и с ней довольно просто обращаться. Однако для этого можно использовать любое другое подходящее приложение, причем некоторые из них намного дешевле других.

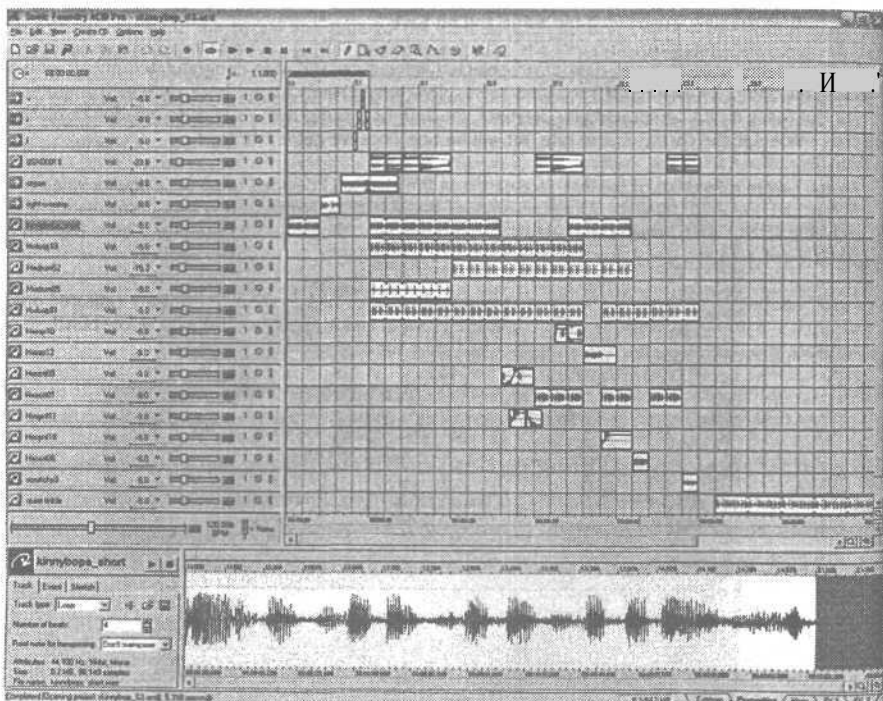
Вообще говоря, лучше выбрать по ширине и высоте такие размеры, чтобы они были кратны четырем, так как кодек Sorenson Spark преобразует видео к виду 4x4 сегмента. То есть, например, если вы задали размеры 157x157, то кодек автоматически добавит по 3 пикселя к ширине и высоте кадра так, чтобы можно было произвести разбиение на сегменты. При этом добавленные области во Flash не будут видны, хотя размер файла немного увеличится.

Настройка

Перед запуском Flash теперь осталось сделать еще одну вещь. Мне могут понадобиться звуковые сигналы, поэтому я создаю следующий список:

- спокойный фоновый звук с тяжелыми басами;
- щелчки, звуки пиканья для анимации текста;
- тяжелый и медленный ритм ударных для основной части сайта;
- скрипучий металлический звуковой сигнал для кнопок;
- тяжелый и громкий (кричащий) ритм ударных для видео;
- более «спокойный» вариант этого ритма для замедленного показа.

Можно, конечно, добавить еще несколько звуковых сигналов, но для начала этого вполне достаточно. При этом лучше начинать создавать звуковые эффекты, уже зная, для каких целей. Вы, наверное, заметили, что мой язык оставляет желать лучшего в описании сигналов. Просто здесь более важно точное понимание того, о каком именно звуке идет речь. Если я его называю, опираясь на то, как он звучит, то другому человеку (если проект разрабатывается группой людей) будет гораздо легче меня понять.



У меня есть большая коллекция звуковых сигналов, собранных из разных дисков и бесплатных файлов, скачанных из сети. Есть множество сайтов, которые бесплатно предлагают различные звуковые эффекты и музыку, но будьте осторожны — если сайт позволяет бесплатно скачать что-то, то это вовсе не означает, что у него обязательно есть разрешение на распространение. Используя такой звук в своем проекте, вы можете понести ответственность за его использование. Так что лучше все-таки семь раз отмерить, а уж потом резать.

Другой вариант — приобрести подходящий микрофон и софт для работы со звуком. Почему бы самому не создать все эффекты? Сколько душе угодно. Я использую Sound Forge для записи и обработки сигналов, и Acid Pro — для создания музыки. Демо-версии обоих программ можно получить на сайте компании Sonic Foundry www.sonicfoundry.com/downloads. Более подробные сведения о работе со звуком можно найти в главах 6 и 7.

Как правило, создание звуковых эффектов происходит не строго по намеченному плану, и они могут звучать не в точности так, как хотелось бы, но это только на данном этапе работы.

Заключение

Возможно, вас данная глава немного разочаровала, так как вы хотели с самого начала наброситься на Flash, минуя все эту спланированную последовательность действий.

Можно поступить и так, если вы располагаете массой свободного времени и достаточными средствами. Все эти этапы позволят просто сбересть и то и другое.

Спланируйте, на какую тему разрабатывается проект, и какой цели нужно достичь. Затем определитесь с тем, какое содержание для этого потребуется. Подумайте, нужно ли на самом деле видео, не будет ли оно бесполезным приложением. Составьте список необходимых вещей, включая текст, видео, картинки, музыку, звуковые и голосовые эффекты.

Может показаться затруднительным создать такой подробный план в самом начале работы. Смысл в том, что он может быть и не очень точным, просто вы создаете основу, опираясь на которую можно продолжать разработку. Теперь, когда мы все это проделали, можно переходить к следующей стадии.

Внутри Flash

Краткое содержание главы:

- как организовать фильм Flash, используя **абстракцию**;
- как импортировать видео во Flash;
- как использовать ActionScript для создания **динамического** видеопроигрывателя Flash.

И вот, наконец, наступил момент, которого вы так ждали — пора открывать Flash и начинать обработку. К сожалению, должен вас разочаровать, все еще рано. Перед этим мы снова должны взять ручку с бумагой и подумать о структуре проекта.

Есть масса различных способов структурирования проекта, и выбор, зачастую, определяется индивидуальными предпочтениями и уровнем владения Flash. Например, я могу пойти двумя путями, чтобы достичь одинакового результата. Остается определить, какой из них лучше, и в этом лучше всего опираться на собственный опыт.

Что я имею в виду? Возможно, вы новичок во Flash и не имеете за спиной **опыта**, поэтому попробую объяснить. Самым очевидным критерием выбора является стремление получить один большой файл Flash, содержащий группу клипов. В этом случае планирование оказывается минимальным, и именно поэтому многие решают идти по данному пути. Однако и он имеет свои трудности. Так что, если вы не обладаете опытом работы во Flash, можете позаимствовать мой.

Ошибок не совершает только тот, кто ничего не делает. Поэтому любой опытный человек совершил их немало. В этом смысле я себя могу считать достаточно искушенным, и одним из трудных уроков, которые я вынес, является стремление представлять проекты в виде совокупности модулей, насколько это возможно. Такое разбиение позволяет впоследствии легко модифицировать и расширять содержание проектов. По мере завершения работы у вас все больше будет возникать желание не-

много изменить имеющиеся вещи или добавить что-нибудь новое. Своясь с первоначальным планом, вы заметите, насколько в итоге изменился проект.

Для примера представьте такую ситуацию. Через полгода после создания проекта заказчик, то есть владелец парка, снабжает меня обильным новым съемочным материалом с просьбой обновить рекламный ролик. Я, разумеется, располагаю небольшими средствами для этого, поэтому начинаю искать способы сделать это как можно эффективнее. Я снова открываю свой файл Flash, который я не трогал все это время. На открытие файла мой бедный компьютер тратит добрых десять минут, а затем его скорость падает до скорости торопящейся на обед улитки. А все из-за того, что видео увеличило файл до гигантских размеров. Каждый раз при экспорте его в сеть на это также будет уходить порядка десяти минут, так как будет происходить сжатие всех звуковых элементов (включая саундтреки). Это может стать просто ночным кошмаром, от которого не можешь проснуться. Поэтому мы пойдем другим путем.

Мы воспользуемся концепцией, которую я называю «абстракцией».

Абстракция

Легендарный джазовый басист, Чарльз Мингус, как-то сказал: «Делать из простых вещей сложные легко, делать из сложных вещей простые ужасно легко, это и есть творчество».

Смысл абстракции состоит в упрощении. Храните свои файлы в как можно более простом виде. Это единственное и главное правило, которое вы должны усвоить для эффективной работы с Flash. Может показаться, что с большими файлами видео следовать абстракции трудно. Тем не менее, Flash MX позволяет сделать это.

Для этого нужно на клипах поместить кнопки, нажатие которых запускает другие клипы. Таким образом, можно создать иерархическую последовательность из клипов во временной шкале, которые связаны между собой одним программным кодом ActionScript, но сами в свою очередь хранятся раздельно. То есть вы как бы пишете сценарий с помощью ActionScript, а он запускает последовательность независимых клипов. Такая абстракция позволяет легко обновить ролик целиком, для этого нужно просто вставить новый кусок в нужное место или заменить им прежний.

Поэтому старайтесь разбить проект как можно больше. Возможно, это вас удивит, но в большинстве сложных сайтов, в разработке которых я принимал участие, к файлам Flash применялась именно такая стратегия их упрощения — из простых элементов разными хитроумными способами комбинировались сложные интересные вещи.

Только, пожалуйста, не нужно паниковать по поводу того, что вся эта структуризация файлов вам не по зубам. Абстракция это довольно простой прием, для понимания которого нужно совсем немного времени. Зато, усвоив его, вы в дальнейшем сэкономите гораздо больше этого же драгоценного времени и других средств.

Структура Flash

Теперь попробуем всю эту теорию применить для создания рекламного ролика парка. Если не применять планирование структуры проекта, то мы придем к наиболее очевидной прямолинейной последовательности, показанной на рисунке



Такая простая структура будет прекрасно работать. Все элементы ролика будут храниться в одном файле, может быть, разбитом на сцены. Вы поместите на экран кнопки, позволяющие переходить на другие клипы. Проблемы начнутся, когда число клипов станет большим: размер и сложность файла увеличатся, а эффективность вашей работы уменьшится.

Я не буду здесь описывать, как создать такую структуру. Есть множество других хороших книг о Flash, в которых это можно узнать. Я предлагаю вам пойти вместе со мной по другому пути: просто немного подумать перед принятием решения и постараться сэкономить время, деньги и нервы. Вы ведь не застрахованы от того, что примерно за неделю до завершения вами работы не заявится заказчик с просьбой вставить в проект еще 200 новых роликов?

Центральным элементом рекламного ролика являются видео клипы и интерфейс для их выбора и просмотра. Кроме того, я хочу создать анимацию вводной части и сделать список разработчиков проекта. Вот и все.

Применение абстракции к элементам проекта очень важно, если мы хотим сделать его доступным для различных преобразований. Ролик должен позволять добавление к нему новых клипов в будущем, причем, в достаточно большом количестве. Мы можем начать планирование с шести клипов, но их число может вырасти впоследствии до шестидесяти, так что мы должны это учитывать.

У рассмотренной нами простой структуры есть один недостаток: если кому-то захочется просмотреть сразу шестой клип, ему придется подождать

дать, пока не загрузятся предыдущие пять. А поскольку при работе с файлами видео скорость компьютера заметно падает, то такое ожидание может быть довольно утомительным. Теперь представьте, что будет, если есть 60 клипов, и нужно просмотреть только последний из них. Весело, не правда ли?

Пора начинать

Во Flash можно с одного фильма загружать другие. Это не так сложно, как может показаться с первого раза. Просто есть какой-то «базовый» файл, содержащий код и кнопки интерфейса для выбора и запуска остальных клипов, но не сами эти клипы.

Откройте файл *video_base.fl*a (содержащийся в группе файлов к данной главе) с примером использования описанного способа.

На что вы обратили внимание, прежде всего? Во временной шкале находится только одна сцена с одним кадром. Это может идти вразрез с вашим прошлым опытом работы или с тем, чему вы научились из других книг, но по мере продвижения в сторону объектно-ориентированного дизайна и программирования во Flash MX (более подробно об этом говорится в главе 11) все станет на свои места. Оставайтесь с нами, и мы покажем вам звезды.

Но достаточно теоретизировать по поводу абстрагирования ролика для парка. Например, кнопки мы преобразуем к активным кнопкам Flash, так чтобы подключенные к ним клипы автоматически убирались с экрана и наследовали свойства с функций, записанных в исходной временной шкале.

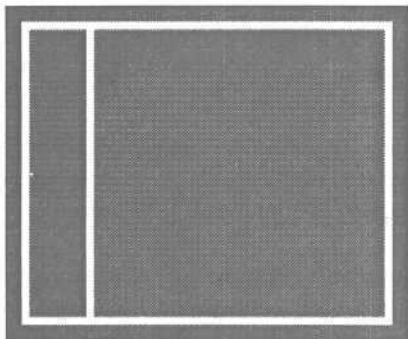
Подготовка к видео

Итак, что у нас есть в этом файле? Две рамки, одна из которых пустая, а вторая содержит пару кнопок с надписями Видео 1 и Видео 2. При размещении в сети я создаю файл SWF, размер которого не будет превышать 1 Кб. И это не удивительно, если учесть его простоту. Однако простой не значит бессодержательный. Давайте вместе пройдем стадии создания файла *video_base.fl*a, обращаясь, когда это потребуется, к оригиналу.

1. Откройте новый фильм Flash, назовите его *video_base.fl*a и сохраните.
2. Добавьте два новых слоя и задайте всем трем названия (по порядку) *video*, *buttons* и *boxes*.

На слое *boxes* размещены рамки, создающие фон для будущего проигрывателя. Левая колонка будет содержать кнопки выбора видеоклипов, а в правой будет происходить воспроизведение выбранного клипа.

3. Нарисуйте свои рамки или скопируйте наши. Толщина рамок 10 пикселей, поэтому если формат изображения выбран 360x360, то правая рамка будет иметь размеры 380x380.



4. На слое *buttons* мы создадим две кнопки. Во-первых, нарисуйте кнопку для Видео 1. Можете при этом не ограничивать свою фантазию, однако мы выбрали просто слово «VIDEO 1».
5. Щелкните на кнопке и нажмите клавишу F8 для преобразования ее в символ. В появившемся окне введите название «vid1But» и в поле Поведение установите кнопку выбора Кнопка.
6. Вернитесь на временную шкалу, выберите кнопку VIDEO 1 и на панели Действия (Actions) введите следующий код:

```
on (release) {  
    loadMovie ("video1.swf", "video_mc");  
}
```

Этот код означает, что когда кто-то щелкнет на созданной нами кнопке, Flash загрузит для воспроизведения файл `video1.swf` в области для клипов, которую мы назовем `video_mc`.

7. Сделайте копию символа `vid1But`, назвав его «`vid2But`».
8. Придайте кнопке желаемый вид и перенесите ее под кнопку VIDEO 2 в первой рамке.

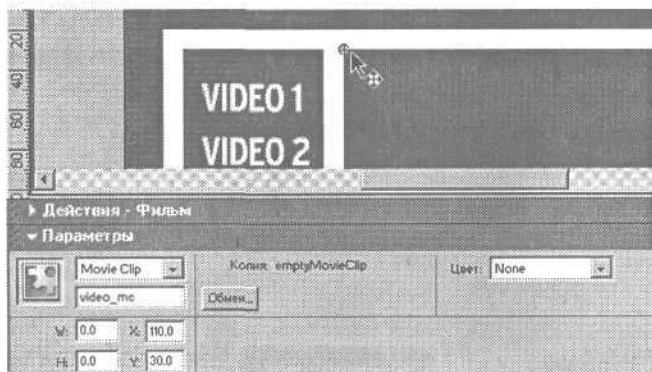


9. Выделите символ `vid2But`, откройте панель Действия и введите следующий код ActionScript:

```
on (release) {  
    loadMovie ("video2.swf", "video_mc");  
}
```

Как видно, в коде изменилось только название файла SWF, который также будет воспроизводиться в области `video_mc`. Давайте создадим ее.

10. Создайте новый клип, назовите его «`emptyMovieClip`», ничегок нему не подключая. Совсем ничего.
11. Вернитесь к основной временной шкале и перенесите копию `emptyMovieClip` в область, соответствующую слою `video`. Подгоните верхний левый угол с таким же внутренним углом правой рамки. На панели Параметры (Property) задайте название соответствующей копии `video_mc`.

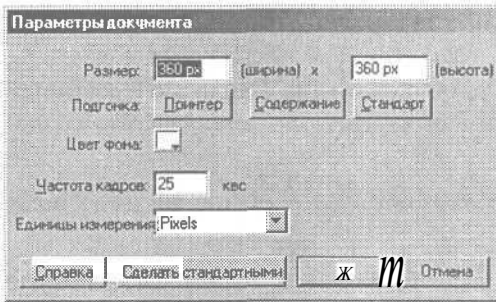


Этим указанием угла пустого клипа мы определили область, в которой будет воспроизводиться видео. Теперь нужно создать наши видео-файлы. Ранее мы задали размеры пустой рамки, поэтому формат изображения в файлах должен быть таким же, то есть 360x360.

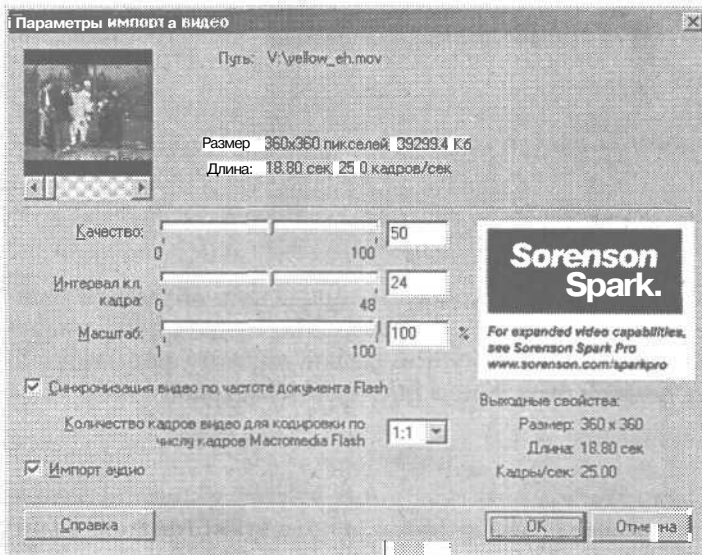
Сборка

Мы уже приготовили клипы, подогнав в них высоту и ширину кадра, осталось их импортировать. Размер файла, используемого мною в примере, очень велик, поэтому можете воспользоваться своим, не таким большим, или скачать мой в Библиотеке `video1 fla`.

1. Создайте новый файл Flash, укажите размеры 360x360 и частоту 25 кадров в секунду или любую другую, соответствующую частоте вашего клипа.

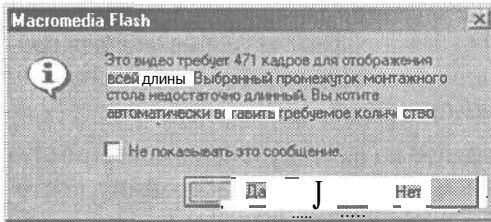


2. Сохраните файл под названием video 1.flav в той же папке, где находится файл video_base.swf.
3. Выделите кадр 1 и щелкните на пункте Импорт в меню Файл. Выберите видеофайл и в открывшемся диалоговом окне щелкните на команде Внедрить видео в документ Flash (Embed video in Macromedia Flash document).

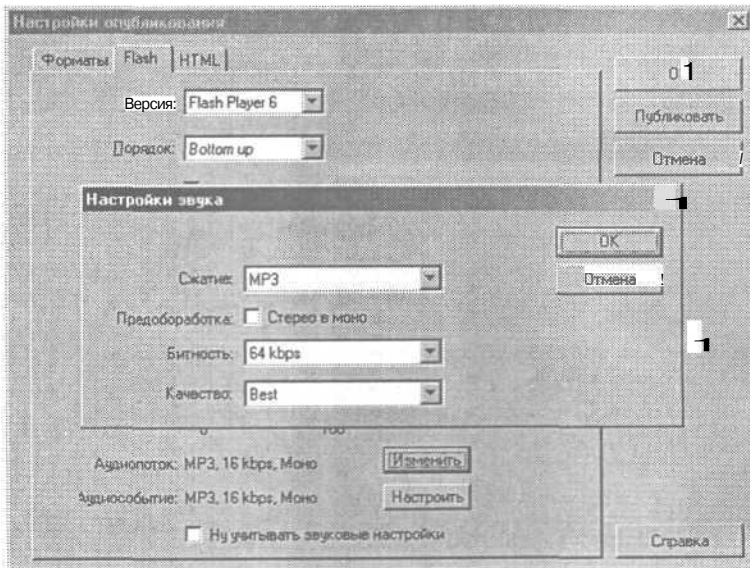


4. Все остальные параметры в окне Параметры импорта видео (Import Video Settings) оставьте по умолчанию и нажмите ОК.

Помните, что во Flash в данное время используется кодек Sorenson Spark для сжатия исходного файла в новый формат видео. Это преобразование может занять немало времени, а вы сходите за чашечкой чая и вооружитесь каким-нибудь журналом, пока не увидите такое диалоговое окно:



5. Нажмите кнопку Да, при этом откроется первый кадр временной шкалы. Если он смещен, установите правильные координаты на панели Свойства, также можете установить ширину и высоту кадра.
6. Осталось установить звук. В меню Файл выберите пункт Общие параметры (Publish Settings) и в открывшемся окне перейдите на вкладку Flash. В поле Аудиопоток (Audio Stream) указаны текущие параметры звука, для их изменения нажмите кнопку Изменить.



7. Установите показанные на рисунке значения, помня при этом, что чем выше качество, тем больше размер файла и тем больше времени

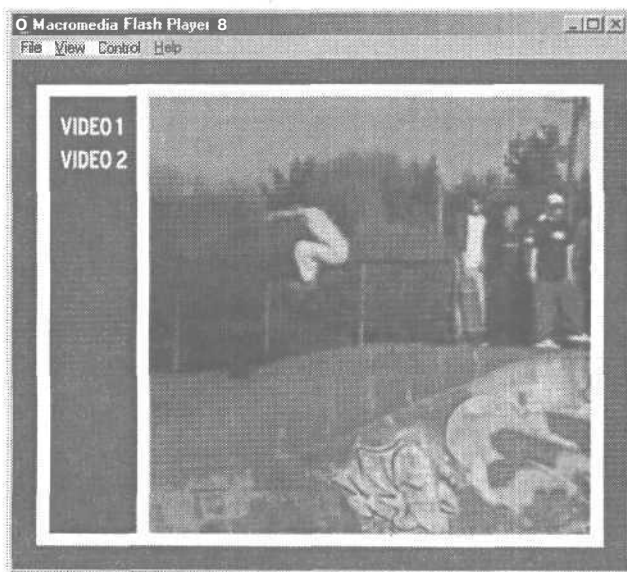
потребуется для его загрузки. Мы установим параметры MP3, 64 kbps, Stereo. При импорте Flash преобразует видео, но не звук, сжатие саундтрека произойдет при экспорте SWF файла.

Имейте в виду, что выбор параметра Stereo для звука, записанного в формате MP3, не увеличивает размер файла, так как при таком сжатии просто сохраняется различное звучание для правого и левого каналов. Учитывая то, что, как правило, в видеофайлах записывается одинаковый звук для обоих каналов, использование этой возможности при преобразовании может быть очень полезным.

8. В окне Общие параметры перейдите на вкладку Форматы и снимите флажок HTML. Мы не собираемся создавать HTML-страницу, состоящую только из нашего ролика, поэтому все, что нам нужно — это файл в формате SWF.
9. Теперь можно публиковать фильм.

Мы имеем в одной папке два файла: video1.swf, который вы только что создали, и video_base.swf с интерфейсом, состоящим из двух рамок, одна из которых содержит две кнопки, а другая пустая. Вообще-то в этом файле еще есть фильм, но он не виден, так как является пустым.

10. Итак, посмотрим, что получилось. Нажмите кнопку VIDEO 1. При этом пустой клип video_mc будет заменен нашим файлом video1.swf, и начнется его воспроизведение.



Такой способ загрузки внешнего файла в уже существующий так называемый базовый фильм является не единственным при подключении видео. Однако он оказывается наиболее эффективным при разработке больших проектов, так как позволяет разбить их на более простые составные части, которые можно хранить отдельно. Данный подход дает большие преимущества в тех случаях, когда над одним проектом работает несколько человек, и каждый создает отдельную его часть.

Теперь нажмите кнопку VIDEO 2 и убедитесь, что при этом видео отключается. Проигрыватель пытается выполнить нашу команду, то есть загрузить файл `video2.swf`, который мы еще не создали. Это будет вашим следующим заданием — составить файлы SWF (с кнопками) для каждого фильма, который мы планируем использовать.

Надеюсь, вы уже поняли, насколько выгодно применение абстракции при планировании. Мы можем предоставить пользователю возможность самому выбирать для просмотра фильмы из тысяч, имеющихся в списке, и самому устанавливать порядок их воспроизведения, а не следовать строго назначенному нами порядку в файле Flash.

Смысл абстракции заключается в том, что мы разработали содержание (клипы) отвлеченно от интерфейса (кнопки управления), то есть получили структуру в виде отдельных модулей. Итак, мы получили основу, так сказать, прототип нужного нам рекламного ролика.

Бразды правления

Давайте добавим немного функциональности. Предоставим пользователю средства управления, которыми обладают стандартные проигрыватели:

- перемотать вперед;
- перемотать назад;
- воспроизвести;
- остановить.

Самый очевидный способ сделать это — просто поместить соответствующие кнопки на каждый используемый клип. Это хорошая идея, но не лучшая. Возникает уже известная нам проблема: при увеличении числа клипов объем работы становится пугающим.

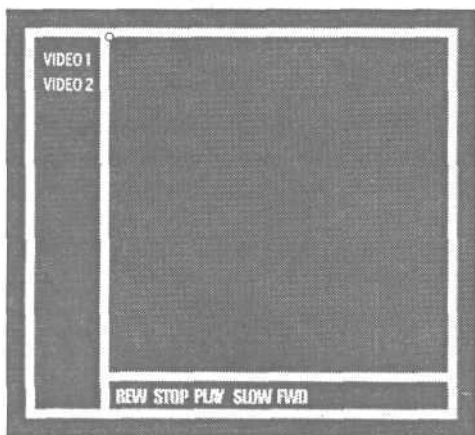
Но сущий кошмар начнется позже, когда после просмотра проекта я решу, что нужно добавить возможность замедленного воспроизведения. Это значит, что мне придется с самого начала просмотреть каждый клип, файл за файлом, подключая новую опцию и каждый раз повторяя экспорт.

К счастью, теперь мы знаем прием, как достичь цели без всей этой рутины. Наш помощник — абстракция. Видеоклипы, по сути, для всего ролика являются переменными величинами. Очень может быть, что многие из них будут добавлены позже, то есть после создания проекта, и экспорт каждого из них будет отнимать определенное время в зависимости от мощности компьютера. Поэтому постараемся сделать так, чтобы они экспортировались только один раз, даже если впоследствии понадобится усовершенствовать управление роликом.

Таким образом, план наших дальнейших действий таков. Непосредственно видеофайлы оставляем без изменений в их первоначальной невинности. Все кнопки управления размещаем на уже известном нам базовом фильме. Если нужно добавить или удалить какой-то из параметров управления, то нам придется сделать это всего один раз с файлом незначительного размера. Следовательно, его экспорт займет намного меньше времени.

Начало и остановка воспроизведения

Перед тем как набрасываться на работу, откройте файл `video_base_2.swf` и пощелкайте кнопками, имитирующими управление обычным проигрывателем. Примерно то же самое мы должны получить в итоге.



1. Откройте файл `video_base.fla` и измените слой `boxes`, как показано на рисунке. Не забудьте, что под видео должно по-прежнему оставаться пространство размером `360x360`.
2. Создайте пять новых кнопок и поместите их на кадр 1 слоя `buttons`. Их внешний вид, как и в случае с кнопками выбора клипа, я решил сделать максимально скромным — прямоугольник с текстом.

3. Щелкните на кнопке **PLAY** и введите на панели Действия следующий код:

```
on(release){  
    video_mc.play();  
}
```

Первая строка указывает Flash, что когда кто-то щелкнет на данной кнопке и затем отпустит кнопку мыши, должно что-то произойти. Вторая строка уточняет, что именно должно случиться. Давайте разберем эту строчку по частям.

Первая часть, `video_mc`, подсказывает Flash, что я обращаюсь к объекту видео (вы, наверняка, помните, что мы дали такое название «пустому» клипу, вместо которого будут загружаться конкретные клипы).

Следующая часть — это точка, говорящая о том, что все, за ней следующее, относится непосредственно к `video_mc`. Можно сказать, что точка выступает в роли связующего элемента в коде, хотя на самом деле все немного сложнее.

Третья часть состоит из `play()`. То есть событие, которое должно произойти, это воспроизведение `video_mc`.

Можно прочесть строку наоборот, и получится «play video», то есть именно то, чего мы хотим от данной кнопки. Как видите, все очень просто. Смысл в том, что мы дали копии пустого клипа название `video_mc` и можем оперировать с ним по своему усмотрению.

4. Теперь по аналогии подключим к кнопке **STOP** такой код:

```
on(release){  
    video_mc.stop();  
}
```

Как видите, все по-прежнему так же просто, но вместо воспроизведения `video_mc` мы даем команду его остановки.

Перемотка вперед, назад и замедленный показ

Теперь, когда у нас есть сравнительно простые кнопки начала и остановки воспроизведения, попробуем создать их более интересных собратьев.

На первый взгляд, к кнопке **FWD** в файле `video_base_2 fla` подключен гораздо более сложный код, нежели к **PLAY** и **STOP**. Но это только на первый взгляд, он намного проще, чем кажется.

Главное отличие состоит в том, что начало и остановка проигрывания являются мгновенными событиями, то есть выполняющимися немедленно после щелчка мыши. С кнопками перемотки дело обстоит ина-

че, подключенное к ним действие должно выполняться до тех пор, пока кнопка находится в нажатом состоянии, и прекращаться, когда пользователь отпускает ее. Поэтому наш код будет состоять из двух частей.

1. Первая часть отвечает за быструю перемотку вперед, когда пользователь нажимает кнопку. Щелкните на кнопке FWD и на панели Действия введите следующий код:

```
on (press) {
    video_mc.onEnterFrame = function () {
        video_mc.gotoAndStop (video_mc._currentframe + 5);
    };
}
```

Давайте посмотрим, что мы написали. Говоря техническим языком, мы воспользовались функцией обратного вызова. Постараюсь вкратце объяснить, что это значит.

Первая строка говорит о том, что должно что-то произойти, когда пользователь нажимает и удерживает кнопку мыши.

С помощью второй строки мы подключаем к клипу «функцию». Функция представляет собой составную часть программного кода, позволяющую увеличить функциональность (несмотря на тавтологию), в данном случае, перематывать видео вперед. Обратите внимание, что она выполняется под значением `onEnterFrame`, то есть применяется к каждому кадру нашего фильма.

В следующей строке мы записали непосредственно функцию, то есть само выполняемое действие. Выражение `video_mc.gotoAndStop` означает: перейти в клипе на определенный кадр и остановить показ. Далее записано `(video_mc._currentframe + 5)`, этим мы указали, что нужно перейти на пять кадров вперед, то есть к текущему кадру прибавили 5. Таким образом, мы симитировали ускоренную перемотку.

2. К введенному коду добавим следующий, прекращающий перемотку, когда кнопку отпускается:

```
on (release) {
    video_mc.onEnterFrame = null;
    video_mc.play ();
}
```

Как видите, этот код намного проще. В первой строке мы указали, что должно произойти определенное событие в тот момент, когда пользователь отпускает кнопку мыши.

Вторая строка дает команду окончания выполнения функции перемотки.

Последняя строчка вам уже известна из кода для кнопки PLAY. Она вновь запускает воспроизведение фильма в обычном режиме.

3. Теперь написать код для кнопки RWD будет легко:

```
on (press) {
    video_mc.onEnterFrame = function () {
        video_mc.gotoAndStop (video_mc._currentframe - 5);
    };
}

on (release) {
    video_mc.onEnterFrame = null;
    video_mc.play ();
}
```

Никаких проблем, не правда ли? Единственное отличие, как и следовало ожидать, в том, что теперь при выполнении функции перемотки осуществляется переход с текущего положения на пять кадров назад.

4. И, наконец, кнопка SLOW. Мы хотим, чтобы воспроизведение осуществлялось в три раза медленнее, чем в нормальном режиме, пока пользователь удерживает левую кнопку мыши нажатой и изучает новый для себя трюк асов из парка. На панели Действия добавьте к кнопке SLOW следующий код:

```
on (press) {
    video_mc.stop ();
    counter = 0;
    video_mc.onEnterFrame = function () {
        counter ++;
        if ((counter % 3) == 0) {
            video_mc.nextFrame();
        };
    };
}

on (release) {
    video_mc.onEnterFrame = null;
    video_mc.play();
}
```

Некоторые операторы вам уже знакомы. Мы по-прежнему указываем, что функция должна выполняться в течение того времени, пока кнопка нажата; как только кнопка отпускается, применение функции прекращается. Фактически, команда остановки выполнения функции записана так же, как и для кнопок FWD и REW.

Код, выполняемый при нажатом положении кнопки, немного сложнее. Он начинается так же:

```
on (press) {
```

Как и раньше, мы хотим, чтобы при нажатии и удерживании кнопки происходило определенное событие.

```
video_mc.stop ();
```

Этим событием является остановка воспроизведения, поэтому соответствующая команда аналогична коду для кнопки STOP.

```
counter = 0;
```

В следующей строке мы объявляем переменную counter и присваиваем ей значение 0. Говоря проще, мы создали счетчик и установили его в начальное (нулевое) положение.

```
video_mc.onEnterFrame = function (){
```

Этой строчкой мы указали, что, как и в случае с кнопками FWD и REW, к каждому кадру клипа будет применяться функция

```
counter ++;
```

На этот раз функция состоит более чем из одной строки. Первая увеличивает значение счетчика на единицу, при каждом переходе к следующему кадру фильма.

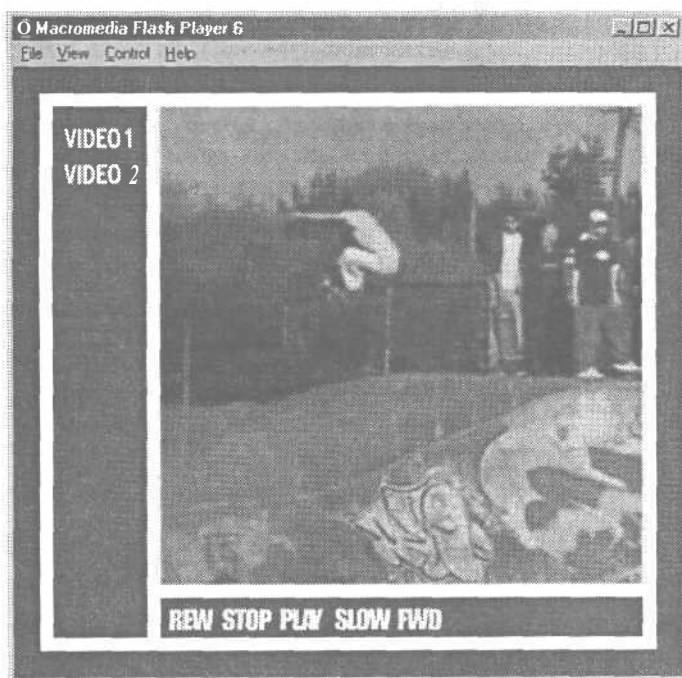
```
if ((counter % 3) == 0) {  
    video_mc.nextFrame();
```

Следующие две строки можно перевести на обычный язык примерно так: «переходить к следующему кадру через каждые три кадра». То есть проигрыватель отсчитывает по три кадра и показывает следующий. При этом скорость воспроизведения уменьшается в три раза, если вы замените тройку десяткой, то она упадет в десять раз.

Если вам интересно знать более подробно, как осуществляется замедленный показ, то можно дать такое объяснение. С каждым следующим кадром значение счетчика возрастает на единицу, принимая значения 1, затем 2, затем 3 и так далее. После увеличения вычисляется число $counter \% 3$.

Дело в том, что хорошо известный нам знак процента в мире ActionScript живет двойной жизнью. Вторая, новая для нас, его сущность выражается в том, что он исполняет роль модуля. То есть написанием данного символа мы даем команду вычислить остаток от деления одного числа на другое. Например, выражение $5 \% 3$ означает нахождение остатка от деления 5 на 3, результат — 2. Таким образом, в нашей функции производится переход от текущего кадра к следующему каждый раз, когда остаток от деления его номера на 3 равен нулю. Это будет происходить с каждым третьим кадром (т.е. 3, 6, 9, 12 и т.д.). Это самый эффективный способ решения подобного рода задач.

Теперь можете экспортировать фильм, открыть файл SWF и наслаждаться плодами своих нелегких трудов. Проект почти завершен...



Заключение

Многое из написанного в этой главе может показаться странным и непохожим на ранее известные вам приемы программирования. Я старался показать, насколько ActionScript просто и логично позволяет достичь цели. Во Flash есть множество способов решения задач, и выбранный нами далеко не всегда оказывается оптимальным.

Если вы приучитесь планировать проекты и как можно больше упрощать их с помощью абстракции, то вы сможете обогащать их содержание до каких угодно объемов без особых проблем. Вам придется просто пристроить к имеющемуся зданию проекта новые блоки (даже если таких блоков много).

Может быть, данная глава получилась излишне технической, но сравните полученные нами в итоге файлы с тем, что вы делали во Flash ранее. То, что вас пугает сложностью, на самом деле отличается простотой и строгостью устройства. Подумайте, что мы смогли создать таким способом. По-моему, замечательный результат!

Подключение звука и распространение проекта

Краткое содержание главы:

- подключение к проекту звука;
- создание индикаторов загрузки и просмотра;
- распространение законченного проекта в сети и на дисках.

Данная глава разбита на три основные части, описывающих три финальных шага, которые нам осталось сделать для завершения работы над роликом для парка. В первой части, прежде чем создавать возможность получения от пользователей отзывов о проекте и публиковать файлы в сети и на CD-ROM, мы попробуем дополнить наши клипы подходящим звуковым сопровождением.

Звук

Есть два способа подключения звука к проекту. Первый довольно прост и, наверняка, вам известен. Воспроизведение звука начинается и прекращается с указанных вами ключевых кадров фильма. Если вам это и нужно, то есть нужно просто проигрывать звук какое-то время, то данный способ вполне подойдет. Он незамысловат и прозрачен, но не дает дополнительных возможностей, если они вам понадобятся.

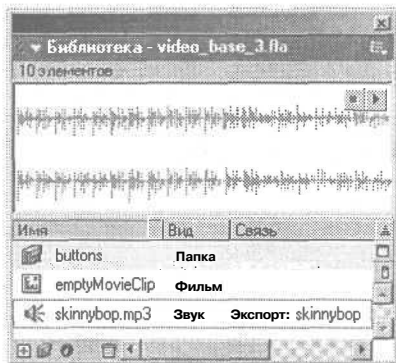
Можно пойти другим путем. Есть способ, позволяющий превратить воспроизведение звука в интерактивный процесс. При этом звук под-

ключается с помощью программного кода. Этот метод подробно рассмотрен в главе 7.

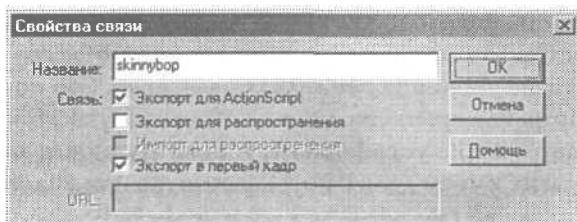
Подключение звука

Не забыли, что в предыдущей главе мы для удобства создали пустой клип с именем `video_mc`? Это позволило нам создать кнопки управления воспроизведением клипов, так как все они загружались под этим именем. Со звуком мы можем использовать тот же прием. С помощью ActionScript можно подключить к фильму звуковой объект, задавая ему определенное имя. При этом появляется возможность распределения звука по колонкам и усиления или приглушения одних звуков на фоне других.

1. Откройте свой файл FLA, с которым вы экспериментируете. Если для начала вам хочется подглядеть, что мы собираемся сделать, можете открыть `video_base3 fla` из приложенных к книге файлов.
2. Давайте немного приберемся в нашей Библиотеке. Создайте в ней новую папку под названием «buttons», и перенесите в нее все свои кнопки.
3. Импортируйте свой звуковой объект с помощью команды Импорт... в меню Файл. Я использовал созданный мною трек с именем `skinnybor.mp3`. Вы, конечно, можете использовать его или любой другой.

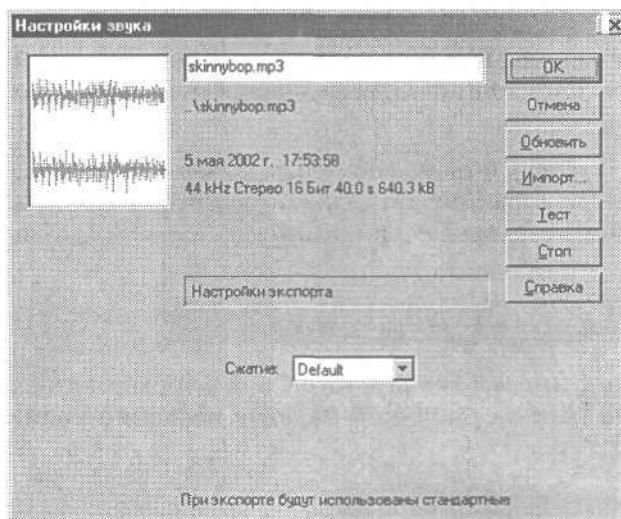


4. Выделите этот файл в Библиотеке и выберите пункт Связи... (Linkage...) из меню, появляющегося при щелчке на значке в верхнем правом углу окна Библиотека. На рисунке показаны свойства файла `skinnybor.mp3` — он будет экспортирован для ActionScript под тем же названием «skinnybor»:



Для ActionScript лучше оставить то же название, даже если вы решили использовать другой звуковой файл, так будет легче сориентироваться в нижеследующем программном коде.

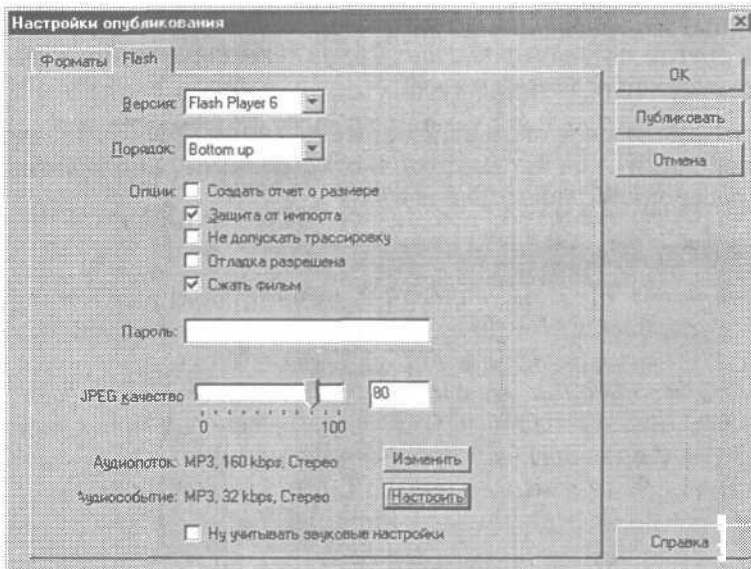
5. Закройте это диалоговое окно и щелкните на синем круглом значке «i» в нижней части окна Библиотека. В открывшемся окне указаны свойства (настройки) нашего файла MP3:



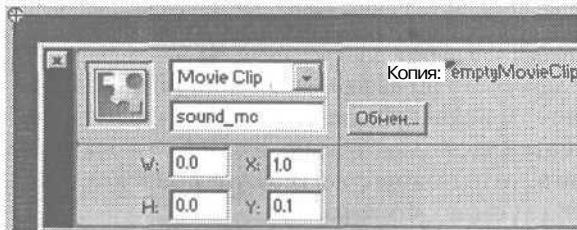
При импорте MP3-файла во Flash он автоматически применяет сжатие в формате MP3. При этом исходный файл, который мы импортируем, был высокого качества, а, следовательно, и слишком большого размера для размещения в сети. Наверняка, это будет справедливо и для любого другого используемого вами файла.

6. Для его сжатия в поле Сжатие (Compression) выберите из списка Default, при этом звук будет сжат в соответствии с параметрами, установленными в окне Настройки опубликования (Publish Settings).

- Откройте это окно с помощью команды **Общие параметры** в меню **Файл**, в поле **Аудиособытие (Audio Event)** нажмите кнопку **Настроить**, и установите для сжатия значения **MP3, 32 kbps, стерео**. Это позволит уменьшить размер файла при подходящем качестве звука. При этом не важно, какие параметры установлены в поле **Аудиопоток (Audio Stream)**, так как мы не имеем дело с потоками звука или видео в нашем файле:



- Создайте пустой клип с именем «emptyMovieClip» в верхнем левом углу вашего фильма (как на рисунке). Задайте название копии «sound_mc»:



- Выделите первый кадр и откройте панель **Действия (Actions)**. Введите следующие строки:

```
tune = new Sound (sound_mc);  
tune.attachSound ('skinnybop');  
tune.start (0, 690);
```

Разберемся, что мы написали. В первой строке создается объект под названием «tune» в пустом клипе с именем копии `video_mc`. Во второй строке мы прикрепили к нему наш звуковой файл. Третья строка запускает воспроизведение нашего звукового объекта, причем в скобках мы указали, что мелодия должна проигрываться с самого начала и повториться 690 раз.

Почему именно 690 раз? А просто так, на ум пришло это число, а вообще подойдет любое достаточное большое. Здесь главное, чтобы музыка повторялась все время, пока пользователь просматривает проект, а он может делать это довольно долго. В нашем случае трек длится 40 секунд, можно подсчитать, что она повторится 690 раз примерно за семь часов.

В данном примере мы задали название цели (sound_mc) для созданного звукового объекта. Вообще говоря, цель можно и не задавать, то есть вполне подошла бы и такая строка `tune = new Sound()`; Однако, как и в случае с видео, указание цели дает новые возможности, например, позволяет устанавливать различные уровни громкости, когда используется несколько звуковых объектов. Кроме того, это помогает избежать многих трудностей при многократном использовании команды `loadMovie`.

К созданному объекту (tune) я прикрепил звуковой файл, используя его имя (skinnybop). Это похоже на применение пустого клипа в случае с видео, разница заключается в том, что звуковой объект существует только виртуально, то есть в программном коде — физически он в фильме не присутствует. Тем не менее, у нас есть возможность с помощью ActionScript управлять подключаемым звуком, так как он имеет имя. Давайте это и попробуем сделать.

Приглушение звука

Напомню, что мы предоставили пользователю возможность замедленного просмотра фильма при нажатом положении кнопки SLOW. Мы хотели, чтобы громкость звука при этом немного уменьшалась, а при нормальном режиме — увеличивалась. Для воплощения этого замысла нам нужно всего лишь добавить несколько строк в написанный для кнопки SLOW программный код.

1. Выделите кнопку SLOW и перейдите на панель Действия. Вставьте в функцию on (press) строки, выделенные жирным шрифтом в нижеследующем коде:

```
on (press){
    video_mc.stop ();
    counter = 0;
    tuneVol = tune.getVolume ();
    video_mc.onEnterFrame = function () {
        counter ++
        if ((counter % 3) == 0){
            video_mc.nextFrame ();
        }
        if (tuneVol > 0){
            tuneVol --;
            tune.setVolume (tuneVol);
        }
    }
};
```

Итак, давайте разберемся, как будет происходить приглушение звука. Первая добавленная строка создает переменную с именем tuneVol и присваивает ей текущее значение громкости звука. Здесь не очень важно разбираться в подробностях, как это происходит. Грубо говоря, Flash просто определяет текущую громкость нашего звукового объекта tune. Громкость звукового объекта является одним из параметров клипа, который можно определить или установить с помощью Flash. В нашем случае это происходит при нажатии кнопки.

Далее в следующей добавленной строке if (tuneVol > 0) проверяется условие, является ли положительным значение tuneVol. Если это так, то выполняются следующие две строки.

В них значение tuneVol уменьшается на единицу, и это значение устанавливает новую громкость нашего звукового объекта. Таким образом, мы получили эффект уменьшения громкости объекта tune на единицу за кадр. Нормальное значение громкости равно 100, поэтому чтобы полностью заглушить звук, потребуется 100 кадров.

В итоге мы обогатили функцию замедленного показа возможностью приглушения звукового объекта. Поэтому когда пользователь отпускает кнопку SLOW, строка video_mc.onEnterFrame = null вместе с прекращением действия этой функции отменяет команды уменьшения громкости звука.

2. Осталось только к коду `on release` добавить строку, вновь устанавливающую громкость объекта `tune` в значение `100`. Именно это и происходит в строке `tune.setVolume(100);`. Она входит в состав функции, применяемой к клипу с именем `video_mc`, и поэтому справедлива для каждого его кадра. Добавьте следующие выделенные жирным шрифтом строки:

```
on (release) {
    video_mc.onEnterFrame = null;
    video_mc.play();
    tune.setVolume (100);
    if (tuneVol == 0){
        tune.stop ();
        tune.start (0,690);
    }
}
```

Это необходимые последние штрихи, с помощью которых мы учтем тот случай, когда пользователь удерживает кнопку **SLOW** нажатой довольно долго, так что громкость объекта `tune` достигает нуля, и звука вообще нет. Я поймал себя на том, что удерживал кнопку в моменты, когда катальщики взмывали высоко над бортами, и отпускал ее, только когда они приземлялись.

Громкость звука в начале максимальна, поэтому я подумал, что при полном отключении звука лучше запускать объект с начала с наибольшей громкостью. Именно это и происходит в добавленных строках — проверяется значение громкости объекта `tune`, и если оно равно `0`, то выполняются следующие две строки: воспроизведение мелодии прекращается и начинается с начала.

Можно пойти дальше и подключить два звуковых объекта. Тогда пока кнопка **SLOW** нажата, вы можете приглушить одну мелодию и усилить другую, в то время как при нормальном показе громкость первой восстанавливается. Такой прием создает эффект микширования звуков. При этом нужно создать два звуковых объекта и, соответственно, два пустых клипа, чтобы управлять каждым из них независимо от другого.

Уже загрузили?

Работа над роликом подходит к концу, однако осталась важная деталь. До этого мы всегда загружали его со своего жесткого диска, но, как вы помните, по нашему первоначальному плану мы должны получить из него работоспособный сайт и CD-ROM. Кроме того, из-за использова-

ния видеофайлов, которые, как правило, велики по размеру, мы должны учесть возможности посетителей, пользующихся модемом. Поэтому теперь нам нужно протестировать свой проект в сети.

В этой книге мы уже использовали программу Bandwidth Profiler для имитации загрузки фильма в глобальную сеть при различных скоростях соединения. Однако в данном случае этого продвинутого приложения недостаточно. Дело в том, что оно симулирует только скорость загрузки экспортируемого в текущий момент фильма, а остальные клипы не загружаются. Между тем, основное содержание несут именно фильмы, поэтому единственный способ проверить, как проект поведет себя в сети — это просто поместить его там.

Как только я сделал это, я заметил, что чего-то недостает: нет возможности узнать, какую часть открываемого фильма осталось загрузить. В зависимости от применяемого мною способа соединения с всемирной паутиной я могу видеть, что определенная часть видео довольно быстро проигрывается, затем останавливается, и вновь воспроизводится весь ролик с начала до конца. Это происходит из-за того, что Flash организует потоки видео, то есть перенос данных производится частями. Такой прием позволяет нам просмотреть уже загруженную часть в то время, пока оставшиеся кадры передаются в фоновом режиме.

Создание индикатора выполнения

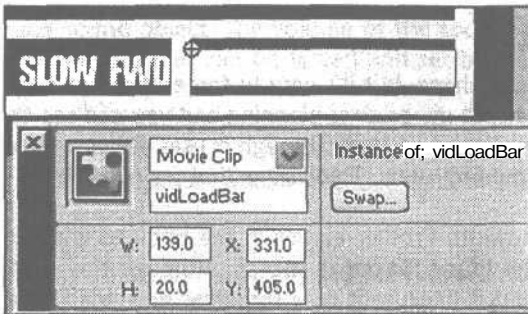
Итак, мы выяснили, что нам нужно создать специальный индикатор, указывающий, какая часть видео уже загружена.

1. Откройте файл `video_base_4 fla`, являющийся прототипом того, что мы хотим получить. Вы заметите белый прямоугольник в правом нижнем углу кадра.

Это и в самом деле просто белый прямоугольник, но он не нарисованный на кадре, а внедренный в клип, то есть он имеет название копии, `vidLoadBar`. Вы уже знаете, что дает нам имя копии — возможность контроля с помощью ActionScript, а также изменения ширины прямоугольника в зависимости от величины загруженной части видео.

2. Откройте свой вариант файла, с которым вы экспериментируете, и нарисуйте на слое `layer` линию, отделяющую индикатор загрузки от кнопок.
3. Создайте новый клип, назовите его «`vidLoadBar`» и нарисуйте в нем белый пустой прямоугольник.
4. Добавьте новый слой с именем `loader` между слоями `buttons` и `boxes`.

5. Перетащите копию vidLoadBar в область кадра на слое loader. С помощью панели Параметры (Property) разместите его, как показано на рисунке. Задайте размеры 139x20 и задайте имя копии vidLoadBar:



6. Откройте панель Действия на первом кадре слоя code, и добавьте к коду следующие строчки:

```
vidLoadBar.onEnterFrame = function (){
    total = video_mc.totalframes;
    soFar = video_mc_framesloaded;
    loaded = soFar / total * 100;
    vidLoadBar._xscale = (loaded);
};
```

Как видите, первая строка нам уже знакома, похожее мы писали для кнопок FWD, RWD и SLOW: к объекту подключается функция, которая будет применяться к каждому кадру. В данном случае таким объектом, вызывающим функцию, является vidLoadBar.

В следующей строке определяется общее число кадров в загружаемом фильме, в то время как в третьей строке определяется число уже загруженных кадров. В четвертой строке мы путем простых арифметических действий находим долю загруженной части в процентном отношении.

Последняя строка изменяет ширину белого прямоугольника в соответствии с числом, вычисленным в предыдущей строке. То есть когда оно станет равным 100%, ширина прямоугольника достигнет наибольшей величины.

7. И, наконец, в нижней части индикатора загрузки мы поместим надпись «LOADING», появляющуюся только, когда ширина прямоуголь-

ника меньше 100%. Для этого добавьте новый слой с именем `loading text` под слоем `loader`, и вставьте надпись в указанное место.

При тестировании проекта на жестком диске прямоугольник с самого начала примет наибольшую ширину, так как Flash в этом случае произведет загрузку очень быстро. Как было выше отмечено, и `Bandwidth Profiler` тоже нам не подойдет, так что придется проверять все в реальных боевых условиях, то есть в сети, но об этом поговорим позже. А пока вы можете, не изменяя свои файлы, просмотреть `video_base_4 fla`, в который внесены изменения, необходимые для загрузки видеофайлов непосредственно с URL.

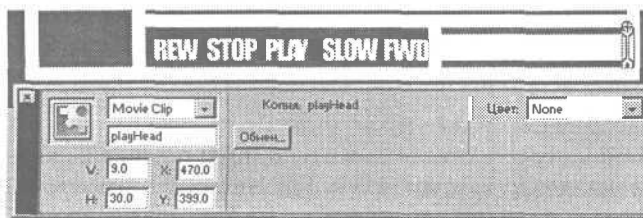
В каком мы месте фильма?

Итак, теперь у нас есть индикатор загрузки видео, но при просмотре неизвестно, в каком месте фильма мы находимся. Вернее, нам известно, ведь пришлось не раз изучить видео вдоль и поперек за все это время. А у будущих пользователей такой возможности нет. Поэтому сейчас мы попробуем создать индикатор просмотра ролика с бегунком.

Создание индикатора просмотра

Для начала можно открыть файл `video_base fla` и посмотреть, о чем идет речь. Если вы щелкните на маленьком значке бегунка в верхней части белого прямоугольника, вы увидите то, что показано на рисунке. Как вы уже догадались, это новый клип под названием `playHead`.

1. Закройте `video_base fla` и откройте свой рабочий файл.
2. Создайте новый клип с именем «`playHead`» и нарисуйте значок, который, по-вашему, может служить бегунком.
3. Добавьте новый слой `playHead` между слоями `buttons` и `loader`. Перетащите на него копию нового клипа `playHead`, и задайте ей имя копии «`playHead`». При этом важно правильно ввести значение координаты `_y`, в то время как значение `_x` будет определено в коде ActionScript, который мы сейчас запишем.



4. Программный код для значка бегунка будет введен в слое `code` на главной временной шкале. Добавьте к имеющемуся коду индикатора загрузки следующие строки:

```
playHead.onEnterFrame = function () {  
    fTotal = video_mc.totalframes;  
    fNow = video_mc.currentframe;  
    fPos = fNow / fTotal * vidLoadBar._width;  
    playHead.x = fPos + vidLoadBar._x;  
};
```

Как вы заметили, здесь много общего с кодом для индикатора загрузки. К фильму прикрепляется функция с тем же названием «`playHead`». Во второй строке определяется общее число кадров в видео, а в третьей — номер текущего просматриваемого кадра.

Четвертая строка напоминает соответствующую строку для индикатора загрузки при нахождении доли загруженной части файла. В ней номер текущего кадра на общее число кадров и умножается на величину `vidLoadBar._width`, означающую ширину белого прямоугольника индикатора. Таким образом, в самом начале воспроизведения полученное число будет очень маленьким, а в конце — оно будет равно 139 (такова установленная нами ширина индикатора).

В последней строке к вычисленному таким образом значению прибавляется величина `_x` индикатора (положение его левого края, откуда бегунок будет начинать свое движение). То есть, в процессе просмотра бегунок будет двигаться слева направо вдоль всей длины индикатора.

Теперь можете запустить фильм и проверить бегунок в действии.

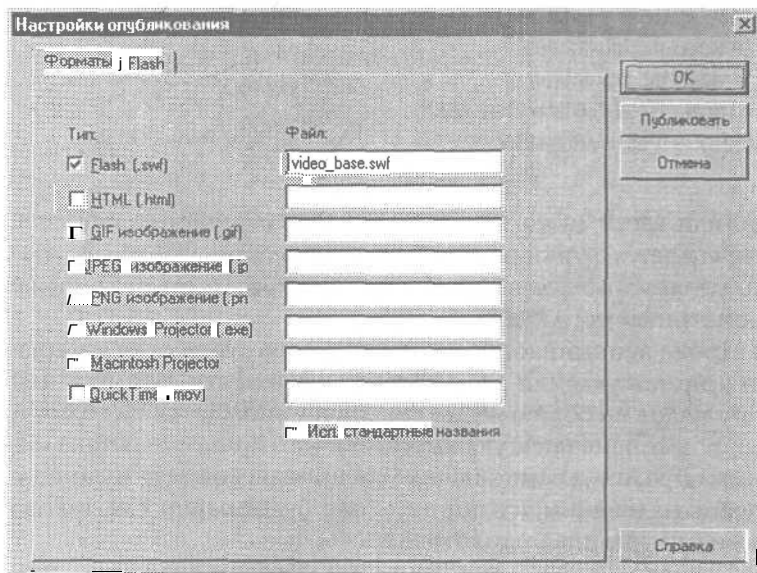
Ну а теперь можно распространять проект?

Ну что ж, теперь, когда у нас есть все нужные фильмы с необходимой оснасткой, работа почти завершена. По плану рекламный ролик для парка должен быть размещен в сети и на CD-ROM. Так как продукт готов к употреблению, нам с вами осталось только договориться о загружаемой в сеть версии проекта.

Подготовка

Для того чтобы наш проект был удобен для использования в сети, мы должны сделать три вещи. Нужно опубликовать наши файлы, убедить-ся, что они загружаются у пользователя, и разместить их на сервере.

1. Прежде всего, откройте свой файл FLA (в моем случае video_base.fla), а затем щелкните на команде Общие параметры... (Publish Settings) в меню Файл.



2. Снимите флажок *Использовать стандартные названия* (Use default names), чтобы вы могли задать имена публикуемых файлов. Мы использовали одно название с добавлением возрастающих цифр по мере увеличения функциональных возможностей. Это должно стать хорошей привычкой, так как позволяет легко найти и вернуться к предыдущей версии, если вы случайно испортили последний вариант. Тем не менее, в законченном Web-сайте будет происходить ссылка на фильм Flash под названием video_base.swf. Поэтому нужно внести изменения.
3. Затем щелкните на команде *Опубликовать*.

Внимательный читатель, наверное, заметил, что я кое-что пропустил. Размер фильма video_base около 150 КБ, поэтому нам нужен файл предварительной загрузки. Проблема в том, что такой размер почти полностью определяется звуковым файлом skinnybor.mp3. Как вы помните, в его Свойствах связи (Linkage Properties) мы установили флажок экспортирования для ActionScript в первый кадр фильма, то есть, прежде всего, будет производиться загрузка звука целиком, а потом всего остального, включая и предварительный загрузчик.

Выход в создании фильма предварительной загрузки, который будет загружать `video_base.swf` точно так же, как и сам `video_base fla` в свою очередь загружает остальные фильмы SWF. Если все это звучит слишком запутанно, можете просто просмотреть файл `preloader fla`.

4. Создайте новый фильм под названием `preloader fla`, и задайте в нем те же размеры кадра, что и в `video_base fla` (500x460).
5. Нарисуйте большой белый прямоугольник в центре и преобразуйте его в фильм. Задайте имя копии «loadBar».



6. Поместите копию элемента `emptyMovieClip` в верхний левый угол сцены и задайте имя копии «video_base_mc».
7. Добавьте новый слой под названием «code» и введите следующий программный код:

```
loadMovie ('video_base.swf', video_base_mc);
```

Он уже знаком нам, так как очень похож на код для кнопок выбора клипов (`VIDEO01`, `VIDEO02`). Мы дали Flash команду загрузить фильм `video_base.swf` в области `emptyMovieClip`, названной `video_base_mc`.

8. Затем добавьте следующие строки:

```
loadBar.onEnterFrame = function () {
    total = video_base_mc.getBytesTotal ();
    soFar = video_base_mc.getBytesLoaded ();
    percent = soFar / total * 100;
    loadBar._xscale = (percent);
};
```

Если вы сравните их с кодом, введенным для индикатора загрузки, то заметите большое сходство, за исключением того, что определяется доля загруженной части фильма `video_base`, и обращение происходит к клипу с именем копии `loadBar` для отображения изменения ширины индикатора. Еще одна разница в том, что доля загруженной части фильма определяется не по числу кадров, а по размеру загруженной части файла.

9. И, наконец, последняя порция кода:

```
if (percent == 100 && total > 10){
    loadBar.onEnterFrame = null;
}
```

```

loadBar._visible = 0;
};
};

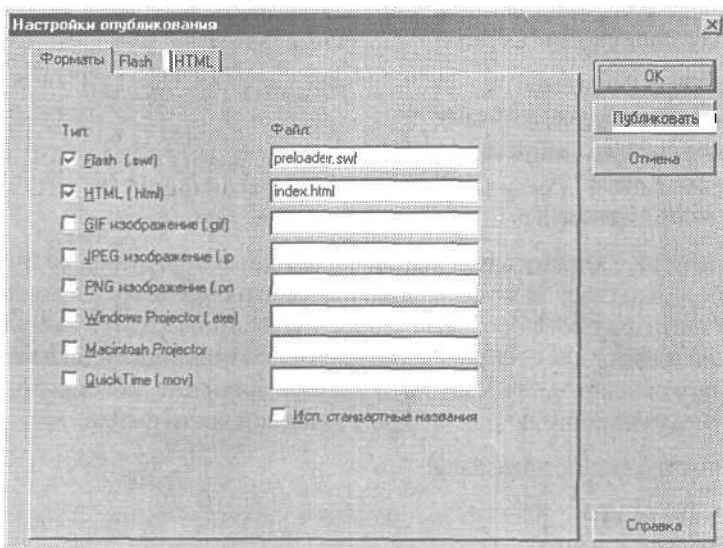
```

Первая строка указывает Flash выполнить следующие строки, если фильм `video_base` полностью загружен. Это просто служебные операции: прекращается действие функции, подключенной к объекту `loadBar`, а затем этот прямоугольник становится невидимым.

Кроме того, в первой строке проверяется, превосходит ли полный размер загружаемого файла 10 байт. Дело в том, что пустой клип записывается как файл размером 4 байта, поэтому условие `total>0` здесь не подходит.

Мы загружаем фильм `video_base.swf` в пустой клип, то есть, используем уже известный нам способ во избежание ненужной путаницы. Вообще говоря, я бы загрузил `video_base.swf` в уровень, а не в цель, так загружаемый фильм имеет те же размеры и те же координаты кадра, что и файл предварительной загрузки. При таком подходе уже не нужен пустой ключевой кадр, а простая строка `loadMovieNum ('video-base.swf', 69);`. Для дальнейшего упрощения можно вместо метода `createEmptyMovieClip` использовать физические копии пустых клипов из Библиотеки.

10. Щелкните на пункте Общие параметры в меню Файл, затем в название фильма Flash введите файл предварительной загрузки и установите флажок для страницы HTML. Назовите HTML-файл «`index.html`»:



11. Последний шаг — перенести все файлы на сервер. На рисунке перечислены все созданные нами документы: четыре документа Flash, четыре фильма Flash и один HTML-файл. Нам следует перенести на сервер только SWF-файлы и HTML-страничку.

Имя	Размер	Тип
preloader	22 КБ	Flash Document
video_base_5	2 187 КБ	Flash Document
video1	1 830 КБ	Flash Document
video2	4 384 КБ	Flash Document
index	1 КБ	HTML Document
preloader	1 КБ	Фильм Flash
video_base	1 КБ	Фильм Flash
video1	322 КБ	Фильм Flash
video2	1 240 КБ	Фильм Flash

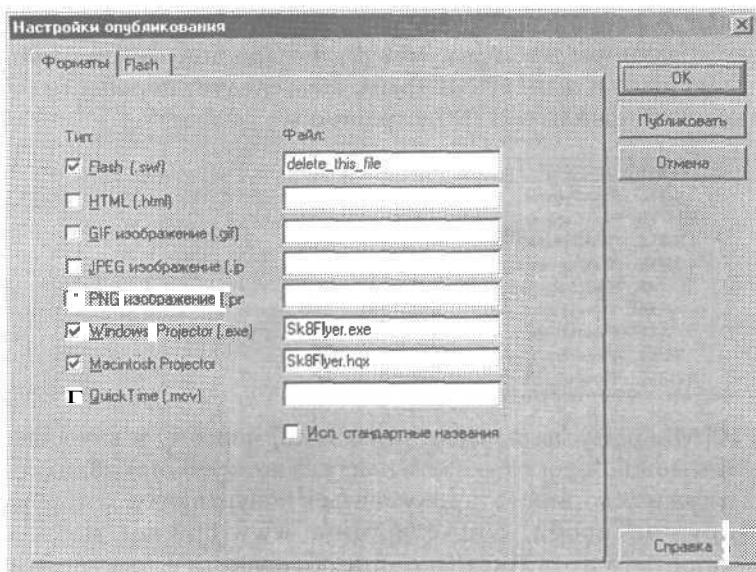
12. Исходный HTML-файл мы назвали index.html, поэтому в качестве адреса файлов можно просто указать папку, в которой они находятся. Для этого нужно скопировать документы в новую папку, которую я, к примеру, на своей Web-странице www.h69.net назвал sk8flyer_tutorial. Это позволяет мне при обращении к указанным файлам просто ввести адрес URL www.h69.net/sk8flyer_tutorial.

Очень важно, чтобы все файлы находились в одной папке, иначе видео загружаться не будет. Если вам нужно использовать большое количество элементов видео, то правильнее будет хранить все фильмы SWF в отдельной вложенной папке с названием «video». При этом программный код для кнопок придется изменить, добавив перед именами файлов путь «video/». Например, строка loadMovie («video1.swf", "video_mc"); примет вид 'loadMovie("video/video1.swf", "video_mc");'.

Теоретически, все, что мы должны предпринять для создания CD-ROM-версии, это составить самостоятельную оболочку. Эту несложную задачу мы и рассмотрим. Постараемся создать CD-ROM двойного формата (Маси PC), для чего нам нужен записывающий CD-ROM с подходящим программным обеспечением.

Однако перед этим несколько полезных советов.

13. Сохраните свой файл FLA как video_base_cd fla, в который мы внесем изменения, и установите показанные на рисунке Общие параметры (Файл > Общие параметры...):



14. На компьютере Macintosh вся процедура выглядит точно так же. Мы собираемся создать Windows и Mac-приложения. Как видно на рисунке, также нужно создать файл SWF с названием, напоминающим, что его нужно позже удалить. Зачем же тогда его создавать? Это такая причуда Flash, вернее его окна установки Общих параметров. Если не создавать файл SWF, то не появится вкладка Flash, и поэтому у нас не будет возможности настройки качества звука и других свойств приложения.

15. Нажмите ОК, перейдите на основную временную шкалу и выберите кадр номер1 на слое code. В самом начале подключенного кода нужно добавить две строки для самостоятельной версии рекламного ролика:

```
fsccommand ('fullscreen', 'true');
fsccommand ('allowscale', 'false');
```

В первой строке приложению дается команда развернуть окно с фильмом во весь экран. Следующая строка указывает Flash не увеличивать масштаб изображения при полноэкранном режиме. Напротив, вся область вокруг кадра будет заполнена выбранным для фильма фоновым цветом.

Если мы позволим Flash изменять масштаб изображения в соответствии с размерами окна, то пользователь со скромным компьютером,

выбрав разрешение 1024x768 и тем самым сильно загрузив свой процессор, увидит не классно сделанный ролик, а череду прерывающихся кадров. А впрочем, можете изменить вторую строку и просмотреть масштабированный фильм на своей машине.

В полноэкранном режиме можно в любой момент нажать клавишу Esc для возврата в оконный режим.

16. Теперь нужно подключить к ролику еще одну важную возможность, которая была не нужна в Web-версии, поскольку в ней нет полноэкранного режима. Это кнопка выхода из приложения. Многие люди (в том числе и автор этих строк) часто забывают об этом. Создайте новую кнопку (Ctrl + F8) под названием «quitBut», и нарисуйте на ней что-нибудь подходящее — значок «х» или «Quit», короче, что вам по душе.
17. Перенесите копию кнопки из Библиотеки на слой layer.
18. Щелкните на кнопке QUIT и на панели Действия введите следующий код ActionScript:

```
on (release){  
    fscommand ("quit");  
}
```

Как видите, тут все просто. При нажатии пользователем кнопки QUIT приложение закрывается.

19. Наконец, осталось создать приложения. Для этого нужно щелкнуть на команде Опубликовать (Publish) в меню Файл. Так как Mac Projector создается в сжатом формате, его нужно перенести на компьютер Mac и восстановить.

Пока исходный файл video_base.swf и все созданные фильмы SWF находятся в одной папке, у приложения не будет проблем с их загрузкой. Если вы располагаете временем, можете обновить ролики фильмами получше и экспортировать все с более высоким качеством звука. Размеры файлов при этом заметно увеличатся, но для CD-ROM-версии это не очень важно.

Если же вам по нраву потратить свое время как-нибудь иначе, нежели возиться с проектом, то можете использовать те же фильмы SWF, что и в Web-версии. Для придания приемлемого внешнего вида можно все файлы скрыть, за исключением приложения, так что конечный пользователь будет иметь дело только с одним файлом. Скрытые фильмы будут так же без проблем загружаться.

Заключение

Хотя в этой главе было рассмотрено много вопросов, все они, тем не менее, являются основополагающими. Все вышерассмотренное может вам пригодиться при создании будущих проектов. Нужно просто заметить файлы своими, а основу сохранить, и вы получите новый ролик.

В последних трех главах говорилось о том, как создать прототип рекламного ролика для скэйт-парка. Для получения готового к использованию ролика нужны еще некоторые действия по отладке и улучшению интерфейса. Для описания всех этих операций потребовалось бы написать отдельную книгу, и все же все необходимые сведения по данным второстепенным действиям и указанные в книге файлы можно скачать на странице www.friendsofed.com.

Надеюсь, обильные комментарии вам не мешают, а только помогают легко разобраться в том, что мы делали. Вы поймете, что сложность здесь только видимая — нужно из отдельных блоков собрать нечто большее, чем простая сумма частей — создать хороший проект.

До этого автор старался придерживаться шутливой тона изъяснения материала. Однако читателю не следует думать, что рассмотренные вещи так же несерьезны. Тем более это справедливо по отношению к следующей главе, которая потребует от вас большего внимания. Итак, удачи!

Раздел 3

Передовые идеи

Программирование с использованием видео и звука

Краткое содержание главы:

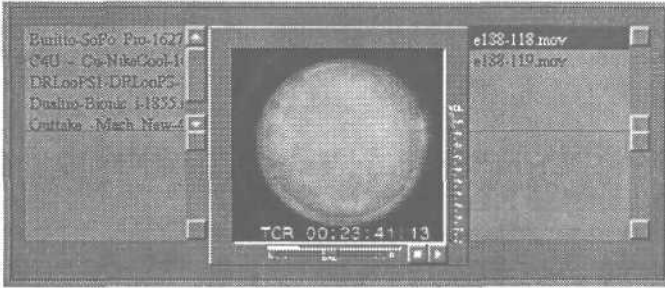
- введение в объектно-ориентированное программирование (ООП);
- создание пользовательских компонентов;
- создание универсального проигрывателя.

Если вы все-таки дошли до этой части книги, то, наверняка, уже приобрели все навыки, необходимые для эффективной работы в мире Flash MX video. В данном последнем разделе мы попробуем подняться на новый уровень, вооружившись всем, чему научились.

По правде говоря, рассматриваемые в разделе вещи нельзя назвать простыми. Поэтому без опыта работы с ActionScript из предыдущих глав к концу этого раздела вы мало что приобретете, кроме усталости и головной боли. Более того, даже имея подобный опыт, вам придется быть внимательней, так как предстоит встреча с новыми концепциями.

Без них книгу о видео во Flash MX нельзя считать законченной. Это все равно, что пролежать все дни каникул на солнечном пляже какого-нибудь райского уголка планеты, так и не узнав всей его красоты и экзотики. Осилив данный раздел с помощью наших советов, вы в награду получите полностью настраиваемый проигрыватель звука и видео в качестве компонента Flash.

Что это значит? Это значит, вы станете обладателем проигрывателя, который можно использовать во всех будущих проектах простым переносом с небольшой подстройкой. На рисунке показано то, о чем идет речь.



Перед тем как приступить к реализации проекта, рассмотрим вкратце теорию вопроса.

Объекты и объектно-ориентированное программирование

Большинство людей при слове «объект» или «объектно-ориентированный» инстинктивно вспоминают о «программировании». Это происходит во многом благодаря тому, что данные термины в сочетании дают название новой концепции. Принципы объектно-ориентированного программирования (ООП) были заложены в 1960-х годах, когда возникла потребность программирования сложных задач более простыми и понятными способами.

Такая потребность с годами все возрастала, что привело не только к разработке множества языков объектно-ориентированного программирования, но и к созданию так называемых заготовок **drag-and-drop**, то есть готовых для использования объектов. Более того, применение таких объектов настолько упростило работу, что во многих случаях вовсе отпадает необходимость в программном коде. Одним из больших сторонников использования заготовок drag-and-drop является Macromedia Flash. Не нужно определять для Flash, что такое фильм, вместо этого вы просто создаете его и используете. Возможно, вы об этом и не подозревали, но при каждом использовании фильма вы на самом деле практиковались в ООП.

Главная предпосылка здесь в том, что объекты сохраняют связанные с ним данные и программный код. Они привлекают тем, что пользовате-

лю не нужно знать внутреннее устройство объекта; достаточно знать, какими данными его «заполнить» и к чему это приведет. Такой подход часто описывается как **черный ящик** — мы не знаем, как он устроен; нам и не нужно это знать, пока он делает то, что мы от него хотим.

Тем не менее, каждый полезный объект был кем-то запрограммирован. В этой главе мы будем не только использовать их, но и попробуем создать свои многократно используемые объекты — компоненты проигрывателей.

Простейший объект

Наиболее общим объектом во Flash является Object. Какой бы объект не был создан, он будет наследовать методы (поведение) и свойства (характеристики) исходного объекта. По сути, любой новый объект перенимает «гены» родительского объекта. Например, познакомьтесь с Фредом:

```
fred = new Object();
```

Здесь Фред — это новый объект, унаследовавший свойства объекта Object. Пока он еще довольно примитивен. Однако он может превратиться в новый фильм:

```
fred = new MovieClip();
```

Теперь у Фреда появились какая-то индивидуальность. Он приобрел все свойства и методы объекта MovieClip, который в свою очередь унаследовал все свойства и параметры объекта Object. Так что Фред стал намного круче. С помощью ActionScript мы можем получить доступ к множеству таких методов, как play, stop и enterFrame, а также к многочисленным свойствам, включая _x, _y (определяющим положение Фреда в кадре) и _width, _height.

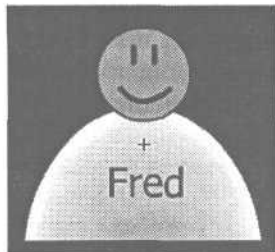
До появления версии Macromedia Flash MX новый клип создавался с помощью команды Новый символ в меню Вставить с последующим добавлением графических элементов в его временную шкалу. Такой новый символ тоже является *объектом* в смысле привязанности его программного кода и переменных к соответствующей временной шкале, но он будет более «объектно-ориентированным», когда не требуется статичное изображение.

Динамический Фред

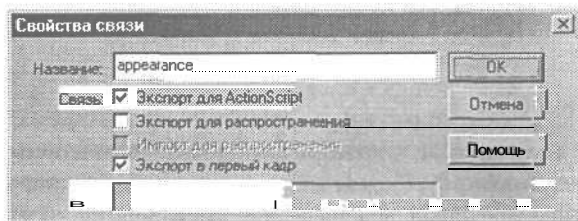
В предыдущих версиях Flash внешний вид фильма из Библиотеки можно было изменить с помощью метода attachMovie. Во Flash MX наряду с изменением изображения в фильме также есть возможность динамич-

чески составлять фильмы. Давайте попробуем создать динамического Фреда и придадим ему какую-нибудь наружность.

1. Создайте новый графический символ с помощью команды Новый символ (New Symbol) в меню Вставить (Insert). Нарисуйте на сцене Фреда на свой вкус:



2. Зайдите в Библиотеку и выделите созданного вами человечка. Из меню Библиотеки (оно появляется при щелчке на значке в верхнем правом углу панели) выберите команду Связи. Установите флажок Экспорт для ActionScript и в поле Название введите «appearance»:



3. Затем для первого кадра основной временной шкалы введите следующий программный код:

```
fred = createEmptyMovieClip ("fred", 1);  
fred.attachMovie ("appearance", "newAppearance", 1);
```

Теперь наш Фред не только представляет собой объект со свойствами и методами объекта `MovieClip`, но и приобрел внешний вид с названием `appearance`, который создан не с помощью диалогового окна Создать новый символ (Create New Symbol) а взят из графического символа.

4. Сохраните фильм под названием `fred1.fla`.

Методы и свойства

Что же такое методы и свойства, и для чего они нужны? Метод, по сути, является функцией, применяемой к объекту, то есть определенным действием над ним. Свойства — это обычные свойства. Они описывают параметры объекта, положение, размеры и т.д.

Самым простым методом, известным всем пользователям Flash, является `play`. Так как в принципе любой объект может быть преобразован в объект `MovieClip` посредством `ActionScript` или с помощью команды Новый символ в меню Вставить, то к нему можно применить метод `play` и другие подключенные методы.

Кроме того, можно создать и применить к копии объекта новый метод, например:

```
fred.rotate = function (n) {  
    This._rotation += n;  
};
```

Так, при каждом обращении к `fred.rotate(number)` Фред будет поворачиваться на заданное число градусов. Эта функция `rotate (number)` является *методом*, применяемым к Фреду.

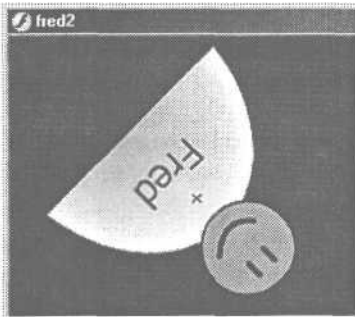
Вращение Фреда

Давайте испытаем новый метод.

1. Откройте файл `fred1 fla`, если он еще не открыт после последнего упражнения. Добавьте следующие функции непосредственно за введенными ранее строками:

```
this.onEnterFrame = function () {  
    fred.rotate (5);  
};  
fred.rotate = function (n) {  
    this.rotation += n;  
};
```

2. Сохраните файл как `fred2 fla`.
3. Теперь можно протестировать получившийся фильм (`Ctrl + Enter`), при этом вы должны увидеть Фреда, вращающимся по часовой стрелке:



Если этого не происходит, проверьте в окне Параметры связи (Linkage Properties), установили ли вы флажок Экспорт (Export) в первый кадр.

Понятно, что Фреда заставляет вращаться его свойство `_rotation`. Свойство является определенной характеристикой, принимающей различные значения для разных экземпляров объекта. Так, все объекты `MovieClip` имеют свойство `_rotation`, но его значение уникально для каждого экземпляра. Поэтому если, например, `fred._rotation` равно 5, то это вовсе не значит, что и `fred2._rotation` также должно равняться 5.

Классы и наследование

Как мы ранее выяснили, наследование означает простую передачу «генов» от одного объекта другому. Мы уже знаем, что такими генами являются методы и свойства, которые наследуются от объекта `Object`, или выражаясь точнее, **класса** `Object`. В случае с Фредом мы можем превратить его в класс, из которого позже можно создать двойников Фреда.

Знакомьтесь, Фред Младший

1. Откройте новый фильм и введите для первого кадра основной временной шкалы следующий программный код:

```
Fred = function () {
    This.x = 100
};
```

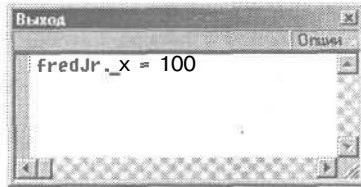
Как правило, названия классов начинают с заглавной буквы, а названия элементов данного класса — со строчной буквы. Это помогает легко их различать в ActionScript.

В нашем случае `Fred` это простейший класс, но при создании нового Фреда (`new Fred()`) мы увидим, что он наследует свойства своего «родителя».

- Следом добавьте следующие строки:

```
fredJr = new Fred ();
trace("fredJr._x = " + fredJr._x);
```

- Теперь проверьте свой фильм и просмотрите окно Выход (Output):



Как видите, координата `_x` объекта `fredJr` тоже равна `100`, так как она унаследовала данное свойство у класса `Fred`.

- Сохраните фильм как `fred3 fla`.

Прототипы

Одним из последних приемов, которые мы рассмотрим при работе с объектами, будет установка цепочки наследования с использованием свойств `prototype` и `_proto_`. Они позволяют определить самый верхний уровень любого класса. В нашем случае объект `Object` использует прототипы для определения класса. Эти прототипы задают объект `MovieClip` и, в конечном счете, `Фреда` тоже.

Можно для класса создать прототип, который будет применяться к любому представителю данного класса. На практике мы можем убедиться в этом установкой прототипа для класса `MovieClip` и проверкой, как это повлияет на `Фреда`.

Переделки

- Для начала откройте новый фильм. Вновь создайте `Фреда` путем подключения следующей строки `ActionScript` к первому кадру имеющегося слоя:

```
fred = new MovieClip();
```

- Под этой строкой добавьте к классу `MovieClip` следующий прототип:

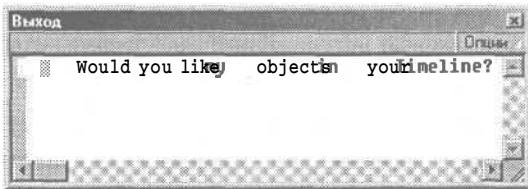
```
MovieClip.prototype.pickupLine = function () {
    Trace("Would you like my objects in your Timeline?");
};
```

Теперь новый метод `pickupLine` наследуется всеми объектами типа `MovieClip`, в том числе и нашим `Фредом`.

3. Для проверки вышесказанного нужно приписать еще одну строку:

```
fred.pickupLine();
```

4. Просмотрите фильм (Ctrl + Enter) и содержимое окна Выход:



Фред думает, что он в этом смысле стал уникальным. Это не так, но говорить ему не будем. Дело в том, что новый метод `pickupLine` доступен всем объектам класса `MovieClip`. Поэтому если мы запишем:

```
bob = new MovieClip();
```

то после записи:

```
bob.pickupLine();
```

будет несколько таких же Фредов.

5. Сохраните свой фильм как `fred4 fla`.

Область действия

Последний прием, который мы должны освоить для эффективной работы с объектами, связан с понятием области действия. Оно, по сути, означает точку (во временной шкале), в которой объявлен данный объект. К нему всегда можно обратиться, указав на его собственное «местонахождение» (`this`). Раньше, когда мы заставляли Фреда вращаться, вы, наверное, заметили, что мы обращались к нему как `this` при объявлении нового метода `rotate`. Мы вводили такую строку:

```
this._rotation+=n;
```

До тех пор, пока мы находимся внутри функции `fred.rotate`, мы знаем, что все действия совершаются над `fred`. Поэтому мы можем обращаться к нему как к `this`. Этот прием широко используется при задании структуры временной шкалы Flash. А так как, чаще всего, временная шкала устанавливается вместе с фильмами, то к ним можно также обращаться посредством `this`.

Основная временная шкала, кроме возможности использования ссылки `this`, имеет еще и собственный уникальный идентификатор `_root`. На каком бы уровне в иерархии вложенных объектов вы не находились, вы всегда

можете обратиться к основной временной шкале посредством `_root`. Такой подход очень похож на указание «абсолютного» пути к файлам на сервере: не важно, в какой из вложенных папок расположен файл, если ссылка начинается с корневого каталога (<http://www.LyuboeImyaDomena.com/>).

Во Flash MX используется новый идентификатор `_global`, который можно использовать для задания переменной или функции так, чтобы они были доступны с любой временной шкалы.

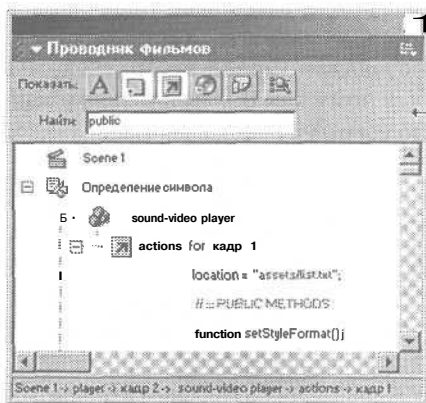
И, наконец, последний уникальный идентификатор, задающий область действия, это `_parent`. Где бы в целевом пути вы не использовали ссылку `_parent`, это всегда будет означать обращение к объекту, содержащему данный объект. Так, если, например, мы поместим новый фильм в существующий фильм, то новый фильм будет обращаться к существующему фильму посредством `_parent`.

Определение области действия объектов

Проводник фильмов и кнопка Вставить целевой путь (Insert a target path) (на панели Действия) очень полезны в тех случаях, когда нужно задать область действия или целевой путь объектов.

Проводник фильмов (Movie Explorer) (команда Проводник фильмов в меню Окно) не только помогает организовать размещение всех составных частей в фильме Flash, но также отображает весь программный код ActionScript, подключенный к фильму, и расположение объектов, включенных в другие объекты.

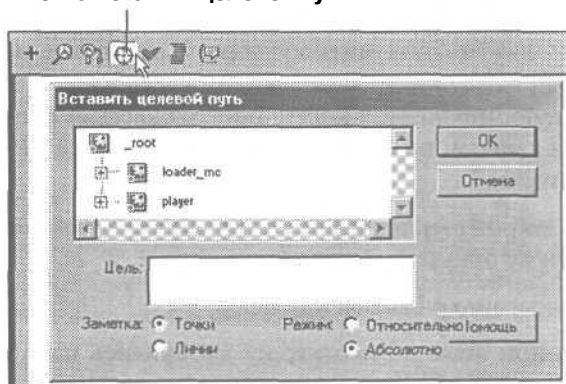
Кроме того, есть возможность поиска какого-либо слова среди объектов с помощью поля Найти.



Другой инструмент, кнопка Вставить целевой путь, позволяет найти любой путь, имеющий отношение к данному фильму, или указать путь к

объекту, начиная с корневой временной шкалы. Для эффективного контроля над клипами, загруженными фильмами или кнопками, необходимо уметь задавать целевой путь в ActionScript. Зная его, всегда легко найти объекты и задать относительные связи между ними:

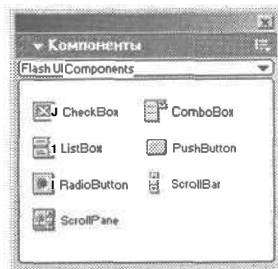
Кнопка Вставить целевой путь



Компоненты

Во Flash MX включен новый объект под названием компонент. Компонентами называют сложные клипы, имеющие установленные свойства и уникальный набор методов ActionScript, позволяющих устанавливать какие-то параметры и дополнительные опции во время прогона файла.

Во Flash MX есть несколько готовых компонентов на панели Компоненты (команда Компоненты в меню Окно):



В последнее время развивается тенденция создания полностью объектно-ориентированных компонентов, что справедливо и по отношению ко Flash MX. Их внешний вид можно настроить так, чтобы они вписывались в общую концепцию дизайна сайта. Значения переменных для компонентов устанавливаются на панели Параметры или в соответствующих интерфейсных меню.

Компонент **ListBox**

Теперь, когда мы уже знаем, что такое компоненты, можно рассмотреть один из них подробнее. Компонент **ListBox** предоставляет готовые окна списка с полосой прокрутки, которые можно вставить в фильм Flash.

Если вы уже уверенно чувствуете себя в объектно-ориентированном программировании и довольно далеко продвинулись в **ActionScript**, то этот компонент поможет вам заметно облегчить создание классов и организацию программных кодов.

Давайте рассмотрим следующие возможности: настраиваемые внешние оболочки, разделение классов на общие и частные, а также панель Определение компонентов.

1. Создайте новый фильм. Откройте панели Библиотека (F11) и Компоненты (Ctrl + F7), если они не открыты по умолчанию.
2. Затем перетащите один экземпляр **ListBox** на сцену.
3. Обратите внимание, что при этом в Библиотеку добавилась папка **Flash UI Components**. Она содержит все элементы для компонентов **ListBox** и **ScrollBar**:



4. Теперь откройте папку **Component Skins**, и в ней перейдите к папке **FScrollBar Skins**.
5. Дважды щелкните на значке `fsb_downArrow` для перехода в режим Редактирования символов. Если теперь вы щелкните на первом кадре слоя **README** и откроете панель Действия (F9), то будет отображен следующий код:

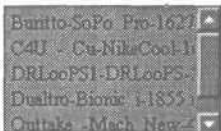
```
component.registerSkinElement(arrow_mc, "arrow");  
component.registerSkinElement(face_mc, "face");
```

```

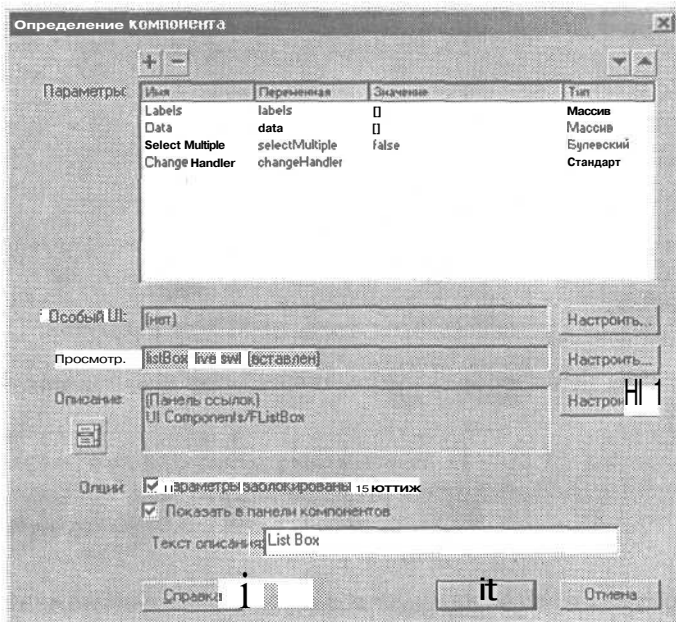
component.registerSkinElement (shadow_mc, "shadow");
component.registerSkinElement (darkshadow_mc, "darkshadow");
component.registerSkinElement (highlight_mc, "highlight");
ramponent.registerSkinBement (highlight3D_mc, "highlight3D");

```

registerSkinElement это метод, позволяющий компоненту перенять свойства, установленные для объекта globalStyleFormat. Изменением свойств такого глобального объекта можно менять соответствующие свойства всех компонентов (например, цвет, как это показано на рисунке):

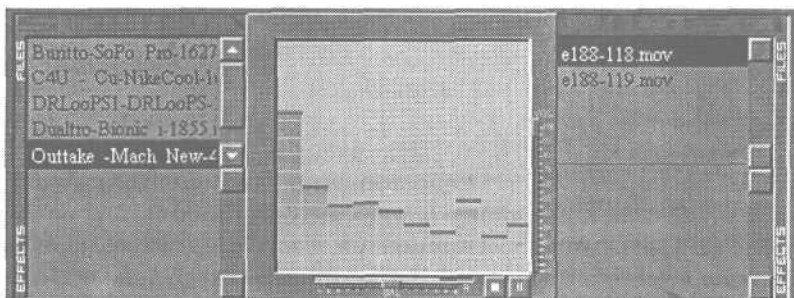


Одно из больших преимуществ использования компонентов связано с панелью Определение компонента (Component Definition). Для ее открытия нужно выбрать один из компонентов, а затем щелкнуть на пункте Определение компонента из меню Библиотеки. На этой панели можно задать определенные значения переменным или предоставить пользователю возможность выбора из массива опций, поэтому каждый экземпляр данного компонента может быть легко настроен:

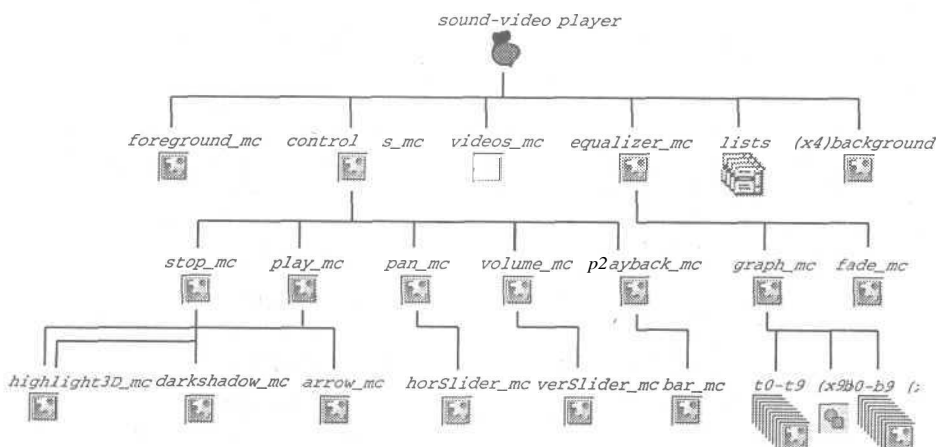


Компонент проигрыватель

И вот, наконец, наступил момент, которого вы ждали: в оставшейся части данной главы мы попробуем создать новый компонент из проигрывателя звуковых и видео объектов. Речь идет о мультимедийном проигрывателе для загрузки файлов видео и MP3. В центральном окне просмотра будет отображаться фильм, а в случае звуковых файлов — визуальное представление звука в форме графического эквалайзера:



Структура данного компонента создана с использованием объектов. Они организованы таким образом, что наглядно иллюстрируют как визуальную, так и программную стратегию классификации объектов, что мы уже применяли при объектно-ориентированном программировании нашего плеера:



В дальнейшем, когда все объекты будут созданы, вы в любой момент можете обратиться к показанной диаграмме для их правильного разме-

щения и создания связей между ними, а также для подгонки их внешнего вида в соответствии с общим видом проигрывателя. На рисунке указаны все объекты и необходимые для их создания графические элементы (которые, впрочем, вы при желании можете изменить для разработки своего дизайна). Весь программный код и исходные файлы для этой главы можно скачать с www.friendsofed.com.

Трудности при объектно-ориентированной разработке могут возникнуть из-за того, что зачастую объекты по отдельности, то есть вне связи с другими объектами из показанного дерева взаимодействий, работают очень плохо или совсем не работают. Поэтому нам придется действовать почти наудачу, так как проверить работоспособность объектов мы сможем только после создания всего фильма целиком.

Разработку начнем с «малых» объектов, таких как кнопки **stop** и **play**, регуляторы **pan** и **volume**, индикатор **playback**, а также **fader** и **graph**. Затем мы их объединим в два объекта **controls** и **equalizer**. И, наконец, из двух последних мы создадим наш **sound-video player**.

Фильм для кнопки stop

Сразу же возьмемся за дело.

1. Откройте новый фильм и создайте в нем новый клип с именем stop.
2. Переименуйте слой в graphics, и в первом кадре на сцене нарисуйте то, что, по-вашему, может быть кнопкой (или можно скопировать нашу кнопку). При этом ее правый верхний угол должен иметь координаты (0,0):



3. Вставьте во второй кадр слоя graphics ключевой кадр и немного видоизмените изображение кнопки так, чтобы было понятно, что она находится в нажатом положении:



4. Вставьте новый слой с названием actions, добавьте во второй кадр ключевой кадр и подключите к нему действие stop.

Кнопка стоп является одним из основных объектов для звуковых и видеопроигрывателей. Тем не менее, она сама тоже в свою очередь состоит из объектов, а точнее — из клипов фильмов. Все они уже автоматически помещаются в библиотеку при перетаскивании компонента `ListBox` на сцену. Параметры цвета для этих клипов, как это было

выше отмечено, будут соответствовать значениям, установленным для объекта `globalStyleFormat`.

5. Откройте панель Компоненты и перетащите компонент `ListBox` на сцену. Теперь выделите его и удалите. При этом все клипы, необходимые для кнопок, останутся в Библиотеке.
6. На панели Действия подключите к первому кадру слоя `layer` следующий код:

```
function pressActions () {
    this.gotoAndStop (2);
}
function releaseActions () {
    _parent._parent.stopS ();
    this.gotoAndStop (1);
    _parent.play_mc.gotoAndStop (1);
}
```

Здесь функция `pressActions` переводит головку проигрывателя на второй кадр клипа так, чтобы отобразить рисунок, соответствующий нажатой кнопке. Затем функция `releaseActions` вновь переносит временную шкалу клипа `stop` в начало, то есть в первый кадр. То же происходит и со шкалой родительского клипа. Строка `_parent.stops()`; запускает в проигрывателе метод, который будет отвечать за остановку воспроизведения.

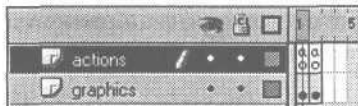
Во *Flash MX* для создания кнопки можно использовать не только объект *Button*. Теперь сами клипы могут реагировать на щелчки пользователя так же, как обычные кнопки. Именно этим мы только что и воспользовались в клипе *stop*.

7. К первому кадру слоя `layer` подключите следующий код:

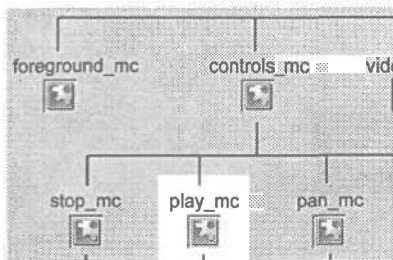
```
this.onPress = pressActions;
this.onRelease = releaseActions;
this.onReleaseOutside = releaseActions;
stop ();
```

Как видите, теперь в ответ на событие `onPress` будет включаться функция `pressActions`, а на события `onRelease` и `onReleaseOutside` — функция `releaseActions`.

Вот и все, мы создали первый объект проигрывателя. Перед тем, как продолжить работу, проверьте, совпадают ли ваши установки для временной шкалы с нашими:



Фильм для кнопки play



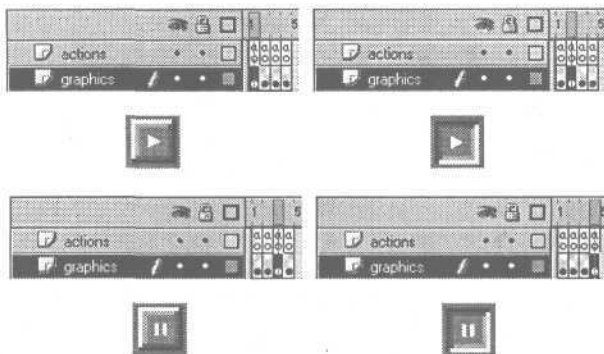
Теперь попробуем создать кнопку play для управления воспроизведением файлов проигрывателем. При этом она должна выполнять два действия, то есть еще в качестве кнопки pause. При воспроизведении файла нажатием кнопки play пользователь сможет временно приостановить его проигрывание, а при повторном нажатии — возобновить воспроизведение, как в обычных CD или DVD проигрывателях.

Создание клипа play во многом схоже с клипом stop, за исключением того, что теперь он должен вдобавок содержать кнопку паузы. При отпускании кнопки play головка проигрывателя будет переходить на изображение кнопки паузы.

Так как клип play также содержит кнопку паузы, то нужно использовать выражение if для проверки того, что кнопке в данный момент соответствует кадр кнопки play или кадр кнопки pause.

1. Создайте новый клип с именем play и добавьте во временную шкалу новый слой. Как и в случае с кнопкой stop, назовите слои actions и graphics.
2. В первом кадре слоя graphics нарисуйте кнопку воспроизведения, проверив при этом, что ее верхний левый угол имеет координаты (0,0).
3. Вставьте в каждый слой по три ключевых кадра, а затем во втором кадре слоя graphics перерисуйте кнопку так, чтобы было видно, что она выделена.
4. В третьем кадре того же слоя измените рисунок на изображение кнопки паузы в не нажатом положении.
5. Затем в кадре 4 немного видоизмените ее, как при нажатии кнопки.

Для ясности можно посмотреть на рисунке все четыре изображения и соответствующие им параметры временной шкалы:



6. К первому кадру слоя actions подключите следующий код для функций `pressActions` и `releaseActions`:

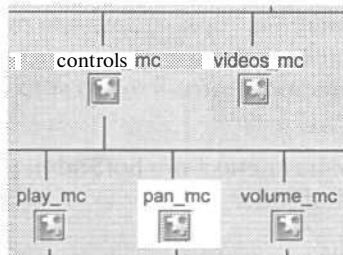
```
function pressActions () {
    if (this._currentframe == 1) {
        this.gotoAndStop (2);
    } else if (this._currentframe == 3) {
        this.gotoAndStop (4);
    }
}

function pressActions () {
    if (this._currentframe == 2) {
        _parent._parent.playS ();
        this.gotoAndStop (3);
    } else if (this._currentframe == 4) {
```

7. К оставшимся трем кадрам слоя actions подключите действие `stop`.
Временная шкала для клипа `play` должна выглядеть так:



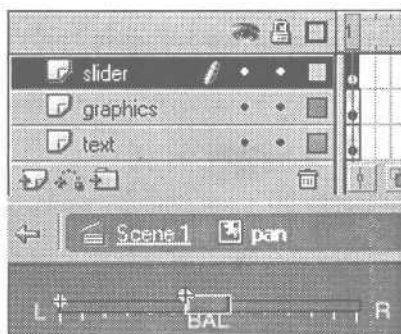
Фильм pan



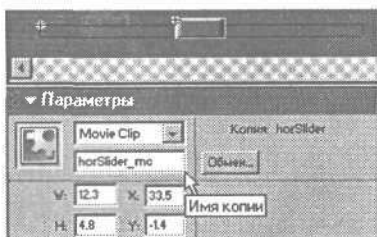
Пойдем дальше и займемся созданием регулятора pan, с помощью которого пользователь сможет устанавливать баланс звука между правым и левым каналами.



1. Как и прежде, начните с создания нового клипа с именем pan.
2. Переименуйте слой в graphics и нарисуйте на его сцене тонкий пустой прямоугольник шириной около 80 пикселей. Он будет отображать дорожку для движения бегунка регулятора.
3. Добавьте новый слой с именем text и с помощью инструмента Текст, находящегося на стандартной панели, поместите по краям созданной дорожки буквы 'L' и 'R', как показано на рисунке. Вооружитесь воображением, например, для нанесения меток вдоль дорожки.
4. Вставьте еще один новый слой с названием slider и нарисуйте в первом кадре фигуру, которую будущий пользователь будет перемещать в качестве бегунка регулятора баланса звука:



5. Затем выделите изображение бегунка и преобразуйте его в клип (Вставить > Преобразовать в символ) под названием horSlider (сокращенно от horizontal slider — горизонтальный бегунок, в отличие от vertical slider — вертикального бегунка, который мы создадим для управления громкостью звука). Не забудьте проверить, что его верхний левый угол имеет координаты (0,0).
6. Во временной шкале фильма рап выделите на сцене клип horSlider, и на панели Параметры задайте имя копии horSlider_mc:



Добавление символов '_mc' к концу имени копии фильма в Flash MX приводит к тому, на панели Действия при введении точки сразу после имени копии будет появляться всплывающая подсказка со свойствами данного объекта MovieClip.

Теперь добавим немного ActionScript, чтобы все это заработало.

7. Создайте новый слой actions и в его первом кадре введите следующий код:

```
function getPan () {
    var x = horSlider_mc.x;
    if (x>=35) {
        (pan = (x-35)/35);
    } else {
        (pan= - (35-x)/35);
    }return pan;
}
function pressAction () {
    horSlider_mc.startDrag (false, 0, horSlider_mc.y, 69, horSlider_mc.y);
    this.onEnterFrame = function () {
        _parent._parent.adjPan (getPan 0*100);
    };
}

function releaseActions () {
```

```

horSlider mc.startDrag ();
}
this.onPress = pressAction;
this.onRelease = releaseAction;
this.onReleaseOutside = releaseAction;

```

Вы, вероятно, уже заметили, что структура этого программного кода аналогична кодам фильмов `stop` и `play`. Главная функциональная разница заключается в том, что бегунок управления балансом звука, то есть клип `horSlider`, должен быть подвижным. Кроме того, нужно определять долю расстояния от текущего положения бегунка до правого или левого конца дорожки для того, чтобы объект `Sound` после обработки этих данных соответствующим образом установил баланс звука между каналами.

Позже мы еще раз воспользуемся данным приемом при создании регулятора громкости объекта `Sound`.

Теперь давайте рассмотрим в деталях, как это будет работать.

Во введенном вами коде для фильма `pan` в функции `pressAction` есть следующая строка:

```
horSlider_mc.startDrag (false, 0, horSlider_mc._y, 69, horSlider_mc._y);
```

Интервал, вдоль которого пользователь может перемещать бегунок, ограничивается числами 0 и 69. Это означает, что клип `horSlider` можно будет перемещать вдоль оси X в диапазоне от 0 до 69. Так как сам регулятор `pan` имеет горизонтальное положение, то мы должны исключить возможность перемещения бегунка по вертикали. Это достигается установкой одного и того же значения `horSlider_mc._y` верхней и нижней координатам.

Сразу после указанной строки записана следующая функция:

```

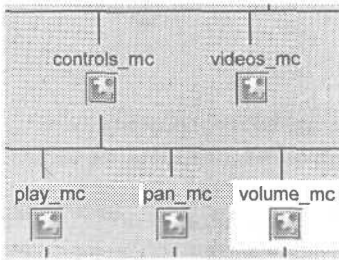
this.onEnterFrame = function () {
    _parent._parent.adjPan (getPan 0*100);
};

```

В ней обработчик `enterFrame` передает данные о текущем положении бегунка в родительский клип `_parent._parent` — как вы помните, им является сам создаваемый нами проигрыватель (можно освежить все в памяти, вернувшись к иерархической структуре в самом начале данного задания). Следующая функция `getPan` непосредственно вычисляет это значение в процентном отношении и возвращает его.

Во Flash MX появилась возможность установки обработчиков событий в фильмах с любой временной шкалы, как в нашем случае с обработчиком `onEnterFrame`.

Фильм Volume



Теперь попробуем создать еще один объект в виде регулятора, который также станет частью пользовательского интерфейса. Он будет очень похож на только что созданный нами клип регулятора баланса звука pan за исключением того, что он должен располагаться вертикально и позволять пользователю управлять **громкостью** воспроизводимого звука.



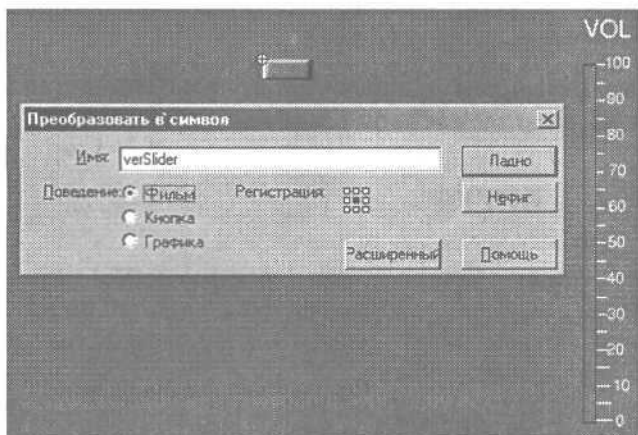
1. Создайте новый фильм, преобразуйте его в символ, назовите `volume` и верхний левый угол поместите в точку $(0,0)$.
2. Переименуйте имеющийся слой в `graphics` и нарисуйте на нем тонкий пустой вертикально расположенный прямоугольник высотой 91 пикселей, который станет дорожкой движения бегунка. Верхний левый угол его должен по оси y иметь координату 21.

Это небольшое пространство в отличие от регулятора `pan` мы оставляем для помещения надписи «VOL» (громкость) над прямоугольником.

3. Добавьте новый слой, назовите его text и с помощью инструмента Текст введите 'VOL' в верхней части регулятора.
4. На этом же слое text нанесите шкалу с цифрами вдоль дорожки для оценки уровня громкости:



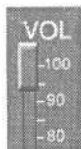
5. Затем вставьте новый слой с именем slider и убедитесь, что он является верхним слоем на вашем монтажном столе. На сцену первого кадра этого слоя перетащите копию фильма horSlider из библиотеки.
6. Выделите horSlider и нажмите F8 для преобразования в символ типа фильм, и назовите его verSlider:



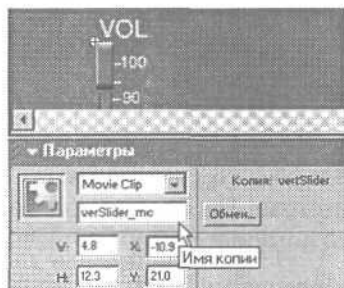
7. Дважды щелкните на фильме verSlider на сцене для перехода в режим Править символы (Edit Symbols). Сочетанием клавиш

Ctrl + Shift + 9 поверните его на угол 90 градусов вдоль часовой стрелки так, чтобы он стал похож на вертикальный бегунок. При этом его верхний левый угол по-прежнему должен иметь координаты (0,0).

- Вернитесь в режим правки для фильма volume и установите бегунок verSlider в верхней части дорожки регулятора:



- Выделите verSlider, и на панели Параметры введите имя копии verSlider_mc:



Это весь графический материал, который нам нужен. Пойдем дальше — воспользуемся ActionScript для того, чтобы все нарисованное нами позволило управлять громкостью звука.

- Вставьте новый слой с названием actions. Откройте панель Действия (F9) и, выделив первый кадр слоя actions, введите следующий код:

```
function getVol () {
    return (112-verSlider_mc._y)/91)*100;
}
```

Как видите, к фильму volume мы применили функцию getVol в противоположность уже использованной нами функции getPan. Во второй строке вычисляется отношение текущего значения свойства `_y` для `verSlider_mc` ко всей длине дорожки.

- Теперь тут же добавьте следующий код:

```
function pressActions () {
    verSlider_mc.startDrag (false, verSlider_mc._x, 21, verSlider_mc._x, 112);
```

```

    this.onEnterFrame = function () {
        _parent._parent.adjVol(getVol ());
    };
}

```

Здесь функция `startDrag` устанавливает ограничения на возможные перемещения объекта. В данном случае пользователь не может передвигать `verSlider_mc` вдоль горизонтальной оси, в то время как по вертикали его координата может меняться от 21 до 112.

12. Наконец, последняя часть, которую нужно добавить:

```

function releaseActions () {
    verSlider mc.stopDrag ();
}
this.onPress = pressActions;
this.onRelease = releaseActions;
this.onReleaseOutside = releaseActions;

```



Есть еще один объект, который нам осталось создать перед тем, как собрать все средства управления вместе. Речь идет об индикаторе воспроизведения, то есть о горизонтальной полосе, увеличивающей свою длину по мере проигрывания звукового или видео файла. Он дает пользователю возможность видеть, какая часть файла уже пройдена, и какая — осталась:



Индикатор воспроизведения

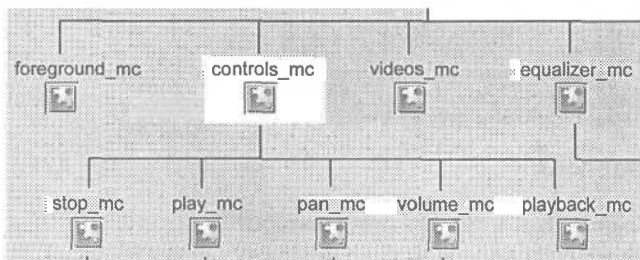
1. Создайте новый символ типа фильм под названием `soundLength` и поместите его, как и раньше, в левом верхнем углу сцены.
2. Переименуйте слой в `actions`, а затем вставьте два новых слоя с именами `edges` и `length`.
3. В первом кадре слоя `edges` нарисуйте тонкий пустой прямоугольник, который будет контуром, внутри которого должен перемещаться

- индикатор. Установите для него ширину 100 пикселей, высоту 3 пикселя, а координаты верхнего левого угла — (0,0).
4. В первом кадре слоя `length` нарисуйте очень маленький прямоугольник в виде штришка высотой 2,3 и шириной 1 пиксель непосредственно с внутренней стороны левого края уже имеющегося контура.
 5. Выделив этот маленький прямоугольник, с помощью клавиши F8 преобразуйте его в символ типа фильм под именем 'length'. Убедитесь, что его верхний левый угол расположен в точке (0,0).
 6. Вернитесь к фильму `soundLength` и откройте панель Параметры (Ctrl + F3). Введите имя копии `bar_mc`.
 7. И, наконец, примените действие `stop` к первому кадру слоя `actions`.

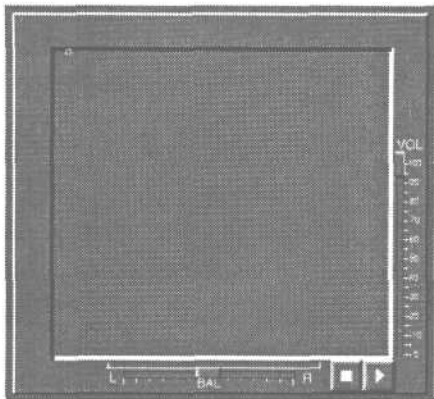
Вам, должно быть, интересно, почему индикатор (фильм `length`) имеет такую незначительную ширину — он почти незаметен, не так ли? Он не виден, когда никакой файл не воспроизводится, но при воспроизведении Flash будет изменять свойство `_width` (ширина) копии `bar_mc` объекта `length` в соответствии с долей уже проигранной на данный момент части файла. Поэтому ширина прямоугольника, то есть фильма `length`, будет меняться так, чтобы отображать, какая часть воспроизводимого файла уже пройдена.

Соответствующий код ActionScript к клипу `length` мы подключим позже, при окончательной сборке фильма `sound-video player`. А сейчас соберем вместе все объекты, которые мы до этого момента создали для получения пользовательского интерфейса.

Создание клипа `controls`



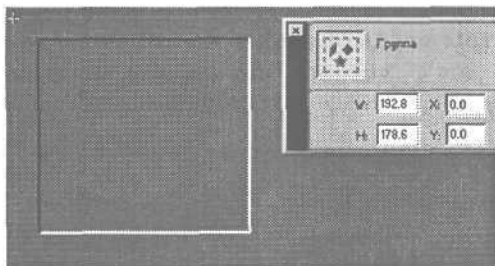
После данного раздела преимущества объектно-ориентированной разработки станут еще более очевидными. Мы соберем вместе все созданные в этой главе объекты в один законченный фильм **controls**, который будет выполнять функции пользовательского интерфейса для нашего проигрывателя:



1. Начните с создания нового символа с именем 'controls'.
2. В этом фильме controls к установленному по умолчанию слою добавьте еще 5 новых, и назовите их 'actions', 'play_stop', 'pan', 'volume', 'indicatorbar' и 'viewer' по порядку снизу вверх.

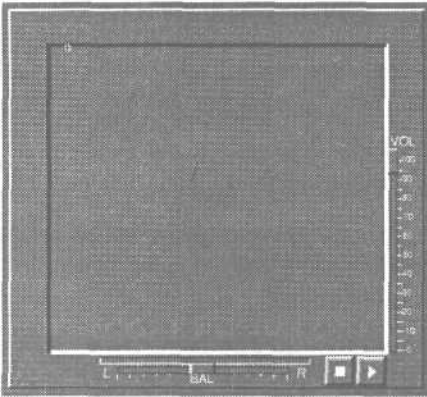
Перед размещением объектов на сцене нужно определиться с формой и размерами всего проигрывателя в целом так, чтобы затем рационально решить вопрос о положении каждого из них. Поэтому в начале нарисуем окно просмотра для компонента player.

3. В первом кадре слоя viewer нарисуйте контур в виде окна, в котором фильм будет воспроизводиться. Форму контуру можете задать любую по желанию, а можете воспользоваться нашим файлом (с размерами и координатами) в качестве шаблона:



4. Затем перетащите из Библиотеки копии каждого из фильмов play, stop, pan, volume и soundLength в первые кадры соответственно названных слоев (soundLength нужно перетащить на слой indicatorbar).
5. Разместите объекты в окне просмотра. В нашем исходном файле они имеют координаты:

- play (157.8, 162.7)
- stop (142.2, 162.7}
- pan (36.0, 166.7)
- volume (176.2, 57.1)
- **indicator bar** (36.5, 163.7)



Теперь, когда объекты расставлены по местам, вновь обратимся к ActionScript, с помощью которого мы установим их связь компонентом sound-video player.

Так как к каждому из этих объектов мы будем обращаться из фильма sound-video player (являющегося для них родительским фильмом `_parent_parent`), то нужно задать для них имена копий.

6. С помощью панели Параметры введите имена копий объектов 'play_mc', 'stop_mc', 'pan_mc', 'volume_mc', а для фильма `soundLength` — 'playback_mc'.
7. И, наконец, в первом кадре слоя actions введите следующий код:

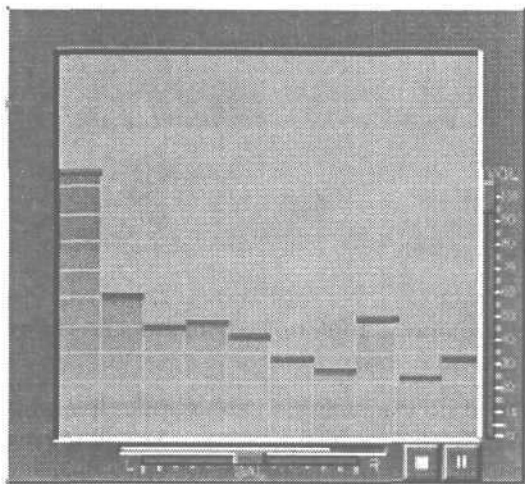
```
stop_mc._visible = false;
play_mc._visible = false;
stop {};
```

Он делает невидимыми кнопки play и stop в начале фильма. Эти кнопки становятся видимыми после того, как звуковой или видеофайл загрузится и будет готов для воспроизведения. Программный код, который это выполняет, будет добавлен позже, когда мы составим законченный компонент.

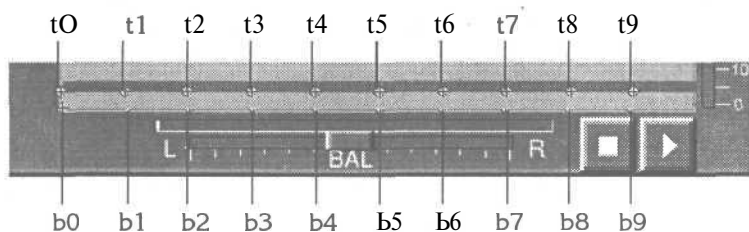
А теперь, после создания фильма `controls`, пойдём дальше — построим следующий объект нашего проигрывателя, а именно, **графический эквалайзер**.

Эквалайзер

Под эквалайзером (часто называемым графическим эквалайзером) понимают визуальное представление частотного диапазона проигрываемого в данный момент звукового файла. В нашем случае мы будем создавать не совсем то, что имеется в виду, и что вы увидели бы на своей аудиосистеме. Тем не менее, это выглядит достаточно эффектно при подключении `ActionScript`. В итоге мы должны примерно следующее:

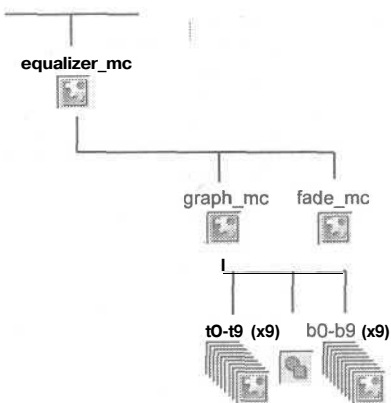


Ядро эквалайзера состоит из 20 фильмов. Десять фильмов будут отображать верхние колонки, а еще десять — нижние. Если вы обратитесь к нашему файлу, то заметите, что они очень маленькие, как это показано на рисунке, однако их высота будет изменяться в соответствии с командами `ActionScript` в фоновом режиме одновременно с воспроизводимой музыкой.



Объекты для эквалайзера

Начнем с создания **клипов top и bar**, а затем просто снимем с них по несколько копий с размещением вдоль нижней кромки родительского объекта **bar graph**. В итоге мы должны получить примерно следующую иерархическую цепочку объектов:



Итак, поехали.

1. Создайте новый фильм под названием 'top', поместив его в верхнем левом углу сцены.
2. В первом кадре установленного по умолчанию слоя нарисуйте пустой прямоугольник размером $15 \times 2,5$, указав при этом для его верхнего левого угла координаты (0,0).
3. Создайте в верхнем левом углу еще один новый фильм и назовите его bar.
4. Нарисуйте прямоугольник размером 15×1 .
5. Откройте Микшер цвета (Shift + F9) и измените значение Альфа на 60%.
6. Вставьте новый слой с именем 'actions', и в его первом кадре введите следующую функцию:

```

function getNum (n) {
    return n._name.substr(1,1)
}
  
```

Как мы уже говорили, с фильма `bar` будут сняты копии так, что все десять похожих объектов разместятся вдоль всей ширины эквалайзера. При этом каждой копии мы дадим соответствующие имена `b0`, `b1`, `b2` и так далее до `b9`. Введенная нами функция `getNum(n)` путем обращения ко второму символу в имени копии фильма `bar` определяет ее номер.

7. Теперь следом введите еще одну функцию:

```
function inv (n) {
    var x = int 9getNum (n);
    return (_parent.maxHeight/2)*(1 - (1/(x+1)));
}
```

Она вычисляет значение `_u` для высоты фильма `bar` и координату `_y` для соответствующего фильма `top`. Теперь при воспроизведении звукового файла верхние края этих фильмов образуют гладкую кривую, снижающуюся слева направо, как бы напоминая стереосистему, в которой усилено звучание басов по сравнению с высокими частотами.

8. Далее добавьте следующие строки:

```
function adjBar () {
    //adjust bottoms
    var startY = this._y;
    var dis = -startY - _parent.amplitude+adj;
    (!_parent.stopPlay) ? (this._y += dis*_parent.frc+random (11) - 5) : this._y;
```

Здесь выполняется вся работа по изменению высоты фильма `bar` пропорционально амплитуде (громкости) воспроизводимого звука. Если вы выделите на сцене прямоугольник, изображающий объект `bar`, и увеличите значение его высоты на панели Параметры, то заметите, что размер возрастет с его нижней стороны.

Функция `adjBar` вначале вычисляет новое значение высоты для `bar`, а затем определяет, каким образом его разместить так, чтобы нижний край клипа по-прежнему оставался выровненным вдоль прямой. Она прибавляет значение переменной `dis` (расстояние) к высоте клипа, если эта высота находится ниже текущей амплитуды, или вычитает единицу, если она находится ниже текущей амплитуды.

Вы можете спросить: зачем также определять значение `random(11)-5` в качестве части высоты. Это позволяет изменять координату `_y` случайным образом в диапазоне 5 пикселей около истинного значения `_y`, что дает больший эффект реалистичности при воспроизведении.

9. Теперь введите еще несколько строк:

```
if (_parent.stopPlay || (this._y>0)){
    this._y = 0;
}
this._height = -this._y;
```

В них проверяется, продолжается ли воспроизведение, а также, не сместилась ли высота фильма `bar` ниже выравнивающей прямой (это происходит, когда значение `_y` становится больше нуля).

Значение `if` будет истинно, если оба названных условия выполняются. Поэтому в обоих случаях координата `_y` принимает значение 0, то есть фильм становится невидимым. Если ни одно из условий не выполняется, то последняя строка просто меняет знак переменной `_height` фильма `bar` на противоположный так, чтобы изображение объекта `bar` стало видимым.

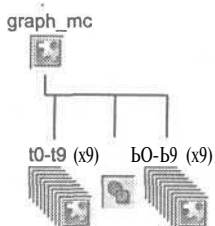
10. И, наконец, осталось ввести последнюю часть кода для фильма `bar`:

```
//adjust tops
if (this._y<_parent ["t"+num]._y) {
    _parent ["t"+num]._y = this._y;
    step = 0;
} else {
    _parent ["t"+num]._y += step;
    step += .90;
}
}
adj = inv (this);
num = getNum (this);
this.onEnterFrame = adjBar;
```

Здесь мы просто изменяем соответствующий фильму `bar` клип `top` таким образом, чтобы он стал равным по размеру с прямоугольником `bar` в том случае, если высота `bar` больше высоты `top`. Аналогично, клип `top` опускается вниз по направлению к `bar`, когда высота `bar` становится меньше высоты `top`.

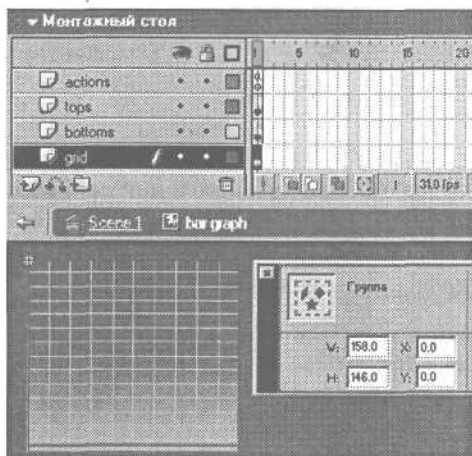
Эффект падения создан так, что `top` движется по направлению к `bar` на большие расстояния при каждом выполнении кода. Последняя строка `this.onEnterFrame = adjBar` применяет функцию `adjBar` к каждому кадру.

Изображение для bar



Попробуем теперь создать новый фильм в виде фоновой сетки, и поместить на нем наши фильмы bar и top так, чтобы все это выглядело наподобие обычных графических эквалайзеров.

1. Как обычно, начните с создания нового символа типа фильм с размещением в верхнем левом углу. Назовите его 'bar graph'.
2. К имеющемуся слою добавьте три новых. Назовите их по порядку 'actions', 'tops', 'bottoms' и 'grid'.
3. В первом кадре слоя grid нарисуйте сетчатую область так, чтобы это было похоже на фон эквалайзера. Если вы используете наши файлы, то на рисунке показаны координаты.



4. Выделите первый кадр слоя bottoms и перетащите на него из Библиотеки десять копий фильма bar. С помощью команды Выравнивание (Ctrl + K) выровняйте их вдоль нижнего края фоновой сетки так, чтобы каждый из них занимал в сетке отдельную ячейку:



Ранее говорилось о том, что компонент проигрыватель будет управлять данными клипами в соответствии с амплитудой воспроизводимого звукового файла. Поэтому нужно каждому прямоугольнику дать название.

5. С помощью панели Параметры задайте каждому клипу, начиная с самого левого, имена `b0`, `b1`, `b2` и так далее до `b9`.
6. Выделите первый кадр слоя `tops` и повторите все действия — перетащите из Библиотеки десять копий фильма `top` и выровняйте их вдоль прямой непосредственно над клипами `bar`:



7. Затем также задайте им имена `t0`, `t1`, `t2`, ..., `t9`.
8. Выделите первый кадр слоя `actions` и откройте панель Действия. Добавьте следующий код:

```
maxHeight = 140;
minHeight = 5;
fric=.10;
```

В первых двух строчках устанавливаются наибольшее и наименьшее возможные значения высоты для подвижных прямоугольников эквалайзера. Это сделано для того, чтобы они не выходили за пределы фоновой сетки. В третьей строчке задается значение трения.

9. Теперь введите следующую функцию:

```
function stopPlaying () {
    this.stopPlay = true;
}
```

Функция `stopPlaying` запускает фильм `bar graph` для установления нулевого значения амплитуды звукового сигнала каждый раз, когда останавливается воспроизведение `graph`. Она доступна также вне символа `bar graph`.

10. Далее введите еще несколько строк:

```
function stopPlaying () {
    stopPlay = false;
    n=0;
    this.onEnterFrame=getAmp;
}
```

Здесь функция `startPlaying` аналогична функции `stopPlaying` в том смысле, что она тоже является общедоступной. Разница в том, что переменная `stopPlay` теперь имеет значение `false`. Кроме того, событие `onEnterFrame` присваивается функция `getAmp`.

11. Последняя часть программы:

```
function getAmp () {
    if (!(n%4) && !stopPlay) {
        amplitude = random (maxHeight-minHeight)+minHeight;
    } else if (!stopPlay) {
        amplitude--;
    } else {
        amplitude=0;
        delete this.onEnterFrame;
        this.gotoAndStop (1);
    }
    n++;
}
this.onEnterFrame = getAmp;
```

Функция `getAmp` устанавливает случайным образом значения амплитуды и частоты для звуковых файлов. Во-первых, при каждом выполнении тела функции значение переменной `p` увеличивается на единицу до тех пор, пока продолжается воспроизведение звука, а при остановке проигрывания ей присваивается значение 0. Во-вторых, проверяется, не равен ли нулю модуль `p`, а также что `stopPlay` имеет значение `false`.

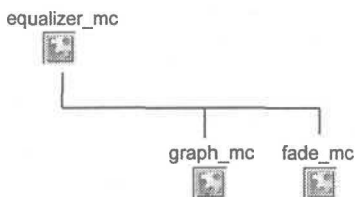
Под модулем числа здесь имеется в виду остаток от деления его на другое число. Он записывается символом %.

С помощью нахождения остатка можно задавать номера кадров, в которых будет выполняться каждый последующий шаг цикла. В нашем случае это будет происходить в каждом четвертом кадре.

Постепенное появление

Теперь займемся созданием клипа с эффектом плавного перехода, когда пользователь переключается от прослушивания звукового файла к просмотру видео или обратно. Для осознания того, как это будет работать, представьте пользователя, который в данный момент слушает музыку. До тех пор, пока он этим занимается, графический эквалайзер будет активным, как бы отображая частотные параметры звука. А теперь представьте, что он захотел посмотреть видео.

Здесь нам поможет фильм `fade` в качестве визуального перехода, при котором одновременно с постепенным «растворением» эквалайзера на первый план будет выходить экран видео. Точно такой же плавный переход должен происходить и в обратную сторону. На нашей иерархической схеме данный фильм занимает показанное на рисунке место:



1. Создайте в верхнем левом углу новый символ типа фильм под названием 'fade'.
2. В первом кадре установленного по умолчанию слоя нарисуйте заливкой черным цветом прямоугольник шириной 158 и высотой 146 пикселей с размещением его верхнего левого угла в точке (0,0).
3. Вставьте новый слой с именем 'actions'. Откройте панель Действия (Actions), и в первом кадре введите следующий код:

```

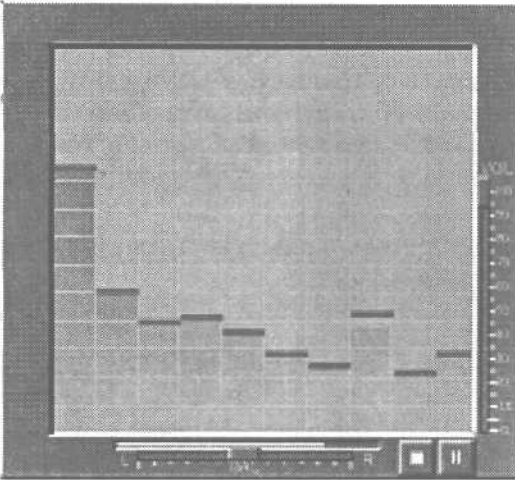
function fadeout () {
    (this.alpha > 0) ? (this._alpha - = 5) : (delete this.onEnterFrame);
}
function fadeIn () {
    this.onEnterFrame = function () {
        if (this._alpha < 100) {
            this._alpha += 5;
        } else {
            _parent.gotoAndStop (1);
            delete this.onEnterFrame;
        }
    };
};
  
```

```
}  
this.onEnterFrame = fadeout;
```

Это все что нужно для клипа `fade`. А теперь перейдем к фильму с эквалайзером и применим там созданный нами эффект `fade`.

Фильм `equalizer`

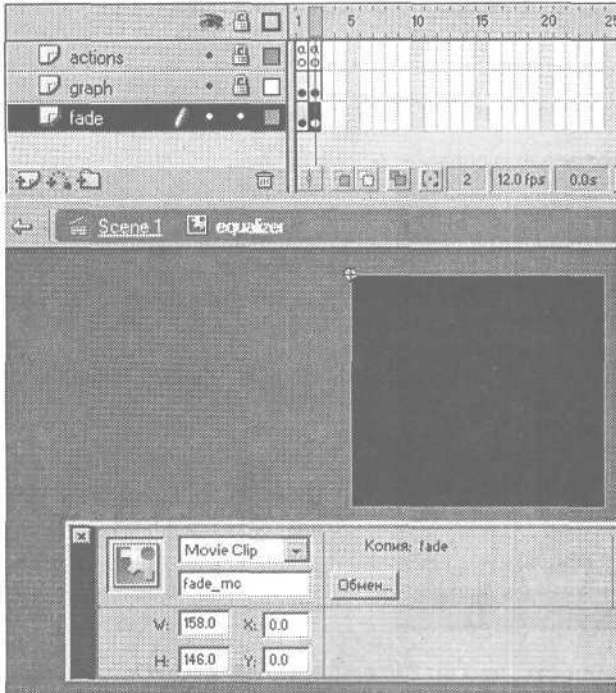
Не забыли, как будет в итоге выглядеть наш эквалайзер?



Тем не менее, нужно учесть, что он будет работать далеко не всегда. То есть иногда в том же пространстве, в котором отображается эквалайзер, будет происходить воспроизведение видео. Поэтому мы сделаем так, чтобы фильм эквалайзера состоял из двух картинок, причем родительский клип, то есть компонент проигрыватель, будет иметь возможность выбирать одну из них в зависимости от типа проигрываемого файла: звукового или видео.

1. Создайте в верхнем левом углу сцены новый символ типа фильм с названием `'equalizer'`.
2. Добавьте к существующему слою два новых, и назовите их по порядку `'actions'`, `'graph'` и `'fade'`.
3. В первом кадре слоя `fade` нарисуйте в верхнем левом углу черный прямоугольник шириной 158 и высотой 146 пикселей.
4. Во второй кадр этого же слоя вставьте ключевой кадр и удалите на нем нарисованный вами прямоугольник.

- Оставаясь во втором кадре, откройте Библиотеку и перетащите копию фильма fade на сцену. Ее верхний левый угол поместите в точку (0,0) и введите имя копии fade_mc.



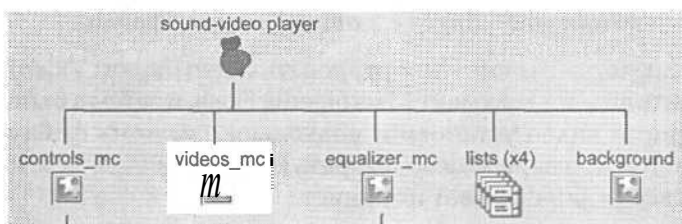
- Во второй кадр слоя graph вставьте ключевой кадр и перетащите в него фильм bar graph из вашей Библиотеки. Фильм должен располагаться в верхнем левом углу.
- Откройте панель Параметры и введите для bar graph имя копии graph_mc.

Действия для фильма equalizer, к счастью, будут очень простыми. Родительский фильм с помощью действия gotoAndStop будет определять, какой из двух кадров воспроизводить в фильме equalizer в зависимости от того, проигрывается звуковой файл или видео. Таким образом, все, что нам нужно, это сохранять выбранный кадр видимым после того, как родительский фильм сделал свой выбор.

- Вставьте ключевой кадр (F6) во второй кадр слоя actions. Откройте панель Действия (F9) и примените действие stop к обоим ключевым кадрам.

Мы уже близки к концу, но осталась еще одна немаловажная вещь: где видео?

Файлы видео

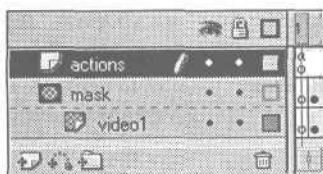


Фильм `videos` действует наподобие прототипа всех видеофайлов, которые мы хотим использовать. В нем будут содержаться отдельные фильмы, названные соответственно `video 1`, `video 2` и т. д. — по одному фильму для каждого используемого нами файла видео.

Подключение видео

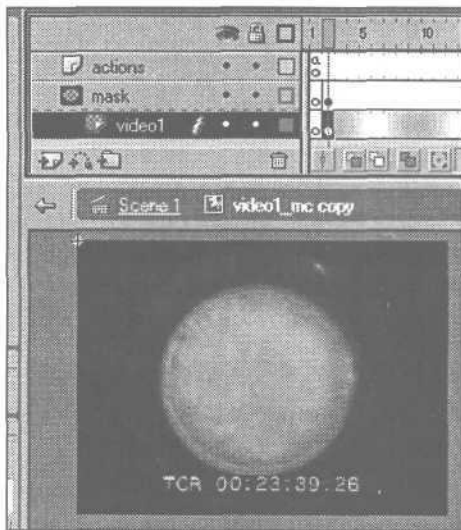
Для рассмотрения того, как это делается, мы поместим в фильме `videos` два файла видео, но вы при желании можете увеличить их число. В этом смысле вам предоставляется полная свобода действий, однако если вы хотите придерживаться нашего примера, то нужно воспользоваться двумя файлами MOV из прилагающегося программного материала.

1. Создайте новый символ типа фильм с названием 'video Г'.
2. Вставьте два новых слоя и назовите их по порядку 'actions', 'mask' и 'video Г'.
3. Дважды щелкните на слое `mask` и в появившемся диалоговом окне Настройки слоя (Layer Properties) установите кнопку выбор Маска (Mask). В соответствующем окне для слоя `video 1` щелкните на кнопке выбора Маскировано (Masked). После этого ваш монтажный стол должен иметь вид, показанный на рисунке.



4. В первом кадре слоя `actions` примените действие `stop`, а во второй кадр вставьте ключевой кадр.

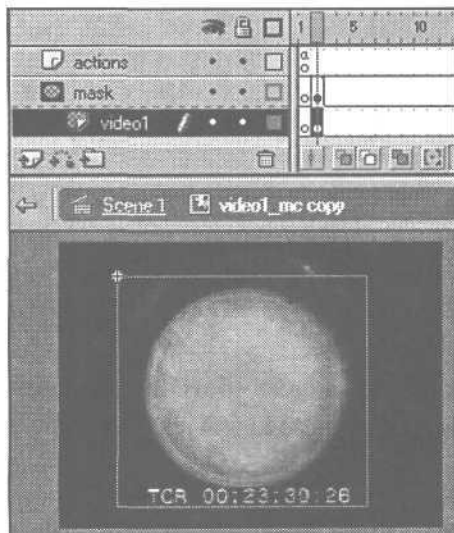
5. Другой ключевой кадр поместите во второй кадр слоя video 1 — здесь будут размещены наши видеофайлы.
6. Оставаясь в этом же втором кадре, произведите импорт необходимого файла. В нашем примере он называется e1-118.mov (он расположен в папке assets в списке прилагающихся к данному разделу файлов).
7. В открывшемся диалоговом окне Импортировать видео (Import Video) выберите Вставить видео в документ Macromedia Flash, и затем в окне Параметры импорта видео установите подходящие вашему файлу настройки. В нашем примере можно выбрать Качество 80%, так как исходный файл все равно староват и зернист.
8. После установки всех настроек нажмите ОК для импорта видео:



Пока видео имеет формат 240x180, то есть он слишком велик для нашего проигрывателя. Поэтому для того чтобы картинка не перекрывала фильм controls (кнопки управления) или еще что-то и соответствовала установленным нами размерам, мы наложим на изображение видео маску нужных нам ширины и высоты. Имейте в виду, что любой включенный, но не отображаемый видеоматериал, в том числе и лишние края изображения, в итоге сильно сказывается на размере файла Flash.

Если вы хотите, чтобы все изображение умещалось в области просмотра, тогда нужно изменить его масштаб в режиме Редактирования video 1 так, чтобы оно целиком было видно в окне просмотра.

9. Вставьте ключевой кадр во второй кадр слоя `mask`.
10. В этом кадре нарисуйте прямоугольник шириной 156 и высотой 144 пикселей. Можно приблизительно нарисовать его, а затем подогнать его размеры на панели `Параметры`.
Не забудьте поместить его в верхнем левом углу. Однако здесь нужно быть внимательнее с выравниванием маски и видео. Мы хотим, чтобы верхний левый угол изображения находился в точке (0,0) и точно подходил под область просмотра. Поэтому маску также нужно расположить в верхнем левом углу, а не подгонять под видео.
11. Выделите контур маски на сцене и на панели `Параметры` установите его верхний левый угол в точку (0,0).
12. Далее во втором кадре слоя `video 1` выделите видео и перетащите его на сцену так, чтобы оно располагалось непосредственно за маской независимо от координат (такой прием может быть полезным при временном отображении слоя `mask` в режиме разметки).



Мы должны настроить проигрыватель так, чтобы он переустанавливал `video 1` на первый кадр после того, как пользователь просмотрел его. Теперь видео готово к следующему запуску.

13. Найдите на временной шкале последний кадр вашего видео (в нашем примере это кадр 90), вставьте в него ключевой кадр на слое `actions` и введите следующий код:

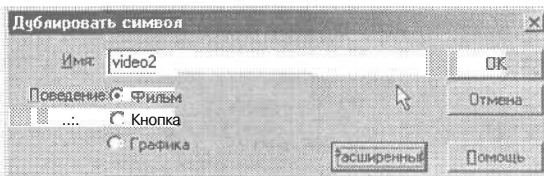

```
this.gotoAndStop (1);
```

14. Теперь вставьте на слое `mask` столько же пустых кадров, сколько их содержится в остальных двух слоях (90 в нашем случае). Не забудьте заблокировать все слои, чтобы маска нормально функционировала.

Это все, что нужно проделать с фильмом `video 1`. Теперь он будет находиться в Библиотеке и ждать того момента, когда мы внедрим в компонент проигрыватель.

А теперь мы создадим еще один фильм для другого видео, но уже с использованием заготовки `video 1`. Так что это не должно занять много времени.

15. Выделите в Библиотеке клип `video 1`, а затем выберите в меню библиотеки команду Дублировать и создайте копию с названием '`video 2`'.



16. Переименуйте слой `video 1` в `video 2` и удалите весь содержащийся в нем видеоматериал.

Теперь осталось просто выделить второй кадр и импортировать новое видео.

17. Щелкните на кадре 2 и выберите команду Импорт в меню Файл для импорта нового файла видео. В нашем примере это `e188-119.mov` (в той же папке `assets`).

18. Увеличьте или уменьшите число кадров в слоях в соответствии с импортированным видео. В нашем случае каждый слой должен содержать 122 кадра.

Итак, два клипа готовы. Повторяя проделанную процедуру, вы можете создать столько фильмов, сколько вам понадобится, просто увеличивая его номер в названии на единицу. Однако будем считать, что имеющегося материала нам вполне достаточно для сборки на его основе клипа **videos**.

19. Создайте новый символ типа фильм с названием '`videos`'.

Данный фильм имеет очень простую структуру. Каждый отдельный созданный нами клип будет находиться в первом кадре соответству-

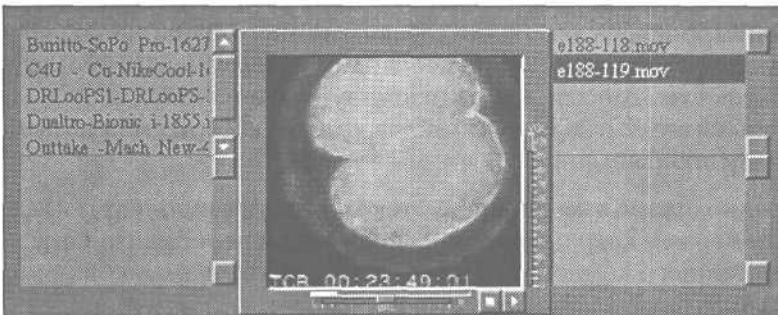
ющего слоя с тем же названием копии, но с добавлением в конце '_mc'. При этом никаких действий подключать не потребуется, так как все клипы будут вызываться непосредственно с проигрывателя.

20. Переименуйте установленный по умолчанию слой в 'video 1' и вставьте в его первый кадр копию фильма video 1.
21. С помощью панели Параметры поместите его в верхнем левом углу и ведите для него имя копии 'video1_mc'. Повторите эти действия на новых слоях с каждым нужным клипом.
22. В нашем примере процедура повторяется только один раз со вторым фильмом: вставьте новый слой с именем 'video 2' и поместите в его верхнем левом углу копию клипа video 2. Не забудьте ввести имя копии 'video2_mc'.

Вот и все. В заключение нам осталось собрать все кирпичики в один компонент проигрыватель.

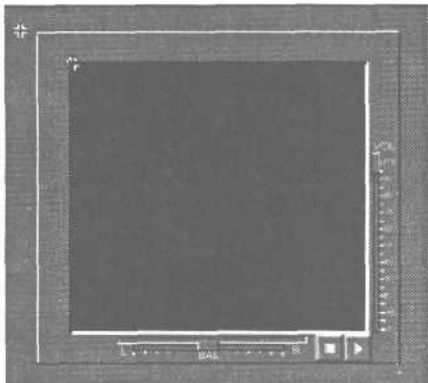
Окончательная сборка

Далее нам предстоит проделать довольно большой объем работы с применением ActionScript, однако после данной финальной стадии мы получим готовый для использования проигрыватель звуковых и видеофайлов, который можно будет подключить при разработке сайтов.



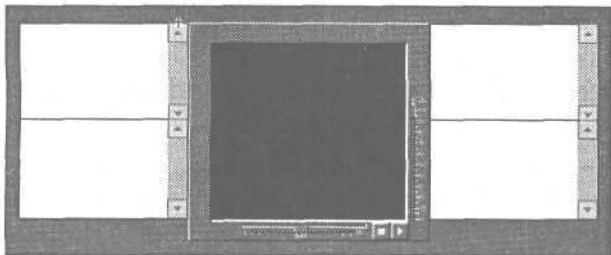
1. Начните с создания нового символа типа фильм с названием 'sound-video player'.
2. Добавьте к имеющемуся пять новых слоев. Назовите их сверху вниз 'actions', 'controls', 'videos', 'equalizer', 'scrollcomponents' и 'background'.
3. В первый кадр слоя controls перетащите из Библиотеки копию фильма controls. Установите для него координаты (9.2, 1.5) и задайте имя копии 'controls_mc'.

4. В первый кадр слоя equalizer перетащите копию фильма equalizer, расположив его в точках (27.5, 18.0). Введите имя копии 'equalizer_mc'.
5. Выделите кадр 1 слоя videos и перетащите на сцену копию фильма videos. Задайте координаты (27.5, 18.0) и имя копии 'videos_mc'.

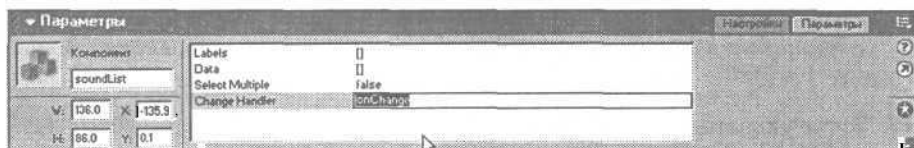


Далее воспользуемся встроенным компонентом Flash под названием **ListBox**. В интерфейсе проигрывателя мы применим его 4 раза: два для отображения воспроизводимых файлов и еще два для отображения звуковых и видео эффектов. В данном разделе мы просто вставим видео и звуковые файлы в первую пару компонентов **ListBox**, а ко второй паре эффектов **ListBox** добавим немного функциональности так, что позже вы можете при необходимости по аналогии создать другие эффекты. Здесь пока мы используем два из них для отображения списка видео и звуковых файлов, один из которых можно выбрать для воспроизведения.

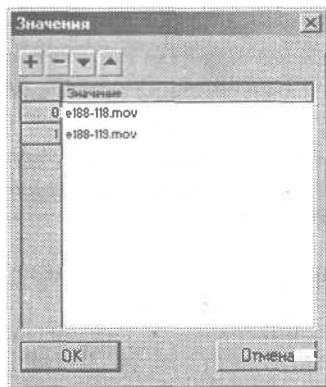
6. Итак, предположим, в вашей Библиотеке уже есть компонент **ListBox**. Выделите первый кадр слоя scroll components и перетащите четыре копии указанного компонента на сцену. Разместите их по бокам от центрального окна, как показано на рисунке:



7. На панели Параметры для каждого компонента установите ширину 136 и высоту 86 пикселей.
8. Выделите по очереди каждый элемент ListBox и введите имя копии: 'soundlist (верхний левый)', 'sound_effects (нижний правый)', 'videoList (верхний правый)' и 'video_effects(нижний правый)'.
9. Оставаясь на панели Параметры, введите в поле Change Handler (вкладка Параметры) для каждого компонента ListBox по очереди значение 'onChange'.



10. Выделите на сцене копию videoList, и на панели Параметры дважды щелкните в поле Labels. В открывшемся окне Значения с помощью кнопки со знаком плюса добавьте два поля и введите в них названия своих файлов видео:



Эти значения будут отображаться в компоненте ListBox, и у пользователя будет возможность выбрать один из файлов для воспроизведения. С копией soundList мы поступим немного иначе. Названия звуковых файлов будут динамически вызываться из текстового файла так, что Flash после анализа его содержимого будет отображать соответствующую информацию в компоненте.

11. Откройте установленный на вашем компьютере по умолчанию текстовый редактор, например, Notepad (Блокнот) или SimpleText, а затем введите следующее:

```
&soundFiles=
```

12. Если вы работаете с нашими примерами файлов, то сразу за данной строкой введите следующие. Помните, что при этом переход на следующую строку должен происходить по заполнению предыдущей, то есть не следует нажимать клавишу Enter для возврата каретки.

```
Buritto-SoPj_Pro-1627.mp3|C4U_-Cu-NikeCool-1689.mp3|DRLoops1-DRLoops-1552.mp3|Dualtro-Bionics-1855.mp3|Outtake_-Mach_New-4417.mp3|Oddgroov-OakLeY-1350.mp3
```

Если вы используете свои собственные файлы MP3, то просто введите название каждого из них (включая расширение), разделяя их знаком '|'.

13. После указания названий всех файлов введите следующий текст (также без возврата каретки):

```
&done=true&
```

14. Сохраните созданный файл под названием list.txt и закройте его.

На данном этапе важно позаботиться об организации ваших файлов. Flash будет обращаться к list.txt и всем файлам из папки assets при выполнении, поэтому их нужно выстроить в соответствии с порядком их воспроизведения. Если вы пользуетесь нашими файлами, то вы должны заметить, что они уже имеются в папке assets.

15. Удостоверьтесь, что у вас есть папка assets, размещенная там же, где и файл FLA с проигрывателем. Если это не так, переместите все видео и звуковые файлы вместе с list.txt в эту папку.

Теперь попробуем связать все созданные объекты фильма воедино с помощью ActionScript.

16. Вернитесь к своему фильму Flash для проигрывателя. Откройте панель Действия (F9) и выделите первый кадр слоя actions. Затем введите строчку:

```
location = "assets/list.txt";
```

Как видите, данная переменная просто указывает путь к только что созданному нами текстовому файлу, содержащему названия MP3 файлов.

Теперь установим общий «просмотр» всех наших компонентов с помощью ActionScript. Ранее в данной главе мы уже обсуждали установку свойств globalStyleFormat. Здесь мы воспользуемся ею с помощью элемента registerSkinElement для придания одинакового

внешнего вида фону, кнопкам и другим составляющим компонентов, причем цвет будет установлен далее.

Следующие несколько шагов включают в себя описания программного кода. Конечно, можно подробно рассмотреть и ввести каждую из строчек вручную, но мы предлагаем поберечь руки и время, то есть просто скопировать все из прилагающегося файла *Chapter_11_code.txt* для шагов с 16 по 23.

17. За последней введенной строкой добавьте следующие:

```
function setStyleFormat () {
    //This function sets the skin colors of the ListBox and playback buttons
    //You can optionally set the font and font size below. Note: the ListBox
    //will adjust to fit an exact row based on the font and font size chosen
    //
    //globalStyleFormat.textFont = "hooge 05_54"; //www.miniml.com
    //globalStyleFormat.textSize = 8;
    //
    globalStyleFormat.textColor = 0x000000;
    globalStyleFormat.background = 0x88894B0;
    globalStyleFormat.backgroundDisabled = 0xCOCOCO;
    globalStyleFormat.selection = 0x01 0EA3;
    globalStyleFormat.selectionDisabled = 0xCOCOCO;
    globalStyleFormat.selectionUnfocused = 0x88894B0;
    //
    globalStyleFormat.arrow = 0xFFFFFFFF;
    globalStyleFormat.face = 0x7080A0;
    globalStyleFormat.shadow = 0x000000;
    globalStyleFormat.darkshadow = 0x373F59;
    globalStyleFormat.highlight = 0xFFFFFFFF;
    globalStyleFormat.highlight3D = 0xA8B0C8;
    globalStyleFormat.foregroundDisabled = 0x7080A0;
    globalStyleFormat.scrollTrack = 0x7080A0;
    //
    globalStyleFormat.applyChanges ();
}
```

18. Далее нужно подключить следующие функции:

```
function adjVol (v) {
    soundIns.setVolume (v);
}
function adjPan (p) {
```

```

    soundIns.setPan (p);
}
function resetPlayButton () {
    controls_mc.play mc.gotoAndStop (1);
}
function buttonsVisible () {
    controls_mc.stop_mc._visible = true;
    controls_mc.play mc._visible = true;
}

```

Данные функции выступают в роли посредника между пользователем, осуществляющим свой выбор, и соответствующими объектами. Среди них есть новый объект под названием `soundIns`, который просто является звуковым элементом типа `Object` для проигрывателя.

19. Теперь добавьте следующие функции:

```

function playS () {
    swich (kind) {
        case "sound" :
            if (soundLoaded) {
                stopSound = false;
                equalizer_mc.gotoAndStop (2);
                soundIns.setVolume (controls_mc.vol_mc.getVol ());
                soundIns.setPan (controls_mc.pan_mc.getPan ());
                if (pauseSound) {
                    startTime = pauseTime/1000;
                    soundIns.start (startTime);
                    pauseSound = false;
                } else {
                    soundIns.start ();
                }
                equalizerjnc.gotoAndStop (2);
                equalizer mc.graph_mc.startPlaying ();
            } else {
                trace ("no sound loaded");
            }
            break;
        case "video":
            if (pauseVideo) {
                this.videoInitialize(numVideo, pauseFrame);
                pauseVideo = false;
            } else {

```

```
        equalizer_mc.gotoAndStop (1);
        this.videoInitialize(numVideo);
    }
    break;
default :
    break;
}
this.onEnterFrame = run;
}
function stops() {
    resetPlayButton();
    controls_mc.playback_mc.bar_mc._width = 0;
    switch (kind) {
    case "sound":
        stopSound = true;
        pauseSound = false;
        equalizer_mc.graph_mc.stopPlaying();
        soundIns.stop();
        break;
    case "video":
        stopVideo = true;
        pauseVideo = false;
        this["videos_mc"]["video"+numVideo+"_mc"].gotoAndStop (1);
        break;
    default :
        break;
    }
    delete this.onEnterFrame;
}
function pauses() {
    switch (kind) {
    case "sound" :
        pauseSound = true;
        pauseTime = soundIns.position;
        equalizer_mc.graph_mc.stopPlaying();
        soundIns.stop();
        break;
    case "video" :
        pauseVideo = true;
        this["videos_mc"]["video"+numVideo+"_mc"].stop();
        pauseFrame = this["videos_mc"]["video"+numVideo+"_mc"]._currentframe;
        break;
    }
```



```

default :
    break;
}
}

```

Здесь функции `playS`, `stopS` и `pauseS` переключаются, когда пользователь нажимает одну из кнопок `play`, `pause` и `stop`. Затем они определяют, нужно ли обрабатывать музыку или фильм, и какие установки должны быть изменены.

Ранее в данном разделе для каждого компонента `ListBox` в фильме `sound-video player` полю `Change Handler` мы задали значение `onChange` на панели `Параметры`. На следующей странице мы с помощью `ActionScript` создадим тело функции `onChange`, то есть определим, что должно произойти после того, как пользователь щелкнул на одной из записей в списке `ListBox`.

20. Добавьте следующие строки:

```

function onChange(component) {
    if (component._name == "soundList") {
        kind = "sound";
        var num = component.getSelectedIndex();
        _root.loader_mc.initLoad("_level0", "SOUND", "loading"+soundArr[num]);
        this.soundInitialize("assets/"+soundArr[num]);
    } else if (component._name == "videoList"){
        stops();
        kind = "video";
        equalizer_mc.fade_mc.fadeIn();
        numVideo = component.getSelectedIndex()+1;
        numOfVideos = component.getLength();
        buttonsVisible();
    }
}
}

```

21. Теперь добавим две новые функции, `soundInitialize` и `videoInitialize`, которые вызываются обработчиком `onChange` каждый раз, когда пользователь выбирает новый файл и нажимает кнопку воспроизведения. Добавьте следующее:

```

function videoInitialize(n,m) {
    var x = numOfVideos+1;
    if (m == null) {
        m = 2;
    }
}

```

```

while (x-) {
    if (x == n) {
        this["videos_mc"]["video"+x+"_mc"].gotoAndPlay(m);
    } else {
        this["videos_mc"]["video"+x+"_mc"].gotoAndStop(1);
    }
}
}

function soundInitialize(path) {
    stops();
    videoInitialize(null)
    soundIns = new Sound(this);
    soundIns.onLoad = function(success) {
        if (success) {
            resetPlayButton();
            buttonsVisible();
            soundLoaded = true;
        } else {
            trace("Download of MP3 file failed!");
        }
    };
    soundIns.loadSound(path, false);
}
}

```

22. Далее мы введем еще одну новую функцию под названием `percent`, которая будет определять в процентном отношении долю уже воспроизведенной части видео или звукового объекта. То есть если пользователь нажмет кнопку паузы, Flash будет знать, на каком месте файла произошла остановка. Поэтому проигрывание будет продолжено с правильной позиции. Для этого введите следующие строки:

```

function percent() {
    switch (kind) {
        case "sound" :
            var num = soundIns.position;
            var denom = soundIns.duration;
            break;
        case "video" :
            var num = this["videos_mc"]["video"+numVideo+"_mc"].currentframe;
            var denom = this["videos_mc"]["video"+numVideo+"_mc"]._totalframes;
            break;
        default :
            break;
    }
}

```

```

    }
    return num/denom;
}
function indicatePlaying(p) {
    controls_mc.playback_mc.bar mc_width = p*100;
}

```

23. Следующая функция, `sortAssets`, разбивает массив, полученный из файла `list.txt`, и содержит полученные значения в переменной `soundArr`. Введите следующее:

```

function sortAssets() {
    soundArr = soundFiles.split("|");
    var l = soundArr.length;
    soundList.removeAll();
    for (var x = 0; x<l; x++) {
        soundList.addItem(soundArr[x]);
    }
}

function loadAssetsList(1) {
    var startTime = getTimer();
    var wait = 3000;
    loadVariables(1, "");
    this.onEnterFrame = function() {
        currTime = getTimer();
        var n = currTime-startTime;
        if (done == "true") {
            sortAssets();
            delete this.onEnterFrame;
        } else if (n>=wait) {
            trace("could not find asset list");
            delete this.onEnterFrame;
        }
    };
}

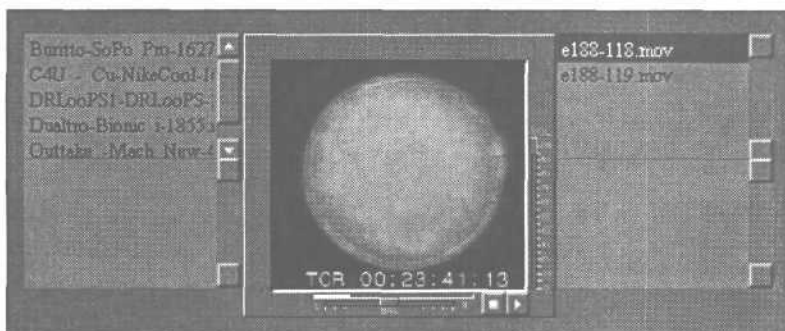
function run() {
    per = percent();
    (per<.99) ? indicatePlaying(per) : stops();
}

setStyleFormat();
loadAssetList(location);

```

Теперь можете выдохнуть — это весь код ActionScript необходимый для фильма sound-video player. Все, что осталось сделать, это закончить с графикой и протестировать созданный проигрыватель.

24. На слое background нарисуйте прямоугольник, охватывающий весь проигрыватель, и на этом работа с графическими элементами закончена.
25. Вернитесь на основную временную шкалу, переименуйте установленный по умолчанию слой в 'player' и перетащите на сцену фильм sound-video player.
26. Наконец, вставьте новый слой с именем 'actions' и примените действие stop к его первому кадру. Теперь проверьте созданный фильм нажатием клавиш Ctrl + Enter. При этом на экране должны отображаться все видео и звуковые компоненты:



27. Сохраните свой фильм и закройте его.

Заключение

Macromedia Flash в начале задумывался как инструмент для анимации с очень ограниченными возможностями написания сценариев (то есть программирования). За последние несколько лет он вобрал в себя язык объектно-ориентированного программирования и превратился в мощный разработчик web-приложений. Теперь пользователь может не только воспользоваться готовыми объектами, но и создать свои с последующим преобразованием их в компоненты Flash.

В данном разделе вы познакомились с базовыми концепциями и теорией объектно-ориентированного программирования. Вы узнали, как применить объектно-ориентированный подход во Flash-разработке, и создали полностью работоспособный видео и звуковой проигрыватель с использованием объектов. Неплохо, не правда ли?

Если вы решили, что ActionScript это ваша стихия, и горите желанием продолжить эксперименты с ним, то загляните на сайт ED, с которого можно скачать видео и звуковой синтезатор. Наш проигрыватель дает возможность его дальнейшей модификации и создания на его основе синтезатора, в котором пользователь мог бы одновременно воспроизводить и видео, и звук, а также задать порядок их воспроизведения.

Для всего этого потребуется еще поработать в ActionScript, но, смеем надеяться, мы помогли вам проникнуться удивительными возможностями применения ActionScript и приемов ООП при создании проектов. Мы будем рады, если смогли вдохновить вас на использование указанных приемов в вашей последующей работе с Flash. Ну что ж, вперед!

Послесловие

Поздравляем вас с окончанием курса Flash MX Video. Теперь вы в полной мере подготовлены к использованию самых передовых технологий подачи и демонстрации материалов при помощи Flash MX.

Однако не стоит успокаиваться на достигнутом. Эта книга познакомила вас с несколькими захватывающими сферами деятельности, и friends of ED (www.friendsofED.com) поможет вам в дальнейшем исследовании мира веб-дизайна.

После этой книги открываются два пути. Но это отнюдь не значит, что вы должны выбрать один из них, напротив — вы вольны пойти по обоим сразу!

Путь первый приводит в мир цифрового видео, знакомство с которым состоялось в главах 2 и 3. Съемка и обработка цифрового видео — это самостоятельный вид искусства, и с ним вас познакомит **DVision**, наш отдел, посвященный цифровому видео.

Зайдите на сайт www.DVision.info, здесь вы увидите издания, посвященные пакетам для цифрового видеомонтажа — Premiere, Final Cut Pro, After Effects, iMovie, QuickTime Pro, и, наряду с ними, книги, посвященные отдельным аспектам — такие, как *Transitions (Voices on the Craft of Digital Editing)* (*Переходы (Об искусстве цифрового монтажа)*).

Путь второй — дальнейшее исследование панорамы Flash MX, начатое нами в последней главе. Здесь вам укажет путь www.flashmxlibrary.com, серия изданий friends of ED, целиком посвященная великолепной новой версии Flash. Здесь вы найдете руководства для начинающих, такие, как *Foundation ActionScript (Основы ActionScript)*, описание более развитых приложений Flash, как *Macromedia Flash MX Studio*, и многое другое.

Не покидайте нас, закрыв последнюю страницу — нам хотелось бы узнать, что вы думаете об этой книге, как мы могли бы ее улучшить, какие книги вы хотели бы получить от нас в будущем. Для нас очень важна обратная связь с читателями, очень часто мы выпускаем книги именно по вашим просьбам. Напишите нам на feedback@friendsofed.com или просто зайдите на форум.

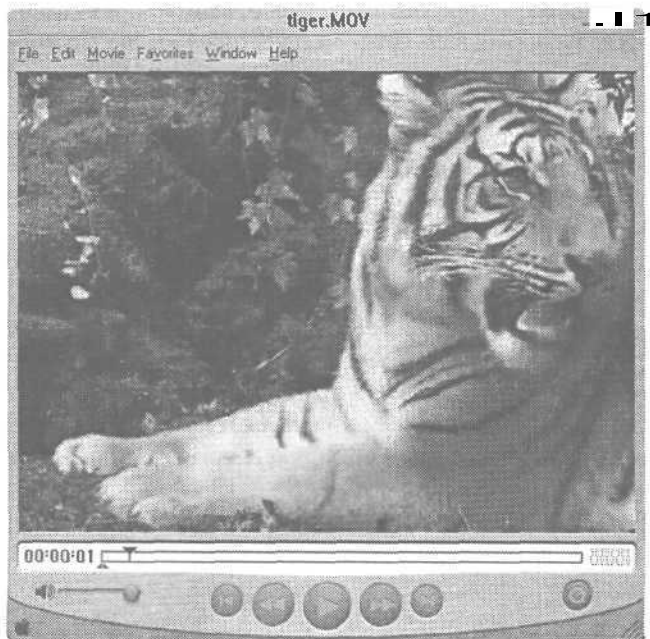
Приложение

Прочие видео надстройки

Для представления видео в сети существуют, помимо Macromedia Flash MX, и другие решения — надстройки (plug-ins) к браузерам. Давайте рассмотрим каждое из них, сравним их достоинства и недостатки, которые мы старались принять во внимание при разработке Flash MX.

QuickTime

QuickTime — проигрыватель видео и звука от Apple; устанавливается одновременно как надстройка и как самостоятельное приложение. Для показа сетевого материала надстройка QuickTime запускает приложение:



Распространение

ЕСЛИ вы являетесь пользователем Mac, то прекрасно осведомлены о QuickTime и его возможностях, поскольку он является важной составляющей частью операционных систем Macintosh (как OS X, так и Classic). Если вы — пользователь PC, и эта надстройка у вас не установлена, вы, возможно, видели логотип QuickTime на мультимедийных CD. Дело в том, что лучшее мультимедийное приложение для создания CD-ROM, Macromedia Director, полностью поддерживает фильмы QuickTime, позволяя встраивать в интерфейс видеоклипы и другие материалы QuickTime. Возможно также, что вам приходилось просматривать рекламные ролики кинофильмов на сайте www.QuickTime.com.

Итак, QuickTime предустановлен на всех Macintosh, а также может быть загружен либо напрямую установлен с мультимедийных CD для Windows и Mac.

Поддерживаемые форматы файлов

QuickTime поддерживает многие форматы видео файлов, включая MOV (стандарт QuickTime), MPG или MPEG, а также DV. Кроме того, поддерживаются интерактивные элементы QuickTime и SWF-файлы Flash 5 (только QuickTime версии 6).

Потоковая передача

QuickTime Streaming Server бесплатен для всех серверов Apple. Его эквивалент для PC, Darwin Streaming Server, также бесплатен для серверов под Linux, Windows и прочих. QuickTime Streaming Server использовался для потоковой прямой трансляции многих событий, в том числе фестиваля Glastonbury и печально известного турне Eminem с мотопилой!

Интерфейс

Интерфейс приложения довольно прост, но QuickTime обладает тем несомненным преимуществом, что его содержимое может быть непосредственно встроено в HTML-страницу. Вставить материал можно с контроллером или без, по выбору:



Очень полезная возможность, позволяющая встраивать видео в веб-страницы и презентации; благодаря этому удастся значительно улучшить восприятие материала пользователем. Мы уже знаем, что и Flash MX обладает такой же способностью.

Прочая информация

Возможно, вы не знаете, что при установке базового приложения QuickTime в вашу систему одновременно устанавливается QuickTime Pro. Для его активации необходимо получить от Apple регистрационный ключ, после чего вы сможете манипулировать различными форматами видео и аудио и редактировать их.

RealOne Player

RealOne Player — медиа-проигрыватель от Real Networks. Базовая инсталляция включает RealOne проигрыватель сам по себе и надстройку браузера.



Распространение

RealOne Player можно получить только путем скачивания из сети, базовая версия бесплатна. Заплатив, вы получите большую гибкость в работе и избавитесь от надоедливых всплывающих окон с напоминаниями.

Поддерживаемые форматы файлов

RealOne Player имеет несколько своих, эксклюзивных форматов — RAM, RV (видео) и RA (аудио), а также умеет воспроизводить стандартные веб-форматы, такие, как MOV или MPEG. Текущая версия способна воспроизводить материал Flash 4 SWF.

Потоковая передача

Сервер RealMedia можно использовать на PC серверах за плату. Цена рассчитывается в зависимости от количества одновременно выдаваемых потоков. Несколько телепрограмм BBC транслируется посредством потоков RealMedia.

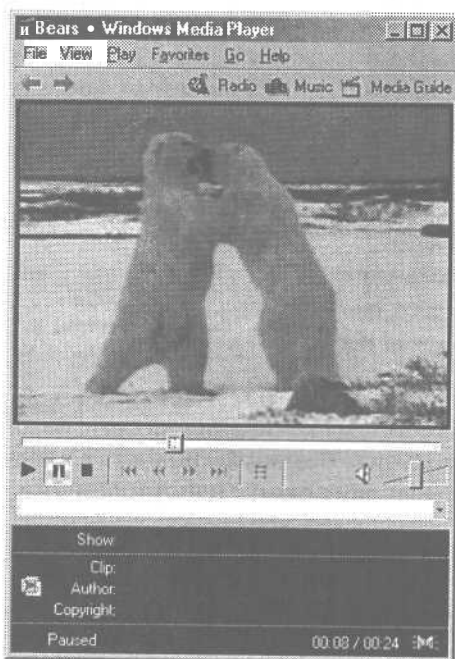
Интерфейс

Интерфейс — самый главный недостаток RealOne Player. Тяжеловесный, перегруженный ссылками и лишними функциями. Еще одна проблема состоит в том, что материал невозможно встроить в сайт или веб-

страницу. При запуске фильма с веб-сайта открывается приложение RealOne Player и фильм воспроизводится в отдельном окне, что сильно портит впечатление для зрителя.

Windows Media Player

Windows Media Player — эквивалент QuickTime для Windows.



Распространение

Пользователи Windows хорошо знакомы с этим приложением, которое используется для воспроизведения всего — от видео файлов до аудио CD. Как и QuickTime на Mac, Windows Media Player входит в комплект поставки Windows; урезанную версию (проигрывающую некоторые эксклюзивные форматы файлов) можно скачать для Mac.

Поддерживаемые форматы файлов

Windows Media Player имеет несколько эксклюзивных форматов — такие, как WMV, ASF, а также некоторые расширения для формата AVI. Поддерживается и большинство стандартных форматов, вроде MPEG и MOV.

Потоковая передача

Потоковая передача фильмов Windows Media Player бесплатна, но только для серверов под управлением Windows. Под Linux, Unix или Mac OS X она недоступна.

Интерфейс

Интерфейс приложения несложный, распределен по категориям, в которых легко ориентироваться. К сожалению, материал, вставленный в HTML-страницу, открывается в отдельном окне приложения, как и у RealOne Player.

К. Бесли, Х. Джиффорд, Б. Моннон, Т. Марк

Flash MX Video
Справочник профессионала

Перевод *А.В. Козлов, М.В. Аникина,*
Абдыманап уулу Айбек
Редактор *В.И. Грушецкий*
Верстка *М.В. Алексеева*

Директор издательства *В.Н. Говорухин*
Главный редактор *Н.В. Григорьева*
Технический директор *Е.В. Новиков*
Главный художник *О.В. Будко*

ПОЛЬЗОВАТЕЛЬ
Para(-)Type
IN LEGAL USE

Подписано в печать 15.09.2003. Формат 70x100 ¹/₁₆.
Гарнитура Балтика. Печать офсетная. 20 печ. л.
Тираж 3000 экз. Заказ № 1298

Издательство «СП ЭКОМ»
117342 Россия, Москва, ул. Бутлерова, д.17, оф. 105
Телефон для оптовых покупателей (095) 330-68-65

Отпечатано в ОАО «Можайский полиграфический комбинат»
143200, г. Можайск, ул. Мира, 93

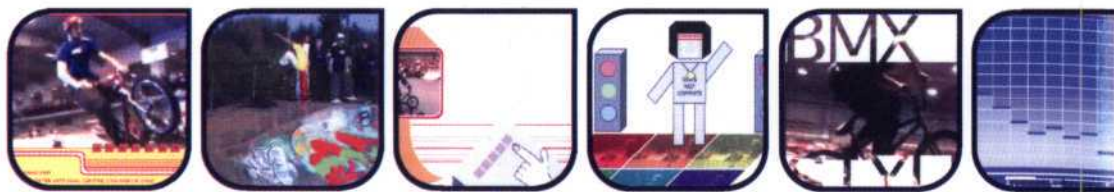
FLASH MX VIDEO

справочник профессионала

Macromedia Flash MX- это возможность импортировать цифровое видео и звук и управлять ими, как и любыми другими мультимедийными объектами. Откройте для себя волшебный мир видеодизайна!

- В первом разделе вы найдете исчерпывающие сведения о том, как импортировать видео и звук во Flash и о способах последующего манипулирования этими данными.
- Во втором разделе вы научитесь создавать настоящие видеофильмы - от подготовки к производству до выпуска конечного продукта, размещения его в сети или на CD-ROM.
- В последнем разделе мы погрузимся в сумрачный мир ActionScript и создадим полностью объектно-ориентированный компонент Flash для воспроизведения звука и видео.

Давайте делать фильмы!



Полная поддержка - на сайте www.friendsofed.com

friends of ED.gotoAndPlay

ISBN 5-9570-0011-6



9 785957 000112 >

ЭКОМ