**Дипломная работа**

**«Macromedia Flash MX»**

Новосибирск, 2008 г.

**Содержание**

Введение …………………………………………………………………………. 3

ЧАСТЬ I. ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ С FLASH

Глава 1. Основы пользовательского интерфейса Flash ……………………… 8

Глава 2. Типовые функции Flash ……………………………………………… 16

Глава 3. Настройка Flash ………………………………………………………. 20

ЧАСТЬ П. РАБОТА СО СТАТИЧНОЙ ГРАФИКОЙ.

Глава 4. Форматы статичной графики ……...………………………………… 24

Глава 5. Рисование ……………….…………………………………………….. 36

Глава 6. Работа с цветом ………………………………………………………. 47

Глава 7. Работа с текстом ……………………………………………………… 56

Глава 8. Импорт графики ……………………………………………………… 66

Глава 9. Работа с графическими фрагментами ………………………………. 69

Глава 10. Образцы и библиотеки. Проводник Flash …………………………. 73

Глава 11. Публикация и экспорт статичной графики ……………………..… 77

ЧАСТЬ III. РАБОТА С АНИМАЦИЕЙ

Глава 12. Форматы анимированной графики и видео ………………………. 80

Глава 13. Покадровая анимация ……………………………………………… 87

Глава 14. Трансформационная анимация ……………………………………. 88

Глава 15. Слои …………………………………………………………………. 89

Глава 16. Импорт анимации и видео …………………………………………. 92

Глава 17. Работа со звуком ……………………………………………………. 95

Глава 18. Создание интерактивных фильмов ………………….…………… 104

Глава 19. Создание приложений Flash ……………………………………… 106

Глава 20. Работа с внешними приложениями ……………………………… 111

Глава 21. Средства отладки сценариев ActionScript ………………………. 116

Заключение …………………………………………………………………… 120

Литература …………………………………………………….……………… 121

**Введение**

Данная работа посвящена последней на данный момент версии программного пакета Macromedia Flash - Flash MX, она же Flash 6.

Скажем сразу, что Flash - пакет компьютерной графики и формат сохранения ее в файле. Скажем больше: это пакет для создания и формат для сохранения двумерной анимированной компьютерной графики, предназначенной, в основном, для публикации в Интернете. Скажем проще: это средство создания мультиков, которые мы можем выложить в Сеть. Скажем чистую правду: именно Flash принес в Интернет высококачественную и компактную анимацию. Скажем еще кое-что: Flash породил целый вид искусства, известный как "Flash-анимация", и целую касту деятелей этого искусства, известную как "Flash-аниматоры".

На сегодняшний момент существует множество Web-сайтов, построенных с использованием технологии Flash. Есть также довольно много программ, использующих для тех или иных целей Flash-графику. Создано большое количество неплохих Flash-фильмов, которые мы можем увидеть на сайтах http://www.mp4.com, http://www.hypnotic.com, http://atomfflms.shockwave.coin и др. Существует Дмитрий Дибров, показывающий в телепередаче "Ночная смена" потрясающие по своей невразумительности Flash-ролики. И, наконец, существует целое сообщество «флэшеров».

Ниже мы подробнее рассмотрим все возможности, предлагаемые Macromedia Flash. И, поскольку Flash, в первую очередь, предназначен для создания интернет-графики, особое внимание мы уделим компьютерным сетям.

История компьютерной графики неотделима от истории персональных компьютеров. В самом деле, во времена больших ЭВМ компьютерная графика если и существовала, то носила чисто утилитарный характер. Например, нарисовать зелеными линиями на черном фоне простейший график, основанный на результатах каких-либо расчетов. Или вывести на экран только что спроектированную печатную плату. Как видите, тогдашние ЭВМ использовались только для дела. А, как поется в одной старой песне, "первым делом - самолеты"

Персональный компьютер (ПК) произвел настоящую революцию в мире вычислительной техники. Изначально подразумевалось, что это чудо может использоваться своим хозяином не только для дела, но и для потехи. (И делу время, и потехе час - так гласит пословица.) А для этого "персоналке" нужны хорошие возможности по выводу сложных графических изображений. Неудивительно, что даже на заре новой эпохи только самые дешевые ПК имели видеоадаптер, приспособленный исключительно для вывода текста (как говорят профессионааьные компьютерщики, алфавитно-цифровой).

Компьютерная графика эволюционировала вместе с компьютерным "железом" и программным обеспечением. Сначала это были корявые картинки, выполненные в шестнадцать цветов огромными пикселами. С совершенствованием графических подсистем и мониторов изображение на них стало выглядеть больше похожим на оригинал и меньше - на плохую мозаику. По мере совершенствования графических файловых форматов, алгоритмов сжатия и программ, обрабатывающих графику, качество изображения улучшалось, а размеры графического файла уменьшались. В результате компьютерные художники из немногочисленных сумасбродов превратились в настоящих профессионалов своего дела.

Сейчас все книги, журналы и газеты верстаются на компьютерах, в Интернете полно электронных репродукций картин и фотографий на любые вкусы (и на любое безвкусие), качество компьютерных игр приближается к качеству блокбастеров десятилетней давности, а сами блокбастеры помещаются на обычных компакт-дисках с надписью "DivX". Казалось бы, все замечательно. Так зачем нужен еще один графический формат - Macrome­dia Flash?

Для того чтобы ответить на этот вопрос, нам нужно обратиться к интернет-графике. Именно для Интернета и был создан Flash.

Что такое Интернет? Нет, не так... Что такое Интернет для большинства его пользователей? Электронная почта (отметаем сразу, ибо не наш профиль), вирусы (давить!), хакеры (давить!) и Всемирная Паутина, или, как говорят еще, World Wide Web, она же )^FrЖилипросто Web. Вот на ней мы остано­вимся подробнее.

Что такое Всемирная Паутина? Это Web-странички. А что такое Web-странички? Это текст, который можно читать, графика, которую можно смотреть, музыка, которую можно слушать, видео, которое также можно смотреть, и файлы, которые можно скачать. Если вы не новичок в Интернете, то сами знаете, что такое Паутина. Она живет, взаимодействует с вами, реагирует на вас и ваши действия. То есть, она интерактивна, иначе говоря, общительна.

Почтовые Web-серверы запрашивают у вас имя и пароль и выдают вам именно вашу почту. Новостные сайты выводят список самых последних событий "бегущей строкой", позволяя вам щелкнуть по нужному заголовку и прочесть больше. Файловые архивы регистрируют каждое скачивание каждого хранящегося на них файла. А сайты с онлайновыми играми? И "продвинутые" сайты с "продвинутыми" интерфейсными элементами, наподобие всплывающих меню или картинок, ползающих за курсором мыши? И концептуальные сайты с потрясающе красивыми заставками? Неужели вы не видели всего этого?!!

А ведь были времена, когда Интернет был полностью текстовым. Да-да, не удивляйтесь. Тим Бернерс-Ли, создавший в 1989 году язык HTML (HyperText Markup Language - язык гипертекстовой разметки), на котором и пишутся Web-страницы, не предусмотрел в нем поддержку графики. Потом, правда, под нажимом общественности консорциум WWWC (World Wide Web Consortium - консорциум всемирно протянутой паутины, ну и название!..), занимающийся развитием языка HTML, внес в него некоторые изменения. Так в Интернет пришла графика.

Но другой недостаток HTML не преодолен до сих пор. Это его порочная неинтерактивность. В самом деле, если вы хотите сделать на своей Web-странице всплывающее меню со ссылками на другие страницы, одним лишь HTML вам не обойтись. Вам нужно будет встраивать в HTML-код страницы программы, написанные на специальном языке JavaScript. Такие программы называются сценариями и служат для управления поведением той или иной части Web-страницы в ответ на то или иное действие пользователя. Как видите, врожденный порок инвалида - HTML - пытаются преодолеть с помощью костылей - сценариев. И не всегда это получается.

То, что дает нам Flash, можно выразить тремя словами: компактность, безопасность, интерактивность. Давайте рассмотрим подробно, что же скрывается за всем этим.

Файлы, хранящие изображения Flash, на самом деле очень компактны. Это достигается одной особенностью, которую мы во всех подробностях рас­смотрим позже. А пока проведем такую аналогию.

Возможно, вы писали какие-либо программы на одном из компилируемых языков программирования (Pascal, C++, Visual Basic и т. п.). При этом процесс написания программы выглядит следующим образом. Вы пишете исходный текст программы в виде текстового файла; при этом текст выглядит так, что с ним удобно работать (конечно, удобно тому, кто знаком с этим языком программирования). После этого вы запускаете особую программу - компилятор - и передаете ей получившийся текстовый файл. Компилятор преобразует набор команд языка в набор инструкций процессора - выполняет его компиляцию - и сохраняет его в исполняемом файле с рас­ширением exe (com, dll, ocx и др.). Если вы откроете исполняемый файл в текстовом редакторе, вы ничего не поймете - этот файл представляет собой мешанину шестнадцатеричных цифр, в которой нет ничего от исходного текстового файла. Однако процессор компьютера прекрасно понимает эту мешанину - для него этот язык "родной".

Так же поступает и Flash. При публикации Flash-изображения оно подвергается такому же преобразованию. В результате создается исключительно компактный файл формата Shockwave/Flash с расширением swf, аналог исполняемого файла обычной программы. Этот файл впоследствии может быть загружен и просмотрен с помощью специального проигрывателя Flash, который можно рассматривать как аналог процессора компьютера.

В последнее время появились технологии, позволяющие внедрять в Web-страницы небольшие программы, которые можно использовать, в том числе, для отображения различной информации и создания пользовательских интерфейсов. Существует две разновидности таких программ: атлеты Java и компоненты ActiveX. Апплеты (от английского applet - приложение) Java пишутся на межплатформенном языке программирования Java, разработанном фирмой Sun. Компоненты ActiveX были разработаны фирмой Microsoft и пишутся на любом языке программирования, их создание. К достоинству этих технологий можно отнести возможность реализации логики любой сложности, к недостатку - трудность создания.

И, наконец, самая "горячая" новинка современных интернет-технологий. Это язык описания документов и, одновременно, программирования Curl, разработанный фирмой Curl Corporation (http://www.curl.cora). Curl позиционируется как "могильщик" языков HTML, JavaScript, апплетов Java, компонентов ActiveX и практически всех графических форматов, используемых в Интернете, в том числе и Flash. Достоинством этого языка выступает простота написания документов и программирования любых сценариев, а недостатком - пока что очень малая распространенность.

Как видите, все вышеперечисленные технологии имеют серьезные недостат­ки, препятствующие их распространению. Только Flash сочетает такие особенности, как широкая распространенность, простота создания графики и реализация программной логики пользовательского интерфейса, высокое качество графики, богатые возможности по ее обработке и компактность получаемого файла. Неудивительно, что 95% пользователей Интернета имеют на своих компьютерах установленный модуль расширения Web-обозревателя, позволяющий просматривать графику Flash.

**ЧАСТЬ I. ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ С FLASH**

**Глава 1. Основы пользовательского интерфейса Flash.**

Рассмотрим базовые принципы работы с пакетом Macromedia Flash.

Итак, нам нужно создать изображение. Статичное или анимированное (то есть, фильм), простое или интерактивное (фактически, программу) - неважно. Что делать?

Прежде всего, четко представлять, что мы хотим сделать. Это означает, что нам нужно предварительно спланировать свое пока еще не существующее изображение. Лучше всего нарисовать его на бумаге, хотя это необязательно. Важно просто хорошо представлять себе окончательный результат. Это нужно хотя бы для того, чтобы в процессе работы не забыть, что мы хотим сделать. Ведь если мы что-то забудем, переделывать уже сделанное будет гораздо труднее, чем делать это сразу, правильно.

Второй этап - собственно рисование. Если в наши планы входит создание анимации, то сначала подготовим неподвижную часть графики. Потом займемся созданием самой анимации. Повторяю, здесь мы создаем саму графику, не затрагивая интерактивные возможности, если они есть. Нарисуем все, что нужно, даже те элементы, которые необходимы только для реализации интерактивных возможностей. Это нужно, чтобы проверить саму композицию нашего изображения и при необходимости вовремя исправить ее.

Интерактивными возможностями займемся на третьем этапе. Именно здесь мы создадим все элементы, которые будут отвечать только за интерактивность. И. конечисх напишем сценарии на языке ActionScript, которые и будут реализовывать эту интерактивность. Здесь же мы выполним отладку и исправим ошибки в этих сценариях.

Самый последний, завершающий этап - это публикация готового изображения в один из форматов, пригодных для распространения. Что подразумевается под публикацией, стоит рассмотреть более подробно.

Когда вы работаете в среде Flash, вам нужно сохранять промежуточные и окончательные результаты работы. Для этого Flash, как и все программы, предоставляет возможность создания файлов своего собственного формата, называемого документом Flash. Это файл с расширением - довольно больших размеров, в котором хранится вся графика и все сценарии ActionScript, привязанные к этой графике. Кроме того, в данном файле хранится вся информация, необходимая для того, чтобы вы могли в любой момент исправить как графику, так и сценарии. Подобная информация зачастую весьма объемиста, поэтому файл документа Flash имеет такие большие размеры - до сотен килобайт.

Вот поэтому Flash поддерживает два графических формата. Один из них служит для сохранения результатов работы в самой среде Flash - это фор­мат документов Flash, записываемых в файлы с расширением fla. Второй формат служит только для распространения графики и включает лишь ту информацию, которая нужна для отображения этой графики и придания ей интерактивности. Эта информация сильно оптимизирована, чтобы уменьшить размер файла и ускорить его обработку. Такие файлы называются распространяемыми файлами Shockwave/Flash. He перепутайте эти два формата данных - они совершенно разные и служат разным целям, хотя и тот, и другой предназначены для сохранения графики.

Формально последним этапом создания Flash-изображения является его распространение. Но мы не будет рассматривать, как это осуществляется. В конце концов, это проблемы уже не Flash. Здесь мы опишем только, как и с помощью какого программного обеспечения можно просмотреть готовое изображение.

Скажем сразу, что изображение Flash можно экспортировать в другом формате, например, Apple QuickTime, AVI или тех же GIF или JPEG. Правда, кое-какая информация при этом потеряется, в частности, сценарии ActionScript. Поэтому в другие, альтернативные Shockwave/Flash, форматы стоит экспортировать только ту графику, которая не имеет никаких интер­активных возможностей.

В отличие от изображений GIF и JPEG, поддерживаемых всеми существующими на данный момент программами Web-обозревателей, изображение Shockwave/Flash для просмотра требует специальной программы - проигрыватвля Shockwave/Flash. Этот проигрыватель может быть встроен в Web-обозреватель в качестве модуля расширения или существовать в виде программы, запускаемой отдельно. В этом формат Shockwave/Flash схож с форматами Apple QuickTime и AVI, для которых также нужны отдельные проигрыватели, в первом случае - одноименный проигрыватель фирмы Apple, во втором - Универсальный проигрыватель фирмы Microsoft, поставляемый в составе Windows.

Но с проблемами распространения своего творчества мы столкнемся еще не скоро. Пока что мы не знакомы с самой средой Flash. Познакомимся же с ней поближе.

Главное (или основное) окно программы Macromedia Flash показано на рис. 1.1. Рассмотрим его подробнее.

Скажем сразу, что Flash - приложение с многодокументным интерфейсом или просто многодокументное приложение. Это значит, что вы можете открыть в одном и том же рабочем окне программы сразу несколько документов. В этом случае окна, содержащие открытые документы, открываются внутри большого окна самой программы. К многодокументным приложениям также относятся Microsoft Word и Adobe Photoshop. Этим они отличаются от приложений с однодокументным интерфейсом (однодокументных приложений), в которых можно открыть только один документ, а чтобы открыть второй, нужно запускать вторую копию приложения. Примерами однодокументных приложений являются, в частности, текстовый редактор Microsoft WordPad и графический редактор Microsoft Paint, поставляемые в составе Windows.

Окно документа Flash занимает большую часть окна программы. Его заголовок совпадает с именем открытого в нем файла. Вы можете перемещать, свертывать и развертывать это окно и изменять его размеры, в общем, проделывать с ним те же манипуляции, что и с любым другим окном Windows. Единственное исключение: вы не можете "вытащить" это окно за пределы окна программы (так называемого родительского окна).

При первом запуске Flash MX выводит на экран еще одно небольшое окно, показанное на рис. 1.2. Это так называемое приглашение, содержащее текст, предлагающий пользователю прочитать некоторые справочные данные или запустить интерактивные презентации, объясняющие, как работать во Flash. Вы можете просмотреть их или сразу же закрыть это окно. При последующих запусках окно-приглашение появляться больше не будет.

Левее окна документа находится главный инструментарий Flash или просто инструментарий. (Его также называют панелью инструментов, но мы не будем употреблять этот термин, чтобы избежать путаницы.) Эта небольшая серая панель, вытянутая по вертикали, содержит в себе набор кнопок. Нажимая кнопки, вы можете выбирать те или иные инструменты, предлагаемые Flash для рисования или правки графики.

Если вы "захватите" инструментарий мышью за особую "ручку", отображаемую в виде набора точек на темно-сером фоне в ее верхней части (рис. 1.3), то он "отклеится" от края родительского окна и превратится в независимое окно (рис. 1.4). Таким образом вы можете увеличить площадь окна документа Flash. Можно перемещать это окно по экрану так же, как любое другое окно. Есть даже возможность "вытащить" его за пределы окна программы, что недоступно для окон документов. Однако менять размеры окна, содержащего инструментарий, вы не можете - они всегда постоянны.

Если же вы не хотите, чтобы инструментарий маячил постоянно перед глазами, вновь "приклейте" его к краю окна. Для этого "поднесите" его мышью к левому или правому краю родительского окна и оставьте там. Отследить момент "приклеивания" и "отклеивания" очень просто: если при перетаскивании инструментарий меняет толстый контур на тонкий, то при отпускании кнопки мыши он будет "приклеен". И наоборот, если тонкий контур меняется на толстый, инструментарий будет "отклеен". Чтобы ин­струментарий ни в коем случае не "приклеивался" к краю окна, при его перетаскивании удерживайте нажатой клавишу <Ctrl>.

Если вы хорошенько присмотритесь к инструментарию, то увидите, что он разделен на четыре области. Перечислим их сверху вниз.

1. Область основных инструментов (заголовок Tools). Здесь находятся кнопки, предоставляющие доступ ко всем инструментам, что предусмотрены во Flash для рисования и правки уже нарисованного.

1. Область вспомогательных инструментов (заголовок View). Здесь находятся всего две кнопки, которые мы рассмотрим в этой главе.
2. Область задания цвета (заголовок Colors). Здесь находятся элементы управления, позволяющие вам задавать цвет.
3. Область модификаторов (заголовок Options). Здесь находятся кнопки, предоставляющие доступ к модификаторам - дополнительным режимам, предусмотренным в том или ином выбранном в данный момент инструменте.

Не всегда в данный момент времени в инструментарии присутствуют все четыре области. Область модификаторов в некоторые моменты может быть пуста.

Панель может быть перемещена в любое место экрана, даже за пределы главного окна программы. Перетаскивать панель можно как за ее "ручку", так и за заголовок ее окна. Кроме того, вы можете изменять размеры окон панелей, что недоступно для инструментария.

Большинство панелей имеют так называемое дополнительное меню. Оно открывается при щелчке мышью по небольшой кнопке, расположенной в правом верхнем углу панели и имеющей изображение списка из трех позиций и небольшой стрелки, направленной вниз (рис. 1.9). В дополнительном меню находятся пункты, выполняющие редко используемые команды.

Пункт Tools служит для вывода на экран или скрытия главного инструментария. Если слева от имени этого пункта стоит галочка, это значит, что инструментарий выведен на экран (или, как еще говорят, что соответствующий пункт меню "включен"). Чтобы убрать его, снова выберите этот пункт; инструментарий исчезнет вместе с галочкой. Если впоследствии вы еще раз выберете этот пункт, инструментарий и галочка снова появятся. Такие пункты меню, меняющие свое состояние на противоположное при выборе, называют выключателями. Вместо выбора этого пункта вы можете нажать "горячую" комбинацию клавиш <Ctrl>+<F2>.

Большую часть меню Window занимает набор аналогичных пунктов-выключателей, служащих для вывода на экран или скрытия различных панелей. Далее в книге мы рассмотрим эти пункты вместе с рассмотрением той или иной панели.

Если вам для какой-то цели нужно закрыть все панели и инструментарии, например, чтобы просмотреть без помехи все изображение, выберите пункт Close All Panels. Все панели будут тут же закрыты. Но имейте в виду, что открывать вам их придется вручную.

Если же вам нужно не закрыть безвозвратно, а просто скрыть на время все панели и инструментарии, выберите пункт Hide Panels в меню View. Этот пункт работает как выключатель, т. е. при первом выборе он скрывает все панели, а при втором - снова выводит их на экран. Вы также можете нажать клавиши <ТаЬ> или <F4> - это проще и быстрее, чем лезть в меню.

Есть еще один весьма удобный способ организовать свое рабочее место: пользовательские раскладки панелей. В двух словах это можно объяснить следующим образом. Вы открываете нужные вам в данный момент панели, располагаете на экране так, как вам удобно, и сохраняете их расположение в настройках Flash. После этого вы можете закрыть их совсем или переместить на другие места. Но чтобы вернуться к прежнему состоянию, вам достаточно будет выбрать сохраненную ранее раскладку, и все панели выстроятся на экране так, как они были выстроены при сохранении выбранной раскладки. Таких сохраненных раскладок может быть сколько угодно.

Чтобы сохранить раскладку панелей, выберите пункт Save Panel Layout. На экране появится диалоговое окно Save Panel Layout (рис. 1.12). Введите в поле ввода Name имя сохраняемой раскладки и нажмите кнопку ОК. Если вы передумали сохранять раскладку, нажмите кнопку Cancel.

**Окно документа**

Окно документа Flash показано на рис. 1.13. Рассмотрим его подробнее.

Как видите, окно документа Flash разделено на две неравные части. Разделены они тонкой серой линией, которую вы можете перетаскивать мышью, меняя размеры этих частей. Ниже этой серой линии, практически сливаясь с ней, находится также небольшая панель с кнопками, которую невозможно скрыть.

Эта панель принадлежит окну документа, а не главному окну программы. Теперь обратим внимание на нижнюю, большую часть окна. В ней на сером окне находится белый прямоугольник, ограниченный тонкими черными иниями.

Небольшое изображение, нарисованное нами для примера, находится как раз внутри этого прямоугольника. Этот прямоугольник схематически представляет наш рисунок, и размеры его совпадают с размерами этого изображения, заданного при его создании.

То, что не входит в пределы этого прямоугольника, не попадает в наше изображение и отбрасывается Flash при сохранении в формате Shockwave/Flash. Назовем этот прямоугольник рабочим листом или просто листом, а всю серую область, где рисуется графика, - рабочей областью.

Как вы уже поняли, изображение рисуется в рабочей области. При этом все,что не попадает на рабочий лист (то есть, находится на окружающем его сером поле), не войдет в окончательное изображение, сохраненное в файле Shockwave/Flash. Однако в файле документа Flash сохраняется все: и попадающее на рабочий лист, и находящееся на сером поле. Этим можно пользоваться, размещая некоторые части изображения на сером поле и перетаскивая их на лист, когда в них появится нужда.

**Глава 2. Типовые функции Flash.**

Разобравшись с пользовательским интерфейсом Flash и предоставляемыми им возможностями, перейдем к работе с файлами. В этой главе мы расскажем о файловых операциях Flash, точнее, об их особенностях по сравнению с другими Windows-приложениями, создающими файлы документов.Когда вы набираете текст в любимом Microsoft Word, вы сохраняете его на жестком диске в виде файла. Файл - это массив информации, записанный на дисковом устройстве и имеющий уникальное имя, по которому его можно однозначно распознать. Кроме имени, файл имеет также набор атрибутов: признак "го ,ько для чтения", даты создания и последнего изменения, комментарий и т. п. Но, как правило, опознается файл по имени. Документы Microsoft Word хранятся в файлах. Документы Macromedia Flash также хранятся в файлах, как и готовые к распространению изображения формата Shockwave/Flash. Сама программа Flash хранится в виде огромного многомегабайтного файла. А уж сколько файлов занимает интерактивное руководство по Flash - страшно представить!Любое приложение, создающее какие-либо документы, должно сохранять их в файлах. А для этого оно должно поддерживать так называемые файловые операции: создание, открытие, закрытие, сохранение, сохранение под другим именем (пересохранение). Таким образом, файловые операции относятся к типовым операциям, которые должны поддерживаться всеми подобными программами. Также к типовым операциям относятся печать документа и предварительный просмотр его перед печатью.В приложениях, написанных для Windows, выполнение типовых операций стандартизировано. Windows предоставляет стандартные диалоговые окна открытия, сохранения файлов, печати и настройки принтера. (Исключение составляют разве только ученические и совсем уж хитроумные программы.) Обычные диалоговые окна вам, конечно, знакомы, поэтому мы не будем их описывать. Сосредоточимся только на особенностях, присущих именно Flash.

Прежде, чем начать работать с документом, его нужно создать. Давайте же выясним, как создать новый документ (точнее, новый файл документа) в среде Flash.

Создать сам документ очень просто. Для этого выберите в меню File пункт New или нажмите комбинацию клавиш <Ctrl>+<N>. Внутри окна программы Flash откроется новое окно документа с пустым рабочим листом и пустой временной линией. На этом этапе никаких новых диалоговых окон, запрашивающих дополнительную информацию, на экране не появляется.

Собственно, когда вы запускаете программу Flash щелчком по ее ярлыку на Рабочем столе, в меню Start (Пуск) или на самом исполняемом файле, новый документ создается автоматически сразу после ее запуска. Прибегать к пункту New меню File вам нужно только тогда, когда вы хотите создать еще один новый документ в процессе работы с программой.

Сохранение документа в файле выполняется после выбора пункта Save меню File или нажатия комбинации клавиш <Ctrl>+<S>. Если сохранение выполняется первый раз, т. е. документ еще не был сохранен в файле, то на экране появится стандартное диалоговое окно сохранения файла Windows, где вы должны будете задать имя файла. Впоследствии Flash будет просто сохранять документ, не спрашивая имени файла.

Flash MX также предоставляет вам возможность сохранить документ в формате Flash 5 (предыдущей версии Flash). Для этого просто выберите в списке типов файла стандартного диалогового окна сохранения пункт Flash 5 Document.

Если вам нужно пересохранить документ в другом файле под другим именем, выберите пункт Save As в меню File или нажмите комбинацию клавиш <Ctrl>+<Shift>+<S>. На экране появится стандартное диалоговое окно сохранения файла Windows, где вы должны будете задать имя нового файла.

Если вы после множества изменений хотите загрузить последнюю сохрг ненную версию документа, выберите пункт Revert в меню File. Учтите только, что этот пункт недоступен, если вы ни разу не сохранили текущий документ или ни разу не изменяли его после загрузки.

Чтобы закрыть открытый файл, активизируйте окно документа, в котором он открыт, и выберите пункт Close меню File или нажмите комбинацию клавиш <Ctrl>+<W>. Хотя проще всего будет закрыть это окно (имеется в виду окно документа, а не окно программы).

Кроме того, Flash предоставляет возможность отправки открытого в его среде документа по электронной почте. Для этого выберите пункт Send в меню File. На экране появится окно программы почтового клиента, установленного в системе в качестве клиента по умолчанию. В этом окне будет сформировано готовое письмо с вложенным в него файлом, содержащим отправляемый документ. Вам останется только вписать адрес, тему и, возможно, текст письма и запустить отправку почты.

**Печать**

Печать документов, созданных в среде Flash, несколько отличается от печати документов в других программах. Сейчас мы рассмотрим эти отличия. И заодно перечислим все операции, связанные с печатью документов.

Собственно печать документа Flash осуществляется обычным для Windows-приложения способом. Вы выбираете пункт Print в меню File или нажимаете комбинацию клавиш <Ctrl>+<P>. После этого на экране появляется стандартное диалоговое окно печати Windows, в котором вы можете выбрать принтер, задать количество копий и выбрать страницы, которые должны быть напечатаны. Flash для печати использует стандартное окно без всяких изменений.

Как и многие Windows-приложения, работающие с документами, Rash предоставляет возможность просмотра документа в том виде, в котором он будет распечатан. Для этого выберите пункт Print Preview в меню File. На экране появится окно предварительного просмотра, где будет показан документ.

В верхней части этого окна находится набор кнопок, с помощью которых над показанным в нем документом выполняются различные манипуляции. Всего этих кнопок семь:

Print - ззапуск печати документа;

Next Page - переход на следующую страницу мента, если она есть;

Prev Page - переход на предыдущую страницу многостраничного документа, если она есть;

One Page или Two Page - одна и та же кнопка, переключающая режимы отображения одной или двух страниц документа одновременно;

Zoom In - увеличение изображения;

Zoom Out - уменьшение изображения;

О Close - закрытие окна предварительного просмотра.

Опять же, как и многие программы, предусматривающие печать своих документов, Flash предоставляет возможность задать параметры печати. Эти параметры включают задание размера

бумаги, отступов, масштаба и центрирования изображения. Чтобы получить доступ к этим настройкам, выберите пункт Page Setup в меню File. На экране появится диалоговое окно Page Setup (рис. 2.6).

**Интерактивная справка Flash**

Для вызова интерактивной справки вам нужно просто нажать клавишу <F1> или выбрать пункт Using Flash в меню Help. После этого на экране появится окно встроенной справочной системы, показанное на рис. 2.8. Как видите, справочная система Flash использует для работы Web-обозреватель, установленный в системе по умолчанию.

В левой части окна справки расположен древовидный список статей. Вы можете щелкнуть мышью по названию статьи, чтобы просмотреть ее содержание, развернуть или свернуть ветвь "дерева" статей. Содержание выбранной статьи отображается справа.Вы можете искать определения различных терминов, встретившихся в руководстве. Выполняется такой поиск ледующим образом. Над списком тем находится небольшое.меню из трех пунктов. Пункт Contents выводит собственно список тем. Пункт Index отображает набор букв алфавита (латинского, естественно). Выберите нужную букву, и на экране появится список терминов, начинающихся на эту букву. Вам останется только щелкнуть нужный термин, чтобы просмотреть его разъяснение.

**Глава 3. Настройка Flash.**

Разработчики практически каждого программного продукта стремятся сделать свое детище популярным, распространить его на максимальное количество пользователей. Любой программист, даже если он написал всего одну примитивнейшую бесплатную утилиту, в своих самых сладких снах видит себя королем программного обеспечения, а свое детище - установленным на каждом компьютере мира. И это нормально. Тщеславие свойственно человеческой природе так же, как и любопытство.

И, наряду с любопытством, оно правит этим миром. Более того, оно движет его вперед, объединяет людей, делает их единым целым - человечеством. В самом деле, представьте на минутку, что было бы, если каждый станет творить только для себя любимого, забыв об остальном человечестве? Мир бы раскололся на множество индивидуальных мирков; в чем-то эти мирки, несомненно, будут богаче большого мира, но в остальном - несравнимо беднее. "Настоящий творец творит только для себя", - сказал какой-то высоколобый мизантроп, но заприте этого умника в уютный карцер, изолируйте его от остального мира и посмотрите - много ли он натворит.

В этой главе мы поговорим о тщеславии, человеколюбии и пользовательских настройках Flash. Как это связано между собой, мы сейчас объясним.

Итак, если программист не относится к числу "настоящих творцов", он стремится сделать свой программный продукт доступным максимальному количеству пользователей. То есть, как можно более популярным. Но, по­скольку люди и их привычки очень разные, возникает проблема: как угодить всем, как сделать, чтобы всем было удобно? Существует два пути решения этой проблемы.

Путь первый: переписывание программы под каждого конкретного пользователя (как правило, за отдельную плату). Такого подхода придерживаются разработчики бухгалтерских, складских, учетных, расчетных и прочих "деловых" программ. Хорошо известно, что бухгалтерия и складской учет в разных странах существенно отличаются, поэтому каждую такую программу приходится иной раз писать заново для каждого конкретного заказчика. А поскольку стандартизацией учета и контроля здесь и не пахнет, разработчики деловых программ всегда будут иметь свой кусок хлеба (и свою головную боль).

Путь второй: создание гибких, настраиваемых программ. Такой подход выгоден для универсальных программ, которыми пользуются все: текстовых редакторов, обработчиков электронных таблиц, систем управления базами данных (СУБД), графических программ, системных утилит, проигрывателей видеофильмов и т. п. Разработчик оснащает свою программу возможностями ее настройки пользователем, эти возможности могут быть более или менее широкими, в зависимости от программы. В самом деле, не будет же он переписывать весь текстовый редактор полностью, чтобы изменить набор кнопок на инструментальной панели, да еще требовать за ую деньги! (Мягко говоря, его не поймут.)

Macromedia Flash относится как раз ко второй категории программ, имею­щих возможности настройки пользователем. Эти возможности довольно широки, мы рассмотрим их в отдельной, третьей главе книги, в ее первом разделе. И рассмотрим очень подробно.

Вы уже знаете, что многим пунктам меню Flash присвоены клавишные комбинации. Таким образом, вы можете вызвать команду, просто нажав эту комбинацию, а не обращаясь к меню - так гораздо быстрее. Но Flash позволяет вам также перенастроить эти комбинации клавиш, если стандартные вас почему-то не удовлетворяют. Настройке клавиатурных комбинаций посвящен второй раздел этой главы.

**Настройка программы**

Все пользовательские настройки Flash осуществляются в диалоговом окне настроек Preferences. Чтобы вызвать его на экран, выберите пункт Preferences в меню Edit (или нажмите комбинацию клавиш <Ctrl>+<U>). Само это диалоговое окно, точнее, вкладка General, где настраиваются основные параметры программы.

В поле ввода Undo Levels задается количество операций пользователя, сведения о которых Flash хранит в памяти. Это нужно для выполнения операции так называемого отката, т.е. отмены результата последней операции пользователя. (Подробнее об откате см. главу 5.) В большинстве случаев лучше оставить значение по умолчанию (100). Если вы хотите сэкономить оперативную память компьютера для других нужд, можете уменьшить это значение вплоть до нуля. Если же, наоборот, вы считаете, что не уверены в себе, то можете увеличить его до 200.

Включение флажка Disable PostScript позволит вам отключить PostScript-вывод при печати на принтере, поддерживающем PostScript. Это может замедлить печать, так что включайте этот флажок только при наличии проблем с выводом изображения на принтер.

Если флажок Shift Select включен (а он включен по умолчанию), то для выделения нескольких графических фрагментов на листе вам придется удерживать нажатой клавишу <Shift>. Если же он выключен, вам не нужно удерживать клавишу <Shift> - Flash будет выделять фрагменты при последовательных щелчках мышью. Включать или не включать этот флажок - на наш взгляд, дело вкуса.

Флажок Show Tooltips включает или отключает вывод всплывающих подсказок при наведении курсора мыши на кнопки и другие элементы управления панелей и инструментариев. Если этот флажок отключен, то в строку статуса окна программы также не будет выводиться справочная информация по пунктам меню.

Включение флажка Disable Panel Docking отменяет возможность "приклеивания" панелей к краям окна программы. Включение флажка Disable Timeline Docking отменяет ту же самую возможность для временньй шкалы. О временньй шкале и анимации вообще см. часть 3.

Если включен флажок Span Based Selection, то при щелчке на растянутом кадре будет выделен весь кадр. Если этот флажок выключен, то при щелчке будет выделен только тот промежуточный кадр, по которому щелкнули мышью. Фактически, флажок Span Based Selection заставляет Flash MX вести себя как предыдущая версия этого пакета Flash 5. Изначально он отключен, включите его, если вы переходите на Flash MX с предыдущей версии.

Если включен флажок Named Anchor on Scene, Flash делает первый кадр каждой сцены именованным "якорем". О сценах и "якорях" см. главу 13.

С помощью переключателей Highlight Color можно задать цвет, которым будут выделяться экземпляры образцов и группы графических фрагментов. (О группах см. главу 5, а об экземплярах - главу 10.) Если выбран переклю­чатель Use this color, вы сможете задать цвет в расположенном правее этого переключателя селекторе цвета. Если же выбран переключатель Use layer color, то Flash будет использовать текущий цвет линии.

Раскрывающийся список Font Mapping Default позволяет установить шрифт, который будет подставляться вместо отсутствующих в системе шрифтов. Выберите любой шрифт, который вам нравится.

**ЧАСТЬ П. РАБОТА СО СТАТИЧНОЙ ГРАФИКОЙ..**

**Глава 4. Форматы статичной графики**

Итак, с пользовательским интерфейсом Macromedia Flash мы разобрались. Как выполняются типичные для всех Windows-приложений задачи, выяснили. Теперь самое время приступить к работе над графикой.

Да, мы так и поступим. Сначала рассмотрим создание статичной графики, потом перейдем к анимации. Ну, а уж после того, как мы научимся делать анимированные картинки, можно рассмотреть и программирование во Flash: интерактивные фильмы, пользовательские интерфейсы интернет-программ и т. п. Таким образом, к концу этой книги вы узнаете все, что может Flash, и сможете использовать это "все" в своих целях.

Но не торопитесь. Сначала немного поговорим о самой компьютерной графике.

Вы скажете: мы уже говорили о ней во введении. Мы узнали, какое место в компьютерном мире вообще и в Интернете в частности она занимает сейчас, даже познакомились с краткой ее историей. Да, кое-что вы уже знаете, но этого для настоящей работы недостаточно. Современный компьютерный художник должен знать значительно больше.

Чтобы полноценно работать с компьютерной графикой, нужно знать, в каких форматах она хранится, что они собой представляют, и какой формат в каком случае используется. Формат хранения графики - это способ записи графической информации в файле на диске. Чтобы корректно отобразить тот или иной графический файл, программное обеспечение должно поддерживать формат, в котором он записан.

Форматов хранения графической информацией трудолюбивое человечество наплодило великое множество. Основная их масса была создана достаточно давно, вместе с программами для обработки графики, т. е. они являлись фирменным, "родным" форматом для той или иной программы. Очень немногие графические форматы были задуманы специально для обмена графическими файлами между разными программами (межпрограммные или программно-независимые форматы). К первым форматам можно отнести известный формат BMP, созданный специально для хранения графики в Windows-программах, а ко вторым - популярнейший формат GIF. В дальнейшем многие форматы графики исчезли (зачастую вместе с программами, для которых они были созданы), и лишь малое число их дожило до нашего времени.

В живом мире действует жестокий закон эволюции. Это же справедливо и для мира компьютеров. Любое устройство, программа, любая технология нежизнеспособны, если они не найдут для себя "экологическую нишу", т. е. будут использоваться людьми. Так же как и любой графический формат. Те из них, что остались "живы", более или менее активно используются и покрывают практически все нужды современных компьютерных художников. Их-то и поддерживает большая часть современного графического программного обеспечения.

**Растровая графика**

Если взять обычную газетную иллюстрацию и хорошенько ее рассмотреть, желательно под лупой, то можно увидеть, что она на самом деле состоит из множества точек. Эти точки могут быть как жирно-черными (в тех местах, где на иллюстрации виден глубокий черный фон), так и более или менее серыми (где присутствуют полутона) или вообще быть почти незаметными (на белых местах). Если рассматривать такую иллюстрацию на некотором расстоянии, а не вплотную, как сделали мы, отдельные точки сливаются в единое изображение. Можно сказать, что обычная газетная иллюстрация - классический пример растровой графики.

Растровая графика впервые была создана полиграфистами для того, чтобы с минимальным расходом краски печатать на бумаге иллюстрации. При этом на исходное изображение словно бы накладывается тонкая сетка, называемая растром, и изображение оказывается разбитым на множество точек (пикселов). После этого остается только вычислить интенсивность черного цвета в каждой точке растра или, для цветного изображения, цвет точки и, возможно, ее прозрачность. Этот процесс называется растеризацией. Полученные в результате растеризации данные либо сразу же переносятся на типографские пленки, либо записываются в файл для дальнейшей обработки.

**Векторная графика**

Предположим, что любое, даже очень сложное графическое изображение можно разбить на простейшие элементы: прямые, кривые, эллипсы, прямоугольники и т. п. Эти простейшие элементы, называемые графическими примитивами, можно описать определенными формулами. В результате получится набор параметров, используя которые, можно точно воссоздать исходный набор графических примитивов, а значит, и исходное изображение.

Такая графика, состоящая из отдельных примитивов, называется векторной графикой. Именно в таком виде сохраняются изображения Flash.

**Гибридная графика**

Собственно, гибридная графика - это разновидность векторной графики, содержащей внутри себя растровые изображения. Благодаря этому часто удается преодолеть главнейшие недостатки и растровой, и векторной графики: большой размер (у растровой графики) и невозможность точной передачи полутоновых изображений (векторной графикой).

Гибридное изображение разбито на фрагменты, часть из которых выполнена в растровом, а часть - в векторном виде. Обычно все полутоновые фрагменты выноски, схемы, карты и т. п. Таким образом достигается компромисс между качеством изображения и размером файла.

Все современные редакторы векторной графики, в том числе Flash, предоставляют возможность создания гибридной графики. Фактических можно даже назвать редакторами гибридной графики. Также гибридную графику создают и программы настольных издательств.

**Применение разных видов графики**

Если на свете существует два вида компьютерной графики, значит, это ко­му-то нужно. Давайте же выясним, кому и, главное, в каких случаях это нужно.

Главный козырь растровой графики - точность передачи сканированных изображений. При этом растровая графика занимает тем больший объем, чем больше само изображение, и не предоставляет никаких возможностей по его обработке (масштабированию, искажению, повороту, перекрашиванию). Главный козырь векторной графики - исключительные возможности обработки изображения. Недостаток - невозможность сохранения в векторном виде полутоновых изображений в близком к оригиналу виде. Исходя из этого, можно определить область применения для каждого вида компьютерной графики.

Итак, растровая графика применяется для:

сохранения полутоновых изображений (сканированные или изначально нарисованные на компьютере картины, фотографии);

создания небольших по размеру изображений для оформления программ или Web-страниц. В этом случае, как правило, критичны скорость вывода на экран и размер "ответственного" за вывод программного кода, а отнюдь не размер изображения.

Векторная графика лучше всего подойдет, если нужно:

сохранить штриховые изображения (карты, чертежи, рисунки карандашом, гравюры) в электронном виде;

создать небольшие изображения, которые'в дальнейшем будут всячески обрабатываться при выводе. Хороший пример таких изображений - шрифты формата TrueType, которые при выводе на экран не только мас­штабируются, но и раскрашиваются в разные цвета, поворачиваются и т. п. и растровую графику. Помните, и, разумеется, знайте об их преимуществах.

Большинство распространенных графических форматов - растровые. Сначала их и рассмотрим.

**BMP**

Формат BMP (BitMaP - битовая матрица) - простейший формат записи растровых изображений. Разработан фирмой Microsoft для сохранения графики в операционной системе Windows и совместимых с ней программах.

Для этого поддержка формата BMP была встроена непосредственно в ядро системы Windows. Также известен под названием DIB (Device Independent Bitmap - битовая матрица, не зависимая от устройства вывода).

Графика сохраняется в файлах с расширением bmp или dib (встречается очень редко). Поддерживает все цветовые видеорежимы. Графические данные могут быть сжаты с использованием простейшего алгоритма RLE (Run Length Encoding - кодирование с переменной длиной строки). Если сжатие не используется, размер графического файла может быть очень велик.

В настоящее время - один из самых распространенных графических форматов. Поддерживается практически всеми графическими программами. Из-за своей простоты требует для вывода очень мало системных ресурсов, поэтому основное его предназначение - хранение изображений, используемых как элементы пользовательского интерфейса операционной системы. В частности, именно в формате BMP хранятся системные "обои", заставки, иконки и т. п.

**PCX**

Формат PCX - один из самых старых графических форматов. Он был разработан в начале восьмидесятых годов не существующей ныне фирмой Z-Soft для собственного растрового графического редактора PC Paintbrush, работавшего в среде MS-DOS. Также поддерживался множеством других программ, работавших в среде DOS и Windows, и продолжает поддерживаться и поныне, хотя и является устаревшим.

**GIF**

Формат GIF (Graphic Interchange Format - формат обмена графикой) был разработан фирмой CompuServe в 1987 году для использования в собственной одноименной компьютерной сети. Получил огромное распространение в компьютерных сетях, в частности, в Интернете. Пожалуй, в настоящее время большинство компьютерной графики, использующейся в Web-дизайне и вообще для распространения изображений в Интернет, сохранено в этом формате.

Графика хранится в файлах с расширением gif. Поддерживаются цветовые видеорежимы до 256 цветов включительно. Для сжатия графики используется алгоритм LZW, разработанный израильскими математиками Лемпелом и Зивом. Графика может быть сохранена с чередованием (по-английски -interleaving) строк; в этом случае изображение как бы постепенно "проявляется" строка за строкой по мере загрузки файла.

В 1989 году формат GIF был расширен; новая версия стандарта получила название С} Р89А. Во-первых, была введена поддержка прозрачности или "прозрачного" цвета, т. е. один цвет из всех доступных на изображении мог быть помечен как прозрачный, и сквозь него будет "просвечивать" фон изображения. (Профессиональные графики часто называют "прозрачный" цвет альфа-каналом, по-английски - alpha channel или просто alpha.) Во-вторых, появилась возможность сохранять в одном файле несколько изображений, которые могут демонстрироваться как фильм (так называемые "анимированные GIF-файлы"; о них мы поговорим в главе 12).

Формат GIF прекрасно подходит для сохранения изображений с резкими цветовыми переходами. В частности, Web-дизайнеры создают в этом формате элементы оформления и рекламные баннерыдля своих страниц, а поддержка прозрачности и анимации им только на руку. Иногда в этом формате создаются начальные заставки и графические элементы интерфейса программ.

**PNG**

Формат PNG (Portable Network Graphic -- переносимая сетевая графика) разработан сообществом независимых профаммистов в качестве замены ус­таревающего и переходящего в разряд коммерческих продуктов формата GIF. Хотя он и поддерживается в настоящее время многими фафическими пакетами, однако большой популярности в Интернете пока не снискал, хотя автору встречались сайты с PNG-фафикой. Также используется для хранения фафики, разрабатываемой в пакете интернет-фафики Macromedia Fireworks.

Графика хранится в файлах с расширением png. Поддерживает все цветовые видеорежимы. Для сжатия графики применяется очень мощный алгоритм Deflate (буквально - Чс , :иии ), обеспечивающий более сильное сжатие по сравнению с LZW. Графика может быть сохранена с чередованием не только строк, но и столбцов, таким образом, изображение будет "проявляться" и по строкам, и по столбцам. Также поддерживаются 256 фадаций прозрачности (альфа-канала) против всего двух у GIF и автоматическая коррекция яркости.

Однако, по сравнению с GIF, формат PNG имеет и недостатки. Первый офомный недостаток - PNG не поддерживает анимацию. Правда, вряд ли это так уж актуально в связи с повсеместным переходом Web-аниматоров на Flash. Второй недостаток - файлы формата I'NCi больше, чем GIF, примерно на один килобайт из-за того, что в заголовке файла хранится гораздо больше информации. На мой взгляд, это тоже не очень значительный недостаток: получился файл чуть больше - ну и что? Так что, в принципе, вышеприведенные недостатки не повлияют на распространение формата PNG в Интернете.

Пока что формат PNG используется для хранения графики, разработанной в Fireworks, и в некоторых "продвинутых" сайтах. Однако, как уже говорилось, его поддерживают практически все графические пакеты.

**JPEG**

Формат JPEG (Joint Picture Encoding Group - группа кодировки неподвижных изображений) разработан одноименной группой программистов специально для распространения высококачественной графики в компьютерных сетях. Именно для этого он и используется в настоящее время. JPEG - второй по распространенности формат графики в Интернете.

Графика сохраняется в файлах с расширениями jpeg, jpe или jpg. Поддерживается только цветовой видеорежим TrueColor (24-битный цвет). Для сжатия графики используется исключительно мощный алгоритм под названием JPEG, фактически включающий в себя несколько алгоритмов сжатия для разных случаев. Этот алгоритм реализует сжатие данных с потерями, когда некоторая часть информации о цвете отбрасывается, и результирующий массив данных становится меньше. Во всех других форматах графики со сжатием применяются алгоритмы сжатия данных без потерь.

При использовании алгоритмов сжатия с потерями качество графики ухудша­ется. Чем сильнее сжатие, тем сильнее искажается изображение. Однако художник может регулировать процент сжатия, выбирая тем самым компромисс между качеством изображения и размером результирующего JPEG-файла.

Формат JPEG, в отличие от GIF и PNG, не поддерживает ни анимацию, ни прозрачность. Однако существует разновидность формата JPEG, называемая "прогрессивный JPEG" (progressive JPEG или p-JPEG), поддерживающая чередование строк.

Область применения формата JPEG достаточно узка - распространение высококачественной полутоновой графики в Интернете. Сканированные полутоновые изображения при использовании умеренного сжатия получаются очень даже неплохими. Особенно популярны "в народе" подборки картин Бориса Вальехо и других так называемых "фэнтезийных" живописцев, сохраненные в JPEG-файлах.

**TIFF**

Формат TIFF (Tag Image File Format - теговый файловый формат изображений) был разработан фирмой Aldus, разработчиком известнейшего пакета настольного издательства PageMaker, для другого своего продукта - растрового редактора PhotoStyler, не дошедшего до наших времен. Применяется для сохранения высококачественной полноцветной графики без потери качества для издательских целей. Часто используется для обмена высококачественной графикой между пользователями различных программ.

Графика записывается в файлы с расширением tif или tiff. Поддерживаются все цветовые видеорежимы, прозрачность и несколько алгоритмов сжатия: LZW, Deflate и JPEG. Фирменной особенностью этого формата является возможность записи в графический файл так называемых тегов: специальных примечаний, вносимых художником или самой программой графического редактора.

Существует две разновидности формата TIFF: совместимый с PC и с Apple Macintosh. Это вызвано различиями в архитектуре перечисленных компьютерных платформ.

Формат TIFF поддерживается всеми графическими программами профессионального уровня. Более того, поддержка какой-либо программой этого формата говорит об ее профессиональной ориентации.

**Векторные форматы**

Векторных форматов почему-то значительно меньше, чем растровых. Вероятно, это связано с тем, что реализовать обработку и вывод векторной графики несравнимо труднее, чем растровой. А люди - увы! - всегда идут по пути наименьшего сопротивления.

**Shockwave/Flash**

Разумеется, мы не могли не начать рассмотрение форматов векторной графики с "родного" формата Macromedia Flash. Как-никак, именно в нем нам предстоит сохранять свои творения.

Формат Shockwave/Flash разработан фирмой Macromedia для сохранения изображений и фильмов, созданных в пакете векторной графики Shockwave. Позднее на основе Shockwave был создан пакет интернет-графики Flash, "унаследовавший" этот формат. Поэтому говорят, что существует единый формат Shockwave/Flash.

Как вы уже знаете, фактически существует два формата представления графики Flash. Во-первых, это формат, в котором сохраняются подготавливаемые в среде Flash изображения и фильмы, - формат документов Flash. Во-вторых, формат, в котором хранится уже завершенная, экспортированная и подготовленная для публикации графика, которую можно загрузить в проигрыватель, - формат распространяемой графики Shockwave/Flash.

Документы Flash сохраняются в файлах с расширением fla. Экспортированная же графика хранится в файлах с расширением swf.

Помимо самого пакета Flash оба этих формата поддерживаются другими продуктами фирмы Macromedia: Dreamweaver, Fireworks и др. Также эти форматы поддерживаются некоторыми другими графическими программами.

**Windows Metafile и Enhanced Windows Metafile**

Формат Windows Metafile - простейший формат записи векторных изображений. Разработан фирмой Microsoft для сохранения векторной графики в операционной системе Windows и совместимых с ней программах. Поддержка этого формата встроена непосредственно в ядро системы Windows.

Возможности формата исключительно слабы, более-менее сложную графику сохранить в нем невозможно.

В настоящее время формат Windows Metafile, в отличие от BMP, распространен очень мало, хотя поддерживается практически всеми графическими программами и требует очень мало системных ресурсов для вывода и обработки. Используется для хранения векторных изображений в некоторых программах (например, начальных заставок или элементов пользовательского интерфейса). В частности, Microsoft Word свой комплект картинок (так называемый "клипарт" от английского clipart) хранит в формате Windows Metafile. Вероятно, фирма Microsoft, хочет показать таким образом, что этот формат тоже пригоден для распространения графики.

Формат Enhanced Windows Metafile - дальнейшее развитие Windows Metafile. Он также разработан Microsoft, однако "в народ" почему-то не продвигался и мало-мальски широкого распространения поэтому не получил. Автору за всю его достаточно длинную карьеру компьютерщика не попадалось на одного файла этого формата. Остается добавить, что формат предписывает сохранять графику в файлах и поддерживается многими современными графическими пакетами.

**Adobe Illustrator**

Этот формат был разработан фирмой Adobe для векторного графического редактора Illustrator. В настоящее время поддерживается практически всеми пакетами векторной графики и используется для обмена векторными изображениями между пользователями различных программ.

Графика сохраняется в файлах с расширением ai. Формат очень устойчив к сбоям, испорченный файл, как правило, с большой вероятностью все же можно прочитать.

**CorelDRAW!**

Был разработан фирмой Corel для векторного редактора CorelDRAW!. Фактически позволяет хранить гибридную графику. Иногда используется для обмена графикой.

Графика сохраняется в файлах с расширением cdr. Имеет несколько привлекательных возможностей по сравнению с Adobe Illustrator (например, сжатие графики, причем растровая и векторная графики сжимаются отдельно), но несовместимость различных версий формата и невысокая устойчивость к сбоям отнюдь не идут на пользу ею популярности.

Этот формат Flash не поддерживается. Чтобы импортировать графическое изображение в формате CorelDRAW!, вам придется преобразовать его в другой формат, например, Adobe Illustrator или Windows Metafile, используя другую графическую программу.

**Encapsulated PostScript**

Этот формат был разработан фирмой Adobe для обмена векторной графикой между пользователями различных программ. Собственно, это даже не формат, а целый язык, базирующийся на языке описания графики для высококачественных принтеров PostScript. Фактически, файл в этом формате можно скопировать на поддерживающий язык PostScript принтер, набрав в командной строке: copy graphic\_file.eps prn и он будет напечатан.

Графика сохраняется в файлах с расширением eps. Такой файл представляет собой простой текстовый документ, содержащий набор команд для принтера напечатать тот или иной примитив. Таким образом, EPS-файл может быть отредактирован в любом текстовом редакторе при наличии знания языка PostScript.

Поддерживается практически всеми графическими пакетами. Поэтому используется для переноса больших графических изображений между различными программами.

**VML**

Формат VML (Vector Markup Language - язык векторной разметки) разработан фирмой Microsoft для использования в собственной программе Web-обозревателя Internet Explorer версии 5.0 и более поздних. Так же, как и Encapsulated PostScript, это не столько формат, сколько текстовый язык описания векторной графики. По задумке, фрагменты таких описаний помещаются внутрь Web-страниц, среди обычного HTML-кода, и описывают их графические элементы. Таким образом, можно будет отказаться от традиционной Web-графики, представляющей собой внедренные элементы, хранящиеся в отдельных файлах.

**PDF**

Формат PDF (Portable Document Format - формат переносимых документов) был разработан фирмой Adobe для создания переносимых платформно-независимых электронных документов. Такие документы могут содержать, кроме форматированного текста, различную векторную и растровую графику, разбиваться на страницы, печататься на принтере или просматриваться на экране компьютера. Помимо этого, документы в этом формате имеют очень малый размер (используется сжатие, причем каждый вид графики сжимается по самому подходящему для него алгоритму), таким образом, их можно распространять через Интернет.

Документы этого формата сохраняются в файлах с расширением pdf. Они создаются с помощью пакета Adobe Acrobat, а читаются - с помощью программы чтения Adobe Acrobat Reader, распространяемой бесплатно. По названию этих двух программ формат PDF получил свое второе название - формат документов Acrobat.

Формат PDF получил огромную популярность для распространения электронных документов с богатым форматированием и графикой. Фактически, этот формат теперь стоит на втором месте, после HTML, по распространенности. Кроме Adobe Acrobat, создание документов PDF поддерживают множество других текстовых и графических пакетов. A Adobe Acrobat Reader -одна из популярнейших в мире программ.

**VRML**

Формат, точнее, язык, VRML (Virtual Reality Modeling Language - язык моделирования виртуальной реальности) был разработан группой независимых разработчиков. Он служит для создания так называемых виртуальных миров и распространения их через Интернет. Описание такого виртуального мира представляет собой текстовый файл с расширением vrm или vrml, содержащий набор команд на языке VRML. Такой файл может быть "проигран" с помощью особых программ -"проигрывателей" VRML; самым распространенным из них является Cortona VRML Client фирмы Parallel Graphics.

Хоть язык VRML и поддерживающее его программное обеспечение, предназначенное для создания и "проигрывания" виртуальных миров были широко разрекламированы в свое время, распространение он получил очень ограниченное. Связано это с тем, что мощностей современных компьютеров пока что не хватает для создания чего-то более сложного, чем странные геометрические фигуры из двух сфер и конуса. К тому же, виртуальная реальность, похоже, потеряла для простых людей свою притягательность. Поэтому будущее VRML так же туманно, как и настоящее...

**Глава 5. Рисование**

Познакомившись с двумя фундаментальными разновидностями компьютерной графики, выяснив их преимущества и недостатки, рассмотрев современные форматы записи графических данных в файл, можно приступить непосредственно к рисованию. Что мы и сделаем в этой главе.

Кратко напомним, что такое векторная графика. Векторное графическое изображение состоит из графических примитивов: простейших составных частей, каждая из которых может быть описана особой формулой. К графическим примитивам относятся прямые и кривые линии, простейшие геометрические фигуры (эллипсы, прямоугольники и пр.), текстовые надписи и импортированные растровые изображения. (Формально, если векторное изображение содержит в своем составе достаточно большое количество растровых изображений, говорят о гибридной графике, но мы не будем играть в терминологические игры, чтобы не запутаться.) Для каждого примитива в графическом файле сохраняются его тип и параметры описывающей его формулы: местоположение, размеры, цвет, кривизна, радиусы и т.п. Графическая программа, считав с диска такой файл, подставляет в соответствующие формулы нужные параметры, и мы получаем на экране или принтере исходное изображение.

Процесс преобразования векторной графики в растровую называется растеризацией. В частности, программа векторного графического редактора выполняет растеризацию всякий раз, когда выводит векторное изображение на экран или принтер - устройства изначально растровые. Обратный процесс - преобразование растровой графики в векторный вид - называется векторизацией. Векторизация растрового изображения чревата большими потерями в качестве, особенно плохо векторизуется высококачественное полутоновое фотографическое изображение.

Для выбора цвета линии и заливки создаваемых графических примитивов служит небольшая область в середине инструментария, носящая заголовок Colors.

Прежде всего, заметим, что в нижней части этой области находятся три небольшие кнопки. Эти кнопки включают особые цветовые режимы, мы рассмотрим их в главе 6. В данный же момент нас больше интересуют два селектора цвета, занимающие большую часть области Colors. С их помощью и задаются цвета графики. (Правда, кроме них, возможность выбора цвета предоставляют различные панели; реализуется это с помощью все тех же селекторов цвета.)

Как видите, селекторов цветов в области Colors два. Верхний из них устанавливает цвет линии об этом говорит значок карандаша слева от селектора. Нижний задает цвет заливки, что подтверждает значок опрокинутого ведра.

**"Линия"**

Как вы уже знаете, этот инструмент служит для проведения прямых линий. Чтобы выбрать его, щелкните в инструментарии кнопку или нажмите на клавиатуре клавишу <N>. После этого курсор мыши примет вид небольшого крестика, - это значит, что теперь вы можете проводить мышью прямые линии.

Чтобы провести на рабочем листе прямую линию с помощью инструмента "линия", сделайте следующее. (Имеется в виду, что вы уже выбрали этот инструмент.) Поместите курсор мыши в том месте, где у вас будет начало линии, и нажмите левую кнопку мыши. После этого, не отпуская эту кнопку, протащите мышь в то место, где у вас будет конец линии. Пока вы буксируете мышь, от начала линии за ее курсором будет тянуться "резиновая" линия. Чтобы поставить вторую точку - конец прямой - и тем самым нарисовать ее, просто отпустите левую кнопку мыши в нужной точке. Все, прямая линия создана.

Вы можете проводить таким образом линии любой длины и под любым углом. Если же вам нужно провести линию под углом, кратным 45°, то во время проведения линии удерживайте нажатой клавишу <Shift>. В этом случае проводимая линия будет автоматически "подгоняться" под подходящий угол.

Если включен модификатор "притягивание" различных частей вашей графики друг к другу, при рисовании линий это учитывается. При буксировке мыши, если какой-либо из концов линии окажется достаточно близко от другой линии (возможно, принадлежащей другому графическому примитиву), правее и ниже курсора мыши появится небольшой кружок, сигнализирующий об этом. Если теперь отпустить кнопку мыши, завершив рисование линии, ее конец будет "приклеен" к уже нарисованной линии, к которой он приблизился.

Вы можете изменять цвет линии с помощью уже знакомого вам селектора цвета линии. Селектор цвета заливки никакого влияния на рисуемые прямые линии не оказывает, так как у линий нет заливки.

**"Прямоугольник" и "эллипс"**

Оба этих инструмента очень похожи. Разница проявляется лишь в результате их применения. Поэтому мы и описываем их здесь вместе.

Прямоугольник создается следующим образом. Поместите курсор мыши в том месте, где у вас будет его левый верхний угол, и нажмите левую кнопку мыши. После этого, не отпуская эту кнопку, протащите мышь в то место, где у вас будет находиться правый нижний угол прямоугольника. Пока вы буксируете мышь, Flash отображает "резиновый" прямоугольник, так что вы всегда будете видеть, что получается. Переместив курсор в нужную точку отпустите левую кнопку мыши - и прямоугольник будет нарисован.

Эллипс создается аналогичным способом. Вы выбираете точку, соответствующую левому верхнему углу воображаемого прямоугольника, в который будет вписан эллипс, и буксируете мышь в точку правого нижнего угла этого прямоугольника. В процессе перетаскивания мыши Flash будет показывать вам "резиновый" эллипс.

Вы можете рисовать, таким образом, прямоугольники и эллипсы любой формы. Если же вам нужно нарисовать правильный квадрат или правильную окружность, то во время рисования удерживайте нажатой клавишу <Shift>.

Каждый нарисованный вами прямоугольник и эллипс будет иметь заливку.

Цвет заливки задается нижним селектором цвета в области Colors инструментария. А цвет линий, как вы помните, задается с помощью верхнего селектора цвета.

Прямоугольники, создаваемые с помощью одноименного инструмента, могут иметь скругленные углы. Для этого служит модификатор "скругленные углы". Включается он нажатием небольшой кнопки, появляющейся в области Options инструментария, если выбран инструмент "прямоугольник".

**"Карандаш"**

Этот инструмент позволяет рисовать линии любой (как говорят профессиональные компьютерные художники - свободной) формы. Работает он так же, как обычный карандаш, отчего и получил свое название. Вы нажимаете левую кнопку мыши и, не отпуская ее, рисуете линию, какая вам нужна. Пример такой линии показан на рис. 5.7.

Чтобы выбрать инструмент "карандаш", щелкните в инструментарии соответствующую кнопку или нажмите клавишу <Y>. Курсор мыши при этом примет вид карандаша.

Чтобы нарисовать с его помощью линию, поместите курсор мыши в точку, где она должна начинаться, и нажмите левую кнопку мыши. После этого, не отпуская левой кнопки, двигайте мышь по нужной траектории и, когда курсор мыши окажется в точке, где должен быть конец линии, отпустите кнопку мыши. Rash будет рисовать проводимую вами линию прямо во время движения мыши. Если вам нужно провести строго прямую горизонтальную или вертикальную линию, то во время рисования удерживайте нажатой клавишу <Shift>.

Вы можете задавать цвет линии, пользуясь верхним селектором цвета, расположенным в области Colors инструментария. Нижний селектор цвета никакого влияния на проводимые "карандашом" линии не оказывает.

Вы, наверно, заметили, что Flash автоматически сглаживает проведенную вами линию. Эта замечательная особенность, введенная программистами фирмы Macromedia, позволяет вам с помощью инструмента "карандаш" в любом случае проводить прямые линии. Как вы знаете, человеческая рука - не очень точный инструмент, и Flash это учитывает, сглаживая мелкие неровности проводимых вами линий. Однако, иногда такая "услужливость" может навредить, например, если вам нужно нарисовать что-то очень точно, со всеми неровностями и шероховатостями. Для регулирования степени сглаживания служит модификатор "сглаживание линий", представляющий собой кнопку, расположенную в области Options инструментария.

Вы можете заметить, что в правом нижнем углу этой кнопки имеется небольшая черная стрелка, направленная направо вниз. При нажатии этой кнопки на экране появится меню, предлагающее пользователю на выбор несколько вариантов. Такие кнопки с меню часто применяются во Flash.

**"Перо"**

Этот мощнейший инструмент поможет вам рисовать как прямые, так и кривые линии, точно указывая начальную и конечную точки и радиус кривизны. Кривые линии, рисуемые с помощью этого инструмента, называют также кривыми Безье по имени математика, выведшего формулу для описания этих кривых. (Прямую линию можно считать вырожденным случаем кривой Безье.) Кроме того, "перо" предоставляет возможность создания ломаных линий, состоящих из множества прямых или кривых отрезков, и весьма сложных геометрических фигур.

Чтобы выбрать инструмент "перо", щелкните в инструментарии соответствующую кнопку или нажмите клавишу <Р>. Курсор мыши при этом примет вид чертежного рейсфедера.

Рисование прямых с помощью "пера" выполняется следующим образом. Поместите курсор мыши в точку, где должно быть начало прямой, и щелкните левой кнопкой мыши. (Имейте в виду - нужно именно щелкнуть!) В этом месте на листе появится небольшая полая точка. Далее поместите курсор в конечную точку прямой и снова щелкните мышью. Прямая будет тотчас проведена.

Если вы продолжите щелкать мышью в разных местах на рабочем листе, то сможете построить ломаную линию, аналогичную показанной на рис. 5.12. Чтобы завершить рисование ломаной, при создании последней ее точки сделайте двойной щелчок мышью вместо одинарного. Вы также можете сразу после последнего щелчка выбрать другой инструмент или щелкнуть где-либо на пустом пространстве рабочего листа, удерживая на­жатой клавишу <Ctrl>.

Существует возможность проводить прямые линии под углом, кратным 45°.

Для этого вам нужно при втором щелчке удерживать нажатой клавишу <Shift>.

При создании ломаной вы можете замкнуть ее, для чего подведите курсор мыши к одной из уже созданных ее точек так, чтобы ниже и правее его появился небольшой кружок, и щелкните левой кнопкой мыши. В этом случае

Flash автоматически создаст замкнутую область и поместит в нее заливку, как показано на рис. 5.13.

**"Кисть"**

Этот инструмент ведет себя аналогично реальной кисти. С его помощью вы можете наносить мазки разной формы и толщины. Пример таких мазков показан на рис. 5.16.

Чтобы выбрать инструмент "кисть", щелкните в инструментарии соответствующую кнопку или нажмите клавишу <В>. Курсор мыши при этом может принимать разные формы; в зависимости от настроек, заданных с помощью модификаторов.

**"Ведро с краской".**

Этот инструмент служит для создания заливок. Заливкой называется особый графический примитив, заполняющий замкнутый контур, образованный линиями. Часто Flash создает заливки автоматически, например, когда вы рисуете, используя инструменты "прямоугольник", "эллипс" и "перо". Если же вы применяете другие инструменты для создания графики, о заливках вам придется позаботиться самостоятельно.

Инструмент "ведро с краской" ведет себя аналогично реальному ведру с краской: если "опрокинуть" его над замкнутым контуром, краска "разольется" и заполнит ее. Если же контур имеет "прореху", например, вы забыли его замкнуть, заливка создана не будет. Как видите, Flash не допускает, чтобы "краска" "вытекла" наружу.

Чтобы выбрать инструмент "ведро с краской", щелкните в инструментарии соответствующую кнопку или нажмите клавишу <К>. Курсор мыши при этом примет вид опрокинутого ведра с краской.

**Правка графики**

Итак, создавать графику мы научились. Теперь давайте выясним, как ее редактировать.

Под редактированием графики мы будем понимать изменение ее формы, а также ее полное или частичное удаление. Как добавить новые линии или фигуры вы уже должны знать - воспользуйтесь для этого одним из рассмотренных нами инструментов рисования.

Для изменения графики Flash предоставляет четыре мощных инструмента: "стрелка выделения", "дополнительная стрелка выделения", "лассо" и "ластик". Первые два инструмента предназначены для выделения и правки элементов изображения, третий - для "вырезания" из фигур "лоскутков" произвольной формы, а четвертый - полного или частичного удаления фигур или их частей. Пользуясь этими четырьмя инструментами, вы сможете проделывать со своей графикой все что угодно.

**Выделение графики**

Для выделения (или, как еще говорят, выбора) фрагментов графики для редактирования служит инструмент "стрелка выделения" (или просто "стрелка"). Щелкните кнопку в инструментарии или нажмите клавишу <V> на клавиатуре. Курсор мыши примет вид небольшой черной стрелки.

**Фрагментация и слияние графики.**

Прежде чем мы приступим к рассмотрению способов правки уже созданной графики, поговорим еще об одной особенности Flash. Это так называемая фрагментация графики. Суть ее в том, что Flash-графика в определенных случаях "разваливается" на независимые фрагменты. Иногда это бывает полезно, иногда не очень.

Фрагментации подвержены сложные фигуры, состоящие из множества графических примитивов, нарисованных разным цветом. Это основное правило фрагментации. Вы можете также разбить на фрагменты и монолитные фигуры, но об этом мы поговорим чуть ниже.

Чтобы лучше понять, что такое фрагментация, приведем небольшой пример. Нарисуем на листе Flash несложную фигуру, показанную на рис. 5.33. На этой фигуре мы и рассмотрим фрагментацию Flash-графики как явление.

Нарисовав фигуру, выберем инструмент "стрелка выделения" и щелкнем на пустом месте, чтобы снять любое выделение. После этого выделим какой-либо из концов линии, выступающих за пределы перечеркнутого круга. Вместо того чтобы выделить всю линию, Flash выделил только ее фрагмент.

Это произошло потому, что линия, рассекшая эллипс на четыре части (две половинки контура и две половинки заливки), и сама оказалась разделенной на три отрезка. На рис. 5.35 это проиллюстрировано наглядно.

Как видите, фрагментация налицо.

Монолитная, казалось бы, фигура "развалилась" сама собой на семь частей. И каждая ее часть является абсолютно независимым фрагментом.

С фрагментацией тесно связана и другая особенность Flash-графики - слияние фигур. Заключается оно в том, что если два или более графических фрагментов, нарисованных одним цветом, соприкасаются, они превращаются в единое целое.

**Правка графики**

Flash предоставляет практически те же возможности по редактированию созданной графики, что и другие программы векторных графических редакторов. Если вы знакомы с ними, вам будет совсем просто. Если же нет, не беда - все операции правки графики выполняются исключительно просто и наглядно, так что вы быстро приобретете нужные навыки. Рассмотрим их подробнее.

Проверьте, выбран ли у вас инструмент "стрелка выделения" на панели инструментов. Подавляющее большинство операций по правке графики выполняются с помощью именно этого, хорошо уже знакомого вам инструмента. Если для каких-либо операций понадобятся другие инструменты, мы специально предупредим об этом.

**Перемещение и удаление графики**

Перемещение и удаление графики - самые простые и наглядные операции. Мы рассмотрим их вместе.

Прежде всего, вы можете перемещать выделенные фрагменты. Для этого сначала выделите их. Далее установите курсор мыши на какую-либо часть выделенного фрагмента, нажмите левую кнопку мыши и, не отпуская ее, перетащите выделенный фрагмент на новое место, после чего отпустите кнопку. Если вам нужно переместить графику в направлении под углом, кратным 45°, удерживайте во время перетаскивания клавишу <Shift>.

Для точного перемещения графики можно воспользоваться клавишами-стрелками. Выделите нужный фрагмент и нажимайте клавиши-стрелки, пока он не окажется на нужном месте. При однократном нажатии на клавишу-стрелку происходит смещение на один пиксел. Если же вы при этом будете удерживать клавишу <Shifl>, при каждом нажатии выбранный фрагмент будет смещаться сразу на 8 пикселов.

На новое место можно переместить не сам выделенный фрагмент, а его копию. Для этого просто перетащите его, удерживая нажатой клавишу <Alt>. Вы также можете выбрать пункт Duplicate в меню Edit или нажать комбинацию клавиш <Ctrl>+<D>, чтобы создать новую копию выделенного фрагмента, а потом перетащить ее на новое место обычным способом.

Удалить ненужную графику совсем просто. Для этого выберите ее и либо нажмите клавишу <Del> или <Backspace>, либо выберите пункт Clear в меню Edit.

**"Притягивание" графики**

Ранее мы уже сказали, что графические фрагменты, перетаскиваемые по рабочему листу, могут "притягиваться" к другим фрагментам, линиям сетки и направляющим. Это сильно облегчает рисование сложных геометрических фигур: не нужно точно подгонять одну линию к другой. Обратите внимание, что "притягивается" та точка на линии или заливке, за которую вы "ухватились" мышью. Эта точка (назовем ее "точкой отображается в виде достаточно большой черной окружности).

Однако это может быть и вредно, например, если вы хотите нарисовать очень сложную и извилистую кривую с помощью карандаша или провести две линии очень близко друг к другу. Поэтому Flash предоставляет вам воз­можность временно отключить "притягивание".

Отключить "притягивание" очень просто. Если у вас выбран инструмент "стрелка выделения", становится доступным модификатор "притягивание". Он представляет собой кнопку. По умолчанию эта кнопка нажата, т.е. модификатор включен. Чтобы его отключить, "отожмите" эту кнопку. Впоследствии вы, конечно, сможете снова его включить.

Если инструментарий у вас скрыт, вы можете воспользоваться меню. Пункт-выключатель Snap to Objects меню View и одноименный пункт контекстного меню листа действуют так же. Но быстрее всего, вероятно, нажать комбинацию клавиш <Ctrl>+<Shift>+</>.

Отключить "притягивание" к линиям координатной сетки и направляющим также очень просто. В первом случае отключите пункт-выключатель Snap to Grid подменю Grid меню View или нажмите комбинацию клавиш <Ctrl>+<Shift>+<'>. Во втором случае, по аналогии, отключите пункт-выключатель Snap to Guides подменю Guides меню View или нажмите ком­бинацию клавиш <Ctrl>+<Shift>+<;>. (Подменю Guides и Grid также при­сутствуют в контекстном меню листа.) Чтобы временно отключить "притягивание" к линиям пикселной сетки, вы можете нажать клавишу <С>; когда вы ее отпустите, "притягивание" включится снова.

**Работа с буфером обмена.**

Все современные более-менее сложные Windows-приложения позволяют работать со стандартным буфером обмена Windows. Пользователь может помещать в буфер обмена фрагменты текстов, графических изображений, схем, программных компонентов и затем вставлять их в другое место этого же или совсем другого документа, открытого совсем в другом приложении.

Разумеется, Flash также поддерживает работу с буфером обмена. И предос­тавляет стандартный набор операций, которые мы сейчас перечислим.

Вы можете "вырезать" выделенный графический фрагмент с листа и помес­тить его в буфер обмена. При этом сам выделенный фрагмент с листа пропадает. Чтобы "вырезать" фрагмент, выделите его и выберите пункт Cut меню Edit или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<X>.

Вы также можете скопировать в буфер обмена выделенный графический фрагмент, оставив его на листе. Для этого выберите пункт Сору меню Edit или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<C>.

Чтобы вставить содержимое буфера обмена на лист, выберите пункт Paste меню Edit или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<V>. Содержимое буфера обмена будет помещено в центр листа.

Если вы хотите поместить содержимое буфера обмена в ту же позицию, клг на­ходится изначальный фрагмент, выберите пункт Paste in Place меню Edit или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<Shift>+<V>. Эта операция характерна только для Flash, в других программах она может и не поддерживаться.

**Глава 6. Работа с цветом**

Современные художники, работающие на современных компьютерах, имеют в своем распоряжении богатейший выбор всевозможных цветов. В настоящее время видеокарты могут выводить одновременно на экран миллионы цветов и оттенков - это значительно больше, чем может различить человеческий глаз. Как видите, аппаратные возможности компьютера не накладывают никаких ограничений на творца.

Проблема совсем в другом: сможет ли творец распорядиться этим богатством? Но это уже проблема самого творца. И уж никак не компьютера, и не Flash.

В этой главе мы поведем разговор о средствах, предлагаемых Flash для работы с цветом. Мы подробнее рассмотрим уже знакомый вам селектор цвета, дополнительные инструменты и специальные панели. А в самом конце мы поговорим о том, какие "подводные камни" подстерегают начинающего Flash-художника при публикации его творений.

**Использование селектора цвета**

Наличие селектора цвета - отличительная черта всех продуктов фирмы Macromedia. С ним мы уже знакомы. Мы знаем, как выбрать цвет с его по­мощью. Но всех его возможностей мы еще не знаем.

Давайте взглянем на рис. 6.1, на котором изображен открытый селектор цвета. Мы видим, что большая его часть занята набором небольших квадра­тиков, представляющих все доступные для выбора цвета, - так называемой палитрой. Чтобы выбрать требуемый цвет, поместите курсор мыши над нужным квадратиком и щелкните левой кнопкой мыши. Окно селектора цвета при этом закроется.

Однако 'селектор цвета предоставляет еще одну интересную возможность. Вы можете выбрать цвет, просто "взяв" его из любой точки экрана. Если вы присмотритесь к курсору мыши, то заметите, что он имеет вид небольшой пипетки. Щелкните этой пипеткой по нужной точке экрана, и цвет этой точки будет выбран селектором. (Окно селектора при этом должно быть открыто.) Таким образом, вы можете "взять" для работы цвет окна Windows, цвет открытой в графическом редакторе фотографии и т. п.

В верхней части окна селектора находится небольшое поле ввода. В этом поле отображается шестнадцатеричный код цвета, который находится под курсором мыши. Все цвета, отображаемые на экране компьютера, однозначно идентифицируются этим кодом. Попробуйте перемещать курсор (имеющий вид пипетки) по экрану и посмотрите, как будет изменяться содержимое этого поля ввода. Вы также можете ввести нужное значение кода цвета вручную, если, конечно, знаете его.

**Создание новых цветов. Смеситель**

Конечно, палитра селектора цветов весьма богата, и в некоторых случаях ее богатства может хватить. Однако для работы часто нужны цвета, которые в ней не присутствуют. Поэтому Flash предоставляет инструмент, позволяющий создавать новые цвета, применять их к выбранным на листе графическим фрагментам и добавлять при необходимости в палитру. Этот инструмент - особая панель, называемая смесителем цветов. Сейчас мы его рассмотрим.

Чтобы вызвать смеситель на экран, выберите в меню Window пункт Color Mixer. Вы также можете выбрать пункт Mixer подменю Panels контекстного меню листа или нажать комбинацию клавиш <Shift>+<F9>. Появившаяся на экране панель смесителя показана на рис. 6.8.

Вы можете щелкнуть расположенную в нижнем правом углу этой панели кнопку, имеющую вид треугольной стрелки, направленной вниз. После этого смеситель примет вид такой, как на рис. 6.9. В дальнейшем мы будем работать с развернутым смесителем, так как в таком виде он предоставляет больше возможностей, хоть и занимает больше места на экране.

Посмотрите на левую часть смесителя. Там все знакомо: два селектора цвета - линии и заливки, - которые можно выбирать мышью и раскрывать, три дополнительные кнопки, рассмотренные нами чуть выше. Эти элементы управления дублируют область Colors инструментария и могут пригодиться, если инструментарий скрыт.

В нижней части смесителя расположены два поля выбора цвета. Задать с их помощью нужный цвет очень просто. Для этого сначала выберите нужный селектор цвета - линии или заливки. После этого щелкните мышью по нужной точке центрального поля, чтобы выбрать сам цвет, затем щелкните по нужной точке правого поля, чтобы задать его яркость. После этого заданный вами цвет появится в выбранном селекторе и, одновременно, в расположенной слева области предварительного просмотра.

**Работа с заливками**

А теперь, после рассмотрения различных параметров линий, которые предлагает нам для работы Flash, поговорим о параметрах заливок. Как и в случае линии, выделите нужную заливку и...

**Виды заливки**

Вы уже знаете, что каждый замкнутый контур во Flash может быть заполнен так называемой заливкой. Также вы знаете, что Flash во многих случаях (в частности, при рисовании прямоугольников и эллипсов) сам создает такие заливки. И, конечно, вам уже известно, как задать цвет заливки. Об этом мы говорили еще в главе 5, когда рассматривали операции рисования фигур.

Но разговор о заливках еще не закончен. Сейчас мы рассмотрим еще некоторые параметры этих графических примитивов, которые вы можете задавать. И поможет нам в этом смеситель.

Flash предоставляет художнику несколько видов заливки. Все они задаются с помощью раскрывающегося списка, находящегося правее селектора цвета заливки и расположенного в панели смесителя. Рассмотрим все эти заливки подробнее.

**Простейшие виды заливки**

Простейшие виды заливки - это однотонная, сплошная заливка и ее отсутствие. В последнем случае опытные компьютерные художники часто говорят о так называемой прозрачной заливке, что в случае Flash неверно.

Чтобы отключить создание заливки при рисовании замкнутых фигур, выберите в раскрывающемся списке видов заливки пункт None. Это простейший вид заливки - больше вам не нужно задавать для него никаких параметров.

Чтобы включить создание однотонной заливки, выберите пункт Solid. В этом случае вы можете задать цвет заливки, воспользовавшись соответствующими селекторами цветов в инструментарии и на редакторе свойств. Если у вас открыт смеситель, вы можете воспользоваться им.

**Градиентная заливка**

Кроме простейших видов заливки, Flash может создавать и более сложные. В частности, градиентные заливки, в которых цвет плавно меняется от одного значения к другому, например, от белого к черному. Вы можете увидеть пример градиентной заливки, если посмотрите на любое окно любой программы, работающей в операционной системе Microsoft Windows 98 или более новой ее версии, - цвет его заголовка плавно меняется от синего к светло-голубому.

Прежде чем мы продолжим разговор о градиентных заливках (иногда их для краткости называют градиентами), давайте введем один термин. Назовем цвета, на основании которых будет создаваться градиент, ключевыми. Например, для заголовка окна Windows 98 ключевыми цветами будут синий и светло-голубой. Это поможет нам в дальнейшем.

Flash поддерживает два вида градиентной заливки: линейную и радиальную. В линейной градиентной заливке цвет меняется линейно при движении от одного конца фигуры к другому (рис. 6.14). В радиальной же градиентной заливке цвет меняется при движении от центра воображаемой окружности, вписанной в фигуру, к ее периферии (рис. 6.15). Как видите, названия видов закраски говорят сами за себя.

Чтобы создать линейную или радиальную градиентную заливку, откройте смеситель и выберите в раскрывающемся списке видов заливки соответственно пункт Linear Gradient или Radial Gradient. После этого смеситель сменит свой вид (рис. 6.16). Рассмотрим, что же предлагается нашему вниманию в этом случае.

Ниже раскрывающегося списка видов заливки находится линейка задания ключевых цветов заливки. По этой линейке перемещаются два указателя, задающие нужные цвета. Вы можете выбирать тот или иной указатель, просто щелкнув по нему мышью; выбранный указатель имеет не белую, а черную стрелку. Выбрав указатель, вы можете задать соответствующий ключевой цвет, указав в расположенном правее селекторе цвета или воспользовавшись двумя расположенными ниже панелями. Ну и, конечно, можно перемещать указатели, "захватывая" их мышью, чтобы подобрать нужное расстояние между ключевыми цветами на результирующей заливке.

**Работа с палитрами**

Когда мы знакомились с селектором цветов, то назвали весь доступный в его окне набор цветов палитрой. Как вы уже знаете, в эту палитру можно добавлять новые цвета и градиентные заливки, созданные с помощью смесителя. После этого вы сможете выбрать и применить в выделенной графике созданные вами цвета и заливки, воспользовавшись селектором цвета.

Настала пора подробнее поговорить о палитре и управлении ее содержимым.

Прежде всего, нужно сказать, что каждый документ Flash содержит свою собственную уникальную палитру, т. е. используемый в нем набор цветов.

Эта палитра сохраняется и в файле формата Shockwave/Flash, предназначенного для распространения. Отсюда следует, что, умело подобрав палитру и, возможно, ограничив количество используемых в изображении цветов, вы можете значительно уменьшить размер этого файла.

Вся работа над содержимым палитры осуществляется в панели Color Swatches. Чтобы вызвать ее на экран, выберите пункт-выключатель Color Swatches в меню Window или нажмите комбинацию клавиш <Ctrl>+<F9>. Можно также выбрать пункт Swatches подменю Panels контекстного меню листа. Сама панель Color Swatches показана на рис. 6.34.

Эта панель состоит из двух частей. В верхней ее части находится уже знакомая вам палитра цветов, в нижней - различные виды градиентных и графических заливок, как заданные по умолчанию самой профаммои, так и созданные пользователем. Вы можете перемещать мышью серую полосу, разделяющую эти части. Для этого установите на нее курсор мыши, нажмите левую кнопку, не отпуская ее, перетащите полосу верх или вниз и отпустите кнопку мыши.

Выбирать любые цвета или виды заливок можно, просто щелкая по ним мышью, как вы это делали в окне селектора цветов. Выбранный цвет или вид заливки выделяется белым квадратом. Если на рабочем листе у вас была выбрана какая-либо заливка, то выбранный в панели Color Swatches цвет или вид заливки будет тотчас к ней применен. (Для линий это почему-то не срабатывает.)

Вы уже знаете, как добавлять в палитру новые цвета. Рассмотрим теперь, что еще можно с ними делать.

Если Flash позволяет вам добавлять в палитру новые цвета и виды заливок, то уж конечно он должен позволить вам и удалять их. В самом деле, что делать, если палитра разрастется до нечеловеческих размеров и перестанет помещаться в панели Color Swatches? Только удалить ненужное. А удаляются ненужные цвета и виды заливки из палитры очень просто. Для этого выделите цвет или вид заливки, который вы хотите удалить, и выберите пункт Delete Swatch дополнительного меню.

Кроме того, есть возможность продублировать выбранный цвет или вид заливки, чтобы в дальнейшем внести в него изменения. Для этого выделите нужный цвет или вид заливки, который вы хотите удалить, и выберите пункт Duplicate Swatch дополнительного меню.

Если вы создали, с вашей точки зрения, исключительно удачную палитру и хотите сохранить ее для дальнейшего использования, то можете сохранить ее на диске. Палитра сохраняется в файле с расширением (так называемые файлы цветовых наборов Rash) или act (цветовые таблицы). Цветовые наборы Flash могут быть использованы только в самом Flash, а цветовые таблицы поддерживаются также другими графическими программами, в частности, Macromedia Fireworks и Adobe Photoshop. Кроме того, вы можете импортировать таблицы цветов из файлов GIF. Учтите только, что градиенты импортировать из и экспортировать в файлы цветовых таблиц и GIF нельзя.

Чтобы сохранить цветовую палитру в файле, выберите пункт Save Colors дополнительного меню. На экране появится стандартное диалоговое окно сохранения файла Windows. Задайте имя файла, выберите его формат и нажмите кнопку сохранения.

Также вы можете сохранить созданную вами палитру в качестве палитры по умолчанию, которая будет использоваться во всех вновь создаваемых документах Flash. Для этого выберите пункт Save as Default дополнительного меню.

В дальнейшем сохраненные вами палитры можно снова загрузить. Чтобы загрузить палитру из файла, затерев существующую, выберите в дополнительном меню пункт Replace Colors. Если же вы хотите добавить цвета из загружаемой палитры в существующую палитру, выберите пункт Add Colors. В обоих этих случаях на экране появится стандартное диалоговое окно открытия файла Windows. Выберите формат файла, сам файл и нажмите кнопку открытия. Таким образом вы можете импортировать палитры из файлов цветовых наборов Flash, цветовых таблиц и файлов GIF.

Чтобы загрузить палитру по умолчанию, выберите в дополнительном меню пункт Load Default Colors. А чтобы удалить из палитры все цвета, оставив только черный и белый, и все виды заливки, кроме простейшего черно-белого линейного градиента, выберите пункт Clear Colors.

Пункт Web 216 дополнительного меню позволит вам загрузить так называемую безопасную палитру цветов WWW. Эта палитра включает в себя 216 цветов, которые гарантированно будут отображаться на любом компьютере. Если вы создаете изображения для публикации в Интернете, то лучше всего использовать именно эту палитру.

Последний пункт дополнительного меню Sort by Color позволит вам отсортировать цвета в палитре по оттенку. К сожалению, вернуть обычный порядок сортировки цветов после этого невозможно.

**Глава 7. Работа с текстом.**

Текст, помещаемый на рабочий лист, находится в так называемом текстовом блоке. Это особый вид графических примитивов, служащий именно для размещения текста. (Опытные компьютерщики такие "вместилища" называют контейнерами; в частности, текстовый блок - контейнер для текста; по аналогии, рабочий лист - контейнер для графики.) Блок текста имеет прямоугольную форму, собственно текст помещается внутри его границ.

Создание текстовых блоков выполняется с помощью инструмента "текст". Чтобы выбрать его, щелкните в инструментарии кнопку или нажмите на клавиатуре клавишу <Т>. После этого курсор мыши примет вид небольшого крестика с буквой "А".

Это означает также, что вы можете перенести текст из другого Windows-приложения, например, Microsoft Word. В случае достаточно большого текста это может быть более удобным, чем набор его прямо во Flash, т.к. Word содержит, кроме всего прочего, систему проверки орфографии, которая, несмотря на все свои недостатки, все же может принести пользу.

Вы, вероятно, уже заметили, что при наборе текста созданный нами тексто­вый блок растягивается только по вертикали. Если текст подходит к правой границе текстового блока, то переносится на следующую строку, для чего блок растягивается вниз. Горизонтальный же размер (ширина) остается постоянным, таким, каким мы его задали при создании текстового блока. Подобный текстовый блок называется блоком с фиксированной шириной (см. рис. 7.2).

Текстовый блок создается примерно так же, как прямоугольник. Поместите курсор мыши в том месте, где у вас будет его левый верхний угол, и нажмите левую кнопку мыши. Затем, не отпуская эту кнопку, протащите мышь в то место, где у вас будет находиться правый нижний угол текстового блока. Пока вы буксируете мышь, Flash будет отображать синий "резиновый" прямоугольник, так что вы всегда будете видеть, что же у вас получается. Переместив курсор мыши в нужную точку, отпустите левую кнопку мыши - текстовый блок будет нарисован, и внутри него появится текстовый курсор (рис. 7.2). Наберите нужный текст и либо создайте новый текстовый блок, либо смените инструмент на "стрелку выделения".

Для переноса текста на новую строку при наборе текста вы можете использовать клавишу <Enter>. Также вы можете редактировать текст, пользуясь клавишами-стрелками, <Backspace>, <Del> и др.

Пользуясь мышью или обычными, используемыми в других программах, клавиатурными комбинациями, можно выделять фрагменты текста и работать с буфером обмена Windows. В частности, вы можете выделять слова двойным щелчком мыши и выделять все содержимое текстового блока нажатием клавишной комбинации <Ctrl>+<A>.

Вы также можете, выбрав инструмент "текст", просто щелкнуть по листу мышью.

**Форматирование**

Текстовые возможности Flash отнюдь не исчерпываются набором про­стого текста, разбитого на абзацы с помощью клавиши <Entei>. Ваш текст может содержать фрагменты, набранные разными шрифтами, с разным выравниванием и отступами. Фактически, текстовые возможности Flash равны аналогичным возможностям языка HTML, а то и превосходят их.

**Форматирование текста**

Форматирование текста во Flash, а именно, задание шрифта и цвета, выполняется с помощью все того же редактора свойств.

Сразу заметен раскрывающийся список шрифтов, который позволяет вам выбрать собственно шрифт. В этом списке перечислены все шрифты, установленные в системе. Когда вы выбираете в раскрытом списке шрифт, левее в небольшом окошке для вашего удобства показывается пример текста, на бранного этим шрифтом.

Правее этого списка находится поле ввода с регулятором, с помощью кото­рого задается размер (или, как говорят полиграфисты, кегль) шрифта в пунктах. Правее же поля ввода размера находятся селектор цвета шрифта, а также кнопки В (включение или отключение "жирности" шрифта) и I (включение или отключение курсивного начертания). Все это знакомо вам по текстовым редакторам.

Чтобы изменить какие-либо параметры всего текста в текстовом блоке или его фрагмента, вам нужно будет выделить этот текст или фрагмент. Это выполняется так же, как и в других программах редактирования текста. Если же вы хотите изменить форматирование всего текста в текстовом блоке, просто выделите текстовый блок одиночным щелчком мыши при выбранном инструменте "стрелка выделения".

Flash поддерживает векторные шрифты стандартов TrueType и Туре 1.

Для выбора шрифта вы можете воспользоваться подменю Font меню Text. Там перечислены все установленные в системе шрифты. Аналогично выбрать размер шрифта можно, воспользовавшись подменю Size меню Text, -там перечислены наиболее употребительные размеры шрифтов. Это может быть полезно, если редактора свойств нет на экране. .

В том же меню (Text) есть пункт Style, позволяющий задать и другие параметры шрифта. В частности:

пункт Bold (или комбинация клавиш <Ctrl>+<SMft>+<B>) включает или отключает "жирность" шрифта;

пункт Italic (или комбинация клавиш <Ctrl>+<Shift>+<I>) включает или отключает курсивное начертание.

Оба этих пункта работают как выключатели. А пункт Plain (и комбинация клавиш <Ctrl>+<Shift>+<P>) позволяют вернуть шрифту обычный вид.

Но вернемся к редактору свойств.

Ниже и левее списка шрифтов находится другое поле ввода с регулятором. С его помощью задается так называемый трекинг, т. е. дополнительное пространство между символами. Это иногда бывает очень полезно, например, для выделения заголовков - просто задайте для шрифта заголовка большой трекинг, чтобы буквы отстояли далеко друг от друга.

Значение трекинга вводится в пунктах, причем можно вводить как положительные, так и отрицательные значения. В первом случае символы текста раздвигаются, а во втором - сближаются. При этом главное - не переборщить с трекингом, иначе символы или "разбредутся" слишком далеко друг от друга, перестав быть единым текстом, или сольются в точку.

Задавать значения трекинга вы также можете, использовав подменю Tracking меню Text. Пункт Increase (и комбинация клавиш <Ctrl>+<Shift>+<->>) увеличивает трекинг, а пункт Decrease (и комбинация клавиш <Ctrl>+<Shift>+<^>) - уменьшает. Пункт Reset (и комбинация клавиш <Ctrl>+<Shift>+<t>) позволяет вернуть значение трекинга по умолчанию.

Также в редакторе свойств находится флажок Kern. Этот флажок включает или отключает кернинг - специальное управление пространством между символами. Дело в том, что некоторые пары символов требуют задания различных значений промежутка между ними, например, промежуток между "А" и "W" должен быть меньше, чем между "А" и . Именно этим и занимается кернинг, благодаря ему текст выглядит более ровным.

Правее поля задания трекинга расположен еще один раскрывающийся список. Он задает положение символов относительно базовой линии текста - воображаемой линейке, по которой "пишется" текст. В нем доступны три пункта:

О Normal - обычное положение текста, когда он "лежит" на базовой линии;

О Superscript - верхний индекс;

О Subscript - нижний индекс.

Вы также можете воспользоваться пунктами-выключателями Superscript и Subscript подменю Style меню Text. Первый превращает текст в верхний ин- декс, второй - в нижний.

**Поддержка шрифтов во Flash**

А теперь самое время поговорить о поддержке различных шрифтов во Flash.

Сразу скажем, что для создания текстовых блоков Flash позволяет использовать только векторные шрифты, т. е. шрифты формата TrueType и Adobe Туре 1. Любые растровые шрифты, даже если они и установлены в системе (а они наверняка установлены, ведь большинство системных шрифтов Windows - растровые), игнорируются и не показываются в списке шрифтов редактора свойств.

Когда вы экспортируете готовое изображение в формат Shockwave/Flash, то Flash фактически сохраняет в результирующем SWF-файле все использованные в изображении шрифты. Этот процесс называется внедрением шрифтов. Благодаря этому проигрыватель Flash сможет вывести текст, набранный этими шрифтами, даже если они не установлены на компьютере пользователя. Таким образом, вам не нужно ограничивать себя небольшим набором стандартных шрифтов или распространять недостающие шрифты отдельно, чтобы пользователь увидел ваше изображение в первозданном виде.

**Форматирование абзаца.**

Форматирование целого текстового абзаца, а именно, задание выравнивания и отступов, выполняется с помощью все того же редактора свойств. Для этого используются другие элементы управления, не рассмотренные нами ранее. Но сейчас-то мы их рассмотрим.

Чтобы отформатировать какой-либо абзац, сначала поставьте в него текстовый курсор. (Для этого сначала выберите инструмент "текстовый блок".) Вы также можете выделить какой-либо фрагмент текста в этом абзаце - это не сыграет никакой роли.

Выравнивание текста в абзаце задается с помощью находящегося в верхнем правом углу редактора свойств набора из четырех кнопок-переключателей. Также вы можете воспользоваться пунктами подменю Align меню Text.

Перечислим все кнопки этого набора слева направо:

первая кнопка задает выравнивание по левому краю (пункт Align Left подменю Align меню Text или комбинация клавиш <Ctrl>+<Shift>4-^T >);

вторая - выравнивание по центру (пункт Align Center или комбинация клавиш <Ctrl>+<Shift>+<C>);

третья - выравнивание по правому краю (пункт Align Right или комби­нация клавиш <Ctrl>+<Shift>+<R>);

четвертая - - полное выравнивание по обоим краям (пункт Justify или комбинация клавиш <Ctrl>+<Shift>+<J>).

Ниже этого набора кнопок находится кнопка Format. При нажатии этой кнопки на экране появится небольшое диалоговое окно Format Options, с помощью которого вы можете задать другие параметры абзаца.

**Параметры текстового блока.**

Задавать параметры можно не только для фрагмента текста или отдельного абзаца, но и для всего текстового блока. В частности, вы можете сделать текстовый блок вертикальным или дать возможность пользователю выделять находящийся в нем текст.

Чтобы сделать текстовый блок вертикальным, нажмите кнопку, находящуюся левее набора кнопок, задающих выравнивание. При этом на экране появится небольшое меню, в котором вы сможете выбрать нужный пункт. Всего таких пунктов три:

Horizontal - горизонтальный текстовый блок (значение по умолчанию);

Vertical, Left to Right -- вертикальный текстовый блок с направлением текста слева направо;

Vertical, Right to Left -- вертикальный текстовый блок с направлением текста справа налево.

Для европейских языков с направлением текста слева направо второй и третий пункты меню равнозначны. Вы можете выбирать любой.

Ниже кнопки задания направления находится кнопка поворота текста. Изначально в вертикальном блоке текста буквы расположены горизонтально, т. е. одна над другой. Если вы нажмете кнопку поворота (она работает как кнопка-выключатель), то буквы текста будут повернуты на 90°.

Включение флажка Use Device Fonts заставляет Flash использовать для отображения текста, введенного в текстовом блоке, шрифты-псевдонимы. Включение флажка Use Device Fonts аналогично выбору любого шрифта-псевдонима в раскрывающемся списке шрифтов редактора свойств, но затрагивает не выделенный фрагмент текста, а весь текст в текстовом блоке.

**Поля ввода.**

Что такое поле ввода, вы должны знать. Это своего рода небольшой текстовый редактор, помещенный в окно Windows-программы и предназначенный для ввода одно- или многострочного текста. При этом Windows сама управляет текстовым вводом: вы просто набираете символы, пользуетесь клавишами-стрелками, клавишами <Backspace>, <Del>, выделяете текст, пользуетесь буфером обмена, как и в "большом" текстовом редакторе. Программисты фирмы Microsoft много поработали над тем, чтобы сделать поля ввода по-настоящему удобными.

Также вы знаете, что поля ввода поддерживаются языком HTML. Вы можете поместить на своей Web-странице форму с полем ввода, в котором посетитель вашего сайта должен будет ввести, например, свое имя или идентификационный код. Такие поля ввода также предоставляют пользователю базовые возможности редактирования текста (правда, это зависит от операционной системы, под которой работает программа Web-обозревателя).

Flash также поддерживает создание полей ввода. Давайте рассмотрим, как это делается. Однако предупредим сразу, что эта информация дается "на вырост". Сейчас мы рассмотрим только создание полей ввода, как обрабатывать введенные в них данные, вы узнаете в части 3 этой книги.

Итак, прежде всего нам нужно создать обычный текстовый блок. Переключитесь на инструмент "стрелка выделения" и выделите его. После этого обратитесь к редактору свойств, найдите в его верхнем левом углу раскрывающийся список и выберите в нем пункт Input Text.

В поле ввода Maximum Characters задается максимальное количество символов, которое в него сможет вводить пользователь. Если вы не хотите ограничивать это количество, введите 0.

Вы уже знаете, что при экспорте изображения Flash помещает в файл Shockwave/Flash описания всех использованных в изображении шрифтов. Благодаря этому проигрыватель Flash может отобразить это изображение на любом компьютере, даже если нужные шрифты на нем не установлены. Од­нако это может сильно увеличить размер SWF-файла. Есть, конечно, два компромиссных способа решения проблемы, которые были описаны ранее в этой главе, но сейчас речь не об этом.

Дело в том, что Flash позволяет вам выбрать символы использованного в поле ввода шрифта, описания, которых будут внедрены в изображение Flash при его экспорте. Благодаря этому вы можете внести в результирующий файл только те символы шрифта, которые действительно нужны для вывода текста, а значит, уменьшить размер этого файла до необходимого минимума. Выбор символов выполняется в диалоговом окне Character Options, которое появляется на экране при нажатии кнопки Character редактора свойств.

**Динамические текстовые блоки.**

Часто возникает необходимость не показывать какой-то статический текст, а выводить результаты работы какой-либо программы. Например, встроенная в изображение Flash программа может извлекать новости с сайта "Компьюленгы . с "Апорта" - прогноз погоды, а с "Рэмблера" - курсы валют, добавлять к ним текущее время и все это в виде текста выводить на экран. Для этого используется так называемый динамический текстовый блок. (Обычные же текстовые блоки называют иногда статическими.) Пожалуй, нужда в таком динамическом текстовом блоке возникает чаще, чем в обычном поле ввода.

Для вывода текста в динамический блок используются уже знакомые вам переменные. Программа помещает результат своей работы в переменную, а динамический текстовый блок потом забирает их оттуда и выводит на экран. Собственно вывод текста при этом выполняется автоматически самим Flash.

Как и в случае поля ввода, создайте обычный текстовый блок и выделите его с помощью "стрелки". После этого обратитесь к редактору свойств, найдите в его верхнем левом углу раскрывающийся список и выберите в нем пункт Dynamic Text.

**Работа с символами текста как с графикой.**

Для желающих поглумиться над текстом Flash предоставляет возможность работы с символами текста как с графическими фрагментами. Это означает, что вы можете менять форму отдельных символов текста, как если бы это были обычные графические фрагменты, нарисованные с помощью обычных инструментов рисования.

Прежде чем править символы как графику, вам нужно превратить текст в набор графических фрагментов. Для этого выберите текстовый блок, пользуясь инструментом "стрелка выделения", и выберите пункт Break Apart в меню Modify, в контекстном меню текстового блока или нажмите комбинацию клавиш <Ctrl>+<B>. Каждый символ созданного вами текста преобразуется в набор графических фрагментов, который впоследствии будет выделен.

Вы можете применять для бывшего текста любые знакомые вам приемы для изменения формы и цвета графики. В частности, можно расположить символы такого текста по окружности или кривой - похоже, это единственный доступный во Flash прием сделать фигурный заголовок. Также вы можете создавать новые, причудливые шрифты или просто изменять форму отдельных символов. Посмотрите на рис. 7.17, там изображен результат преобразования формы обычной латинской буквы "U" с помощью инструмента "стрелка выделения".

Есть еще одна область применения этой возможности. Помните, мы говорили, что при экспорте изображения в формат Shockwave/Flash все использованные в нем шрифты помещаются в результирующий файл? Мы еще упомянули о двух компромиссных путях решения этой проблемы. Так вот, есть возможность решить ее третьим, более удачным путем. А именно, преобразовать текст, набранный каким-либо сверхсложным и сверхэкзотическим шрифтом, в графику. Ведь чтобы отобразить набор кривых, в который превратится наш текст, проигрывателю Flash не нужно вообще никаких шрифтов. А значит, SWF-файл вашего изображения станет меньше.

Единственный недостаток: вы не сможете преобразовать набор графических фрагментов обратно в текст. А значит, что подобный текст не подлежит редактированию. Поэтому перед преобразованием текста в графику убедитесь, что он не содержит ошибок.

**Подстановка шрифтов.**

Мы много говорили о том, как Flash сохраняет информацию об использованных в изображении шрифтах в файле Shockwave/Flash. Мы узнали, что при отсутствии того или иного шрифта в большинстве случаев изображение будет отображено правильно, т.к. описания шрифтов включаются в сам результирующий файл. Если, конечно, разработчик специально не отключил включение описаний этих шрифтов.

Однако иногда бывает и другая ситуация. Вы отдаете документ Flash (FLA-файл) своему коллеге, а тот пытается открыть его на своем компьютере. И тут выясняется, что в его системе нет шрифта, которым был набран некий текст. Что случится в этом случае?

Как только Flash понадобится для отображения текста этот самый отсутствующий шрифт, он выдаст предупреждение об этом. Давайте рассмотрим его во всех подробностях.

Итак, текст этого предупреждения гласит, что Flash для вывода текста нужен шрифт, который в данный момент не установлен в системе. Чтобы вывести этот текст, Flash предлагает вам выполнить так называемую подстановку шрифта, т.е. заменить отсутствующий шрифт другим, установленным в системе. Конечно, при этом текст исказится, но зато вы сможете просмотреть, отредактировать документ и даже экспортировать его в формат Shockwave/Flash.

Интересная особенность Flash: после подстановки шрифтов он не сохраняет об этом сведений. Давайте поясним это на такой ситуации. Предположим, что вы открыли документ Flash, присланный вашим коллегой, и у вас в системе не оказалось необходимого шрифта. Вы заменили этот шрифт (назовем его изначальным) на другой, установленный в вашей системе, и открыли-таки документ. После этого вы изменили что-то в документе, возможно, отредактировали текст, но не меняли изначальный шрифт на другой. Затем вы сохранили этот документ. Позднее вы все-таки нашли нужный вам шрифт, установили его и снова открыли этот документ. И после этого он будет набран именно тем, изначальным шрифтом.

**Глава 8. Импорт графики.**

Всем хорош Flash. И рисовать на нем можно, и создавать анимацию, и даже писать довольно сложные программы - ну просто мастер на все руки. Неудивительно, что так популярен он на просторах Сети.

Однако Flash может делать не все. Некоторые задачи лучше всего выполнить в других программах, а потом импортировать во Flash результаты их работы. И профессиональные Flash-художники так и делают. В самом деле, каждую задачу лучше всего решать с помощью наиболее подходящего инструмента. Ведь никто не забивает гвозди микроскопом, хотя теоретически это вполне возможно.

Для забивания гвоздей служит молоток (если кто этого не знает). Для редактирования растровой графики служат программы растровых редак­торов, например, Adobe PhotoShop и Macromedia Freehand. Для очень сложной векторной графики есть более мощные, чем Flash, векторные редакторы: Corel DRAW! и Macromedia Freehand. Есть также специальные графические программы, например, AutoDesk AutoCAD, служащая для создания чертежей и схем, и Microsoft Visio, где так удобно и приятно рисовать бланки.

Каждый из перечисленных выше программных пакетов - в своей области узкий специалист. Полнота его односторонняя, как у недоброй памяти флюса, но свою задачу он знает великолепно.

А что же Flash? Задача Flash - принести в Интернет компактную векторную графику и интерактивную анимацию, которая не слишком бы "нагружала" маломощные клиентские компьютеры. Fl^h не работает со сверхсложной графикой, не создает сверхмощные программы и не редактирует растровые изображения. Но свою прямую задачу он выполняет на "отлично", о чем говорят цифры, показывающие его распространение в Сети. У создателей Flash не было намерения объять необъятное.

**Поддержка графических форматов**

Сначала поговорим о том, какие графические форматы поддерживает Flash. Это важно узнать в самую первую очередь, чтобы потом не пытаться импортировать файл заведомо не поддерживаемого формата и не потерять случайно половину содержащихся в файле графических данных.

Если вы хотите импортировать во Flash файл не поддерживаемого им графического формата, вам придется найти программу, поддерживающую этот формат. С ее помощью вы сможете преобразовать этот файл к формату, поддерживаемому Flash. Однако будьте готовы к тому, что при таком преоб­разовании возможны потери некоторой информации.

**Список поддерживаемых форматов**

Приведем список графических форматов, которые можно импортировать во Flash. Этот список оформим в виде табл. 8.1.

Таблица 8.1. Список графических форматов, которые можно импортировать во Flash

|  |  |
| --- | --- |
| Название формата | Расширение файлов |
| Adobe Illustrator | ai, eps |
| AutoDesk AutoCAD | dxf |
| FutureSplash | spl |
| GIF (обычный и анимированный) | gif |
| JPEG | jpg, jpe, jpeg |
| Macromedia Freehand | fh7,ft7,fh8, ft8, fh9, ft9,fh10 |
| Macromedia Shockwave/Flash | swf  J- "1 |
| PNG -■ ■■' ■ ■ | png |
| Метафайлы Windows | wmf |
| Растровые файлы Windows ...^. | bmp |
| Расширенные метафайлы Wndows | emt |

**Импорт графики**

Теперь рассмотрим, как же выполняется сам импорт графики.

Существует два способа импортировать во Flash графику, созданную в другом графическом пакете. Мы рассмотрим их оба.

Первый способ вам уже знаком. Мы пользовались им, когда создавали графическую заливку в главе 6. Давайте вспомним, что мы тогда делали.

Для того чтобы импортировать во Flash чужую графику, выберите пункт Import в меню Fiic или нажмите комбинацию клавиш <Ctrl>+<R>. На экране появится стандартное диалоговое окно открытия файла Windows. Найдите нужный файл и нажмите кнопку открытия файла этого диалогового окна. Как видите, это очень просто.

Второй способ еще проще. Запустите программный пакет, откройте в нем нужный файл или создайте его заново, скопируйте нужный графический фрагмент или все изображение в буфер обмена Windows и вставьте его во Flash.

Импортированное вами графическое изображение будет тотчас помещено на рабочий лист и выделено. Вы можете перемещать его, пользуясь "стрелкой выделения", помещать его в буфер обмена и выполнять другие операции, доступные для выделенных графических фрагментов.

Кроме того, импортированное изображение помещается в качестве образца в библиотеку. Так что вы можете в дальнейшем создавать сколько угодно экземпляров этого изображения на листе, а размер файла Flash при этом не увеличится. (О библиотеках, образцах и экземплярах см. главу 10.)

Кстати, есть возможность импортировать графическое изображение прямо в библиотеку, не помещая его на рабочий лист. Для этого выберите пункт Import to Library в меню File. На экране появится стандартное диалоговое окно открытия файла Windows. Найдите нужный файл и нажмите кнопку открытия файла этого диалогового окна.

**Adobe Illustrator**

Flash позволяет импортировать файлы, созданные в версиях 8.0 или более ранних.

Перед импортом графики Illustrator во Flash разгруппируйте все сгруппированные графические фрагменты. Если этого не сделать, Flash не сможет правильно их обработать.

После выбора файла Illustrator в стандартном диалоговом окне открытия файла на экране появится диалоговое окно выбора параметров импорта Illustrator Import. Оно показано на рис. 8.3.

**AutoDesk AutoCAD**

Flash позволяет импортировать чертежи и схемы AutoCAD версии 10 или более поздней. Учтите, что Flash поддерживает только файлы, сохраненные в текстовом формате. Flash не поддерживает трехмерные чертежи и схемы - только двумерные.

Поскольку AutoCAD не поддерживает стандартные системные шрифты, при импорте возможно искажение текста. Также AutoCAD не поддерживает заливки, поэтому вся импортированная графика будет состоять из одних контуров.

**Глава 9. Работа с графическими фрагментами.**

Здесь мы рассмотрим простейшие манипуляции над графическими фрагментами.

**Изменение порядка наложения**

Вы, вероятно, уже заметили, что графические фрагменты, расположенные на рабочем листе, могут перекрывать друг друга. При этом один фрагментможет оказаться наверху, другой - внизу, а третий - между первыми двумя. Как видите, все графические фрагменты на листе "сложены" согласно особому порядку, называемому порядком перекрытия. (Опытные компьютерные художники часто говорят в этом случае о z-Koopduname, по аналогии с координатами х и у.)

Отсчет в порядке перекрытия ведется, начиная с самого нижнего фрагмента, имеющего номер 0, до самого верхнего. Исходя из этого, фрагменты с большим номером в порядке перекрытия находятся выше фрагментов с меньшим номером. Как видите, здесь все просто.

Когда вы рисуете что-то на уже существующем изображении, Flash руководствуется только одним правилом: то, что нарисовано позже, лежит выше. Поэтому, если вы зачеркнете нарисованный прямоугольник, линии зачеркивания будут находиться поверх него. Есть только одно исключение из этого правила: группы и экземпляры библиотечных образцов всегда нахо­дятся выше обычной графики. Чтобы переместить обычную графику выше, вам придется или сгруппировать ее, или преобразовать в образцы и помес­тить на рабочий лист их экземпляры.

Часто бывает необходимо переместить какой-либо графический фрагмент выше или ниже в "стопке" фрагментов, "сложенных" на листе. Для этого Flash предлагает различные пункты подменю Arrange, расположенное в меню Modify.

Если вам нужно переместить какой-либо графический фрагмент выше или ниже в порядке перекрытия, выберите соответственно пункт Bring to Front или Sent to Back этого подменю. Вы также можете нажать комбинацию кла­виш <аг1>+<8ЫЙ>+<Т>или <Ctrl>+<Shift>+<;> соответственно.

Если вам нужно переместить какой-либо графический фрагмент на самый верх или в самый низ, вам следует выбрать соответственно пункт Bring Forward или Sent Backward этого подменю. Аналогичные комбинации клавиш: <Ctrl>+<t> и <Ctrl>+<i>.

Вы также можете выполнять эти операции сразу над несколькими выделенными графическими фрагментами. Помните только, что порядок перекрытия внутри самой выделенной группы не меняется.

**Выравнивание**

Что такое выравнивание текста, вы знаете. Но о выравнивании графических фрагментов, наверно, слышите впервые.

Однако и графику часто бывает необходимо выровнять, скажем, по верхней стороне листа или по правой стороне самого правого фрагмента в выделенной группе. Часто выполняется также центрирование выделенных фрагментов по горизонтальной или вертикальной оси.

Выравнивание графики выполняется с помощью панели Align. Чтобы вызвать ее на экран, выберите пункт \N?,!, меню Window или нажмите комбинацию клавиш <Ctrl>+<K>. Также вы можете выбрать пункт Align подменю Panels контекстного меню выделенного графического фрагмента.

**Изменение размеров**

Вы уже узнали, как можно изменить размеры графического фрагмента. Для этого достаточно указать новые значения в полях ввода W и Н, находящихся в редакторе свойств и на панели Info, и нажать клавишу <Enter>. Однако это можно сделать и намного удобнее, мышью. Для этого используется модификатор "изменение размера". Проверьте, выбран ли у вас инструмент "трансформатор", выделите на листе нужный графический фрагмент и щелкните кнопку-выключатель.

Если у вас нет на экране инструментария, и, таким образом, кнопка модификатора "изменение размеров" недоступна, вы можете воспользоваться меню. Для этого вам даже не нужно будет выбирать инструмент "транс­форматор". Просто выберите пункт-выключатель Scale подменю Transform меню Modify. Также вы можете выбрать пункт Scale контекстного меню вы­деленного фрагмента.

**Вращение и сдвиг.**

Вращение и сдвиг выделенного графического фрагмента осуществляются также с помощью инструмента "трансформатор". Однако модификатор для этого используется другой - "вращение и сдвиг". Кнопка, с помощью которой он включается, показана на рис. 9.12. Вы также можете выбрать пункт-выключатель Rotate and Skew в подменю Transform меню Modify или одноименный пункт контекстного меню выделенного фрагмента.

Выделенный фрагмент графики будет иметь при этом вид, такой же, как при выбранном модификаторе "изменение размеров". Он снова вписан в прямоугольник выделения, на углах которого расположены такие квадратные маркеры поворота, а на сторонах - маркеры сдвига.

Чтобы повернуть фрагмент, перетащите мышью один из угловых маркеров, не важно, какой. Если вы хотите повернуть фрагмент на угол, кратный 45°, то при перетаскивании маркера удерживайте нажатой клавишу <Shift>. Чтобы выполнить сдвиг, перетащите маркер, находящийся на стороне прямоугольника.

В центре прямоугольника, в который вписан ваш фрагмент, находится белая точка. Это центр вращения фрагмента, вокруг него осуществляется поворот. (Его также называют точкой фиксации.) Вы можете перетащить его в другое место, даже вынести за пределы прямоугольника. Чтобы вернуть его обратно в центр фрагмента, дважды щелкните по нему мышью.

Если вам нужно быстро повернуть выделенный фрагмент на 90°, воспользуйтесь меню. Пункт Rotate 90° CW в подменю Transform меню Modify поворачивает фрагмент на 90° по часовой стрелке, а пункт Rotate 90° CWW - против часовой стрелки. Вместо выбора этих пунктов вы можете нажимать комбинации клавиш <rtrl>+<<^hift>+<9> и <Ctrl>+<Shin>+<7; соответственно.

И, конечно же, вы можете повернуть или сдвинуть фрагмент, воспользовавшись панелью Transform. Выведите ее на экран. После этого включите переключатель Rotate и введите в расположенное справа от него поле ввода нужный угол поворота. Чтобы выполнить сдвиг, вам нужно будет включить переключатель Skew и ввести в расположенные справа от него поля ввода нужные углы сдвига: горизонтального - в левое поле ввода, вертикального - в правое. Теперь остается нажать клавишу <Enter> - и дело сделано.

**Свободная трансформация**

Если при выбранном инструменте "трансформатор" не включен ни один модификатор, то фактически все эти модификаторы включены одновременно. В этом случае говорят о так называемой свободной трансформации. При свободной трансформации вы можете, как угодно изменять выделенный графический фрагмент, не заботясь о том, какой модификатор нужно включить. Однако для того, чтобы применить к выделенному фрагменту требуемое преобразование, следует очень точно позиционировать курсор мыши над нужным маркером.

Чтобы выбрать свободную трансформацию, проще всего отключить все кнопки-выключатели, "ответственные" за модификаторы инструмента "трансформатор". Все эти кнопки находятся в области Options инструментария, Также вы можете выбрать пункт Free Transform подменю Transform меню Modify или контекстного меню выделенного фрагмента.

Чтобы выполнить то или иное преобразование, поместите курсор мыши над нужным маркером прямоугольника выделения. Курсор мыши примет при этом "говорящую" форму. Скорее всего, вам придется некоторое время подвигать курсор возле разных маркеров, прежде чем он примет нужную форму. После этого "захватите" маркер и переместите его на новое место, как и в случае других модификаторов.

По нашему мнению, свободная трансформация более пригодна для достаточно опытных художников. Начинающим же будет очень полезно потренироваться, чтобы овладеть этим мощным инструментом Flash MX.

**Глава 10. Образцы и библиотеки. Проводник Flash**

В предыдущих главах мы много рисовали, редактировали, стирали, модифицировали, преобразовывали, импортировали, в общем, не теряли времени даром. В этой главе мы рисовать не будем, Хватит, научились и рисовать, и править нарисованное, и стирать ненужное. Поговорим об управлении графикой. Да-да, именно так: будем учиться управлять нашей графикой.

А именно, преобразовывать графические фрагменты в образцы и помещать их в библиотеки. А впоследствии извлекать нужный образец из библиотеки и создавать экземпляры этого образца на рабочем листе. Ну и, конечно, управлять содержащимися в библиотеке экземплярами: править, удалять и т. п.

Но зачем? Кому нужны эти библиотеки и образцы? Зачем нам эта лишняя головная боль?

Давайте возьмем гипотетическое изображение Flash, содержащее множество мелких одинаковых элементов. Не будем придумывать, что это за изображение, просто представим себе его. Каждый его элемент описывается некоторым набором графических примитивов, а, значит, занимает место в памяти и на диске. Десять таких элементов занимает в десять раз больше места, сто элементов - в сто раз больше и т. п. Представляете, во сколько раз увеличится результирующий файл Shockwave/Flash?

Теперь давайте рассуждать так. Что представляет собой этот огромный файл? Повторяющийся набор описаний абсолютно одинаковых элементов - и больше ничего. Мы описываем много раз одно и то же! Нельзя ли нарисовать этот повторяющийся элемент один раз, а потом просто ставить в нужных местах ссылку на его описание? В этом случае мы сэкономим уйму места: ссылка на описание графического элемента занимает несравнимо меньше места, чем сам этот графический элемент. Возможно, ли такое сделать во Flash?

Конечно. Если бы не было возможно, мы бы об этом не говорили. В этом случае описание графического элемента помещается в особое хранилище, называемое библиотекой, а сам этот элемент получает название образца(в терминологии Flash - symbol).

Каждому образцу дается при создании уникальное имя, по которому его можно опознать, и определенный тип, задающий его свойства.

Теперь в нужном месте на рабочем листе мы ставим ссылку на этот образец, создавая тем самым экземпляр образца или просто экземпляр (в терминологии Flash - instance).

Такой подход имеет еще одно преимущество. Вы можете изменять все экземпляры одного образца, просто отредактировав этот образец в библиотеке.

После этого Flash сам обновит все экземпляры. Видите, как удобно? Единственный недостаток: вы не сможете изменять созданный таким образом экземпляр так же свободно, как обычную графику. Но этим во многих случаях можно пожертвовать, не так ли?

Более того - вы можете менять некоторые параметры этого экземпляра. Например, сменить его цвет, прозрачность, повернуть его, исказить или из­менить его размеры. И, тем не менее, он останется экземпляром. Если вы отредактируете образец в библиотеке, этот измененный экземпляр также обновится.

Все файлы, что вы импортировали в документ Flash, кроме изображений формата Shockwave/Flash, также оказываются в библиотеке. Все изображения и фильмы, созданные в других программах, все звуки оказываются там и превращаются в образцы. (Кроме изображений в формате Shockwave/Flash - эти помещаются прямо на рабочий лист, и вам придется вручную добавить их в библиотеку.) Это очень удобно: вы можете помещать экземпляры импортированного изображения или звука на лист сколько угодно раз, а размер файла при этом увеличиваться не будет. (Хотя, конечно, все-таки увеличится - при самом импорте.)

Flash также позволяет вам создавать разделяемые библиотеки. Такие библиотеки вы можете выкладывать в Интернет, чтобы другие пользователи Flash могли использовать их содержимое. При этом проигрыватель Flash, загрузив изображение, в котором был использован образец из разделяемой библиотеки, автоматически загружает файл библиотеки и извлекает из нее нужный образец. Таким образом, размер файла Shockwave/Flash уменьшится еще сильнее. Однако если проигрыватель Flash не сможет загрузить файл разделяемой библиотеки, изображение не будет показано правильно.

А напоследок мы поговорим о мощном инструменте организации вашей Flash-графики. Это так называемый Проводник Flash, аналогичный по назначению Проводнику Windows. С его помощью вы сможете быстро отыскивать образцы, экземпляры и прочие графические элементы, содержащиеся в вашем творении.

**Типы образцов**

Flash поддерживает непосредственное создание образцов трех различных типов. Тип образца задается при его создании и впоследствии может быть изменен. Каждый из этих типов обладает своими особыми свойствами и имеет свою особую область применения. Давайте их перечислим и опишем, дав в скобках наименование, принятое в терминологии Flash.

Графический образец (graphic) представляет собой обычное сГатичное или ани-мированное изображение, сделанное во Flash или импортированное из другой программы. Это, вероятно, наиболее часто используемый тип образцов.

Образец-кнопка (button) • - это особый образец, представляющий собой обычную командную кнопку. Такие образцы используются для создания графического пользовательского интерфейса для написанных на Flash программ и рассматриваются во всех подробностях в главе 20.

Образец-клип (movie clip) представляет собой настоящий фильм, созданный во Flash или другом программном пакете, импортированный в текущий документ Flash и помещенный на рабочий лист как часть изображения. Такой образец "работает" совершенно независимо от основного изображения, частью которого является. Образцы-клипы применяются для создания очень сложной анимации. Кроме того, по сравнению с графическими образцами, образцы-клипы предоставляют дополнительные возможности для программирования своего поведения, поэтому часто используются при создании пользовательского интерфейса. Мы изучим образцы-клипы во время рассмотрения создания анимации в среде Flash (часть 3 книги).

Flash предоставляет также возможность переопределить тип на уровне не образца, а экземпляра. Это значит, что вы можете поместить на лист экземпляр какого-либо образца и переопределить его - экземпляра - тип; тип образца при этом не изменится. Сейчас это не принесет нам пользы, но в дальнейшем, при изучении анимации и программирования (части 2 и 5), может пригодиться.

Кроме перечисленных выше трех типов, Flash поддерживает еще три: образец-растровое изображение (bitmap), образец-звук (sound) и образец-импортированный клип (embedded video). Их также собирательно называют импортированными образцами. Как вы уже поняли, образцами-растровыми изображениями становятся импортированные в документ растровые изо­бражения, а образцами-звуками - импортированные звуки. Эти два типа образцов не могут быть созданы непосредственно во Rash, в отличие от графических образцов, кнопок и образцов-клипов.

**Глава 11. Публикация и экспорт статичной графики.**

При публикации Flash не просто сохраняет изображение в другом формате. Он также выполняет весьма сильную оптимизацию, удаляя ненужные цвета, "обрезая" лишнее пустое пространство, сжимая массив точек согласно выбранному вами алгоритму. Благодаря этому графический файл, подготовленный для распространения в Интернете, по электронной почте или иным способом, имеет минимальный размер.

Кроме публикации, вы можете выполнить экспорт изображения. Экспорт отличается от публикации тем, что при создании изображения Flash не проводит никакой оптимизации. Поэтому экспортированные файлы плохо подходят для распространения графики, но зато прекрасно редактируются в других графических программах.

Давайте рассмотрим публикацию и экспорт изображений Flash.

**Выбор формата публикации**

Но, прежде всего, давайте выясним, в каком формате мы будем распространять наше творение. Правильный выбор формата публикации в данном случае решает многое. В самом деле, если вы выберете для распространения изображения какой-нибудь малоупотребительный формат, мало кто сможет его просмотреть. Поэтому нужно ориентироваться на самые популярные форматы графики.

Практически всегда неподвижные, статические изображения сохраняются для дальнейшего распространения в двух форматах: GIF и JPEG. В последнее время, в связи с событиями вокруг формата GIF, постепенно, хотя и очень медленно, набирает популярность формат PNG. (Иногда используется также формат BMP и некоторые другие, но довольно редко.)

Вы, конечно же, заметили, что все три этих формата - растровые. Растровая графика традиционно лидирует, когда дело доходит до распространения изображений. Это происходит потому, что растровая графика требует существенно меньше системных ресурсов для вывода ее на экран. Конечно, она проигрывает векторной в возможности обработки, но скажите честно, вы часто занимаетесь тем, что вращаете, искажаете и перекрашиваете полученные из Сети картины в стиле "фэнтези"? Вы их просто смотрите, не так ли?

Векторная графика, в отличие от растровой, используется практически только при создании изображений. Точнее, использовалась, пока не появился пакет Macromedia Flash и, вместе с ним, одноименный формат векторной графики, предназначенный именно для ее распространения. Можно сказать, что Flash осуществил прорыв, принес векторную графику конечному потребителю графических изображений. Но даже графика Flash имеет достаточно узкую область применения, остальные области прочно удерживает за собой растровая графика.

Так какой же формат выбрать? На каком из них остановиться?

Формат GIF, как вы уже знаете из главы 4, прекрасно подходит для сохранения и распространения штриховых рисунков с небольшим количеством цветов, таких, как схемы, карты, карандашные рисунки, элементы оформления Web-страниц и т.п. При сохранении графики в формате GIF выполняется сжатие массива данных по алгоритму, исключающему потери данных. Так что файлы GIF получаются весьма компактными.

Формат PNG - прекрасная альтернатива GIF. Он особенно хорош тем, что разработан группой независимых исследователей и абсолютно бесплатен. GIF же - увы! - стоит денег. Фирма UniSys, владеющая патентом на алгоритм сжатия, на редкость некстати вспомнила об этом. Так что если вы озабочены возможным переходом формата GIF в разряд платных продуктов, сохраняйте свою графику в файлах PNG. Тем более, что PNG предоставляет несравнимо больше "наворотов", чем старенький GIF.

**Предварительный просмотр публикуемой графики**

Flash предоставляет возможность просмотреть публикуемый файл перед собственно публикацией. Для этого выберите пункт Publish Preview в меню File. Откроется подменю, содержащее несколько пунктов, соответствующих каждому формату публикации, который вы выбрали. Если вы выберете один из этих пунктов, Flash выполнит экспорт вашего изображения в соответствующем формате и тут же откроет его в программе, зарегистрированной для этого типа файлов. Для файлов GIF, JPEG и PNG такой программой будет установленный на вашем компьютере Web-обозреватель. Когда вы закроете эту программу, созданный Flash файл будет удален.

Вы также можете быстро просмотреть результаты экспорта изображения в так называемый формат по умолчанию. Таким форматом Flash считает тот, что выбран в окне Publish Settings (см. рис. 11.1) первым, если считать сверху вниз. В нашем случае таким форматом будет GIF. Чтобы просмотреть опубликованное в таком формате по умолчанию изображение, выберите пункт Default - (<Название формата>) в подменю Publish Preview меню File или нажмите клавишу <F12>.

**Экспорт изображения**

Напомним, чем отличается публикация от экспорта. При публикации результирующее изображение максимально оптимизируется, чтобы достичь максимального качества отображения и минимального размера файла. Такое изображение может передаваться по компьютерным сетям, электронной почте, на дискетах и другими путями, выводиться на экран и распечатываться. Редактирование подобного изображения в других графических программах обычно не предусматривается. При экспорте, наоборот, изображение подготавливается именно к правке в других программах, поэтому оптимизация не проводится.

Как правило, любое изображение, созданное в любой программе векторной графики (не только Flash), требует дополнительной правки. Поэтому, вероятно, не стоит сразу публиковать его, а лучше, сначала экспортировать и подправить в другой программе.

**ЧАСТЬ III. РАБОТА С АНИМАЦИЕЙ**

**Глава 12. Форматы анимированной графики и видео.**

История компьютерного видео насчитывает всего лет десять. И это понятно: чтобы хранить полноразмерные фильмы, нужно очень много дискового пространства, а чтобы его нормально обрабатывать, нужны исключительно мощные компьютеры. Если вы смотрите фильмы в формате MPEG 4 (это могут быть как фильмы на дисках DVD, так и фильмы, записанные на обычных CD в формате DivX), то знаете, что, не имея современного ком­пьютера, о компьютерном видео лучше забыть. Иначе вы рискуете смотреть не фильм, а слайд-шоу со скоростью один кадр секунд этак в пять.

Как раз десять лет назад появились достаточно быстродействующие компьютеры и достаточно емкие жесткие диски, чтобы обрабатывать и хранить компьютерное видео. И именно в то время возникло понятие "мультимедиа" (multimedia), обозначавшее совокупность текстовых, цифровых, звуковых и видеоданных, объединенных в единое целое.

Давайте выясним, почему для хранения и обработки видео нужно больше ресурсов, чем для обычной компьютерной графики. Вероятно, вы знаете, что кино- или видеофильм можно представить как набор множества неподвижных изображений (кадров), очень быстро, несколько раз в секунду, сменяющих друг друга. Так как человеческий глаз не может уследить за сменой одного изображения другим, субъективно этот непрерывный "поток" изображений выглядит как одна движущаяся картинка. Для достижения такого эффекта частота кадров фильма должна быть довольно велика.

Каждый фильм фактически представляет собой огромное количество статичных изображений. Чтобы сохранить их в цифровом виде, нужен носитель весьма большой емкости. Поэтому для сохранения фильмов используют алгоритмы сжатия с потерями. Но здесь первопроходцев компьютерного кино подстерегает другой подводный камень: для распаковки таких фильмов и вывода изображения на экран нужны огромные вычислительные мощности. А такие мощности появились в распоряжении рядовых пользователей только десять лет назад. Да и то, тогдашние фильмы не блистали качеством.

Вероятно, самым первым форматом записи фильмов был QuickTime, разработанный фирмой Apple для компьютеров Macintosh и позднее перенесенный на Microsoft Windows. Фирма Microsoft в ответ разработала программный пакет Video for Windows и формат записи фильмов AVI, позволяющий использовать различные алгоритмы сжатия. Несколькими годами позднее, когда мощности персональных компьютеров еще возросли, независимая группа MPEG разработала очень сильный для своего времени алгоритм сжатия MPEG 1. Именно в этом формате записывались диски VideoCD - первый и не слишком удачный опыт создать цифровое кино "для народа".

Вероятно, вы видели эти диски, не могли не видеть. VideoCD - это обычный компакт-диск, на который записывался один огромный файл формата MPEG 1 с расширением dat; на одном таком диске помещалось около часа видео, поэтому полноразмерные фильмы поставлялись на двух дисках. Качество видео было очень плохим, фильмы смотрели на компьютерах с помощью специальных программ, открывая их в маленьком окошке, или на особых проигрывателях, похожих на видеомагнитофоны. Бизнес по продаже дисков VideoCD был развит вольно сильно: автор припоминает, что видел в Волгограде магазин, торгующий фирменными дисками с отечественным кино. Но все, же VideoCD - один из тех многочисленных блинов, что, будучи первыми, выходят комом.

Настоящую жизнь компьютерному кино дали появившиеся во второй половине девяностых годов лазерные диски DVD. Эти лазерные диски высокой плотности позволяют записывать до 6 гигабайт информации, чего вполне хватает не только на полноразмерный фильм высокого качества, но и на несколько звуковых дорожек, интерактивные меню и прочие "бонусы". Для записи фильмов на такие диски используется алгоритм MPEG 2.

Существуют также альтернативные форматы записи видео на обычные компакт-диски. Разработан SuperVideoCO на который помещается до двух часов видео, сжатого по алгоритму MPEG 2 создан великий и ужасный DivX альтернативная реализация алгоритма MPEG 4 - позволяющий записывать полноразмерный фильм отличного качества на один CD. Эти форматы, в основном DivX, составляют серьезную конкуренцию DVD, будущее кото­рого сегодня выглядит не так уж и безоблачно.

**Покадровая и трансформационная анимация**

Эти два способа создания движущегося изображения почти аналогичны двум разновидностям компьютерной графики Рассмотрим их подробнее.

**Покадровая анимация**

Покадровая анимация (ее еще называют классической) представляет собой набор изображений различных фаз движения - кадров, прокручиваемых с большой скоростью. Это самый старый и самый надежный способ сохранения движущегося изображения на каком-либо носителе (пленке, бумаге, магнитной ленте, жестком диске, CD). Пример покадровой анимации из пяти кадров показан на рис. 12.1.

Каждый из множества кадров, составляющих фильм, занимает при хранении определенное пространство на диске. Предположим, что это пространство составляет 100 килобайт - для хранения полноцветного изображения высокого разрешения в формате JPEG этого даже маловато. Теперь предположим, что таких изображений у нас 100 000 - такой длинный у нас фильм. Умножаем 100 на 100 000 и получаем 10 000 000, т. е. примерно 10 гигабайт. (Примерно, потому что гигабайт - это не 1 000 000 000, а 1 073 741 824 байт.) Выходит, для хранения фильма нам нужен целый жесткий диск, а уж сколько для этого понадобится компакт-дисков, просто страшно подумать. Этот фильм даже на DVD не влезет!

Что делать?

Конечно же, надо сжать фильм посильнее. И, если можно, сжать не только сами кадры, но и саму их последовательность, выбросив часть информации, без которой можно и обойтись. И, разумеется, сжать звуковое сопровождение.

Для сжатия фильмов используются специальные алгоритмы, реализующие сжатие с потерями. В этом случае при сжатии какая-то часть информации, не очень нужная при воспроизведении, действительно выбрасывается из фильма, за счет чего последний становится заметно меньше. Более того, эти алгоритмы анализируют каждый кадр фильма и сохраняют в результирующем файле только данные о разнице между соседними кадрами. Это еще сильнее уменьшает размер сжатого фильма.

Вот поэтому для сохранения покадровой анимации и применяются только растровые форматы. Если же сохранить такую анимацию в векторном формате, ее будет очень трудно, практически невозможно сжать с применение одного из обычно применяемых для этого алгоритмов. Да и процессорн ч ресурсов во время вывода на экран такая анимация будет потреблять значительно больше растровой.

Такие алгоритмы, как MPEG 4 и DivX, позволяют поместить полноразмерный фильм на обычный компакт-диск, т. е. размер сжатого с их помощью видеофайла составляет 600-720 мегабайт. Это уже вполне приемлемые параметры, учитывая объемы современных жестких дисков. Именно эти два алгоритма, как нам кажется, и совершили "компьютерно-киношную" революцию, создав высококачественное цифровое кино "для народа". То есть, сделали то, чего не удалось сделать VideoCD.

Но здесь возникает другая проблема. Сжатые сильными алгоритмами фильмы могут "осилить" только достаточно мощные компьютеры. Если вы попробуете просмотреть фильм DivX на компьютере пятилетней давности, то увидите не нормальный фильм, а слайд-шоу. Это происходит потому, что слабосильный процессор не успевает распаковывать данные и выдавать их на экран и поэтому вынужден пропускать целые кадры. К счастью, никому в голову не приходит запускать цифровое кино на старых компьютерах. А современная компьютерная техника сейчас достаточно дешева.

**Трансформационная анимация**

Как вы знаете, отдельные кадры покадровой анимации сохраняются в виде растровых изображений. В векторном виде сохранять их, вроде бы, неудобно - об этом уже говорилось. Теперь давайте предположим, что мы не вняли голосу разума и все-таки сохранили каждый кадр такой анимации в векторном виде. (Скажем честно, такая анимация, какая изображена на рис. 12.1, сама просится в векторный вид. Сами посмотрите - ведь простейшая графика, линии и заливки.) Далее, предположим, что мы можем описывать с помощью формул не только форму кривых и прочих графических примитивов, но и их поведение. Следовательно, мы можем изменить форму "рта", просто вызвав нужную формулу и подставив в нее нужные параметры. Что у нас получится?

А получится у нас трансформационная анимация. От покадровой она отличается тем, что не описывает каждый кадр последовательности отдельно, а сразу задает поведение того или иного графического примитива. А собственно созданием анимации, движения, занимается программное обеспечение, поддерживающее соответствующий формат файлов.

Как вы уже поняли, векторная графика и трансформационная анимация -братья навек. Как трансформационная анимация немыслима без векторной графики (к растровому изображению ее применить просто невозможно), так и векторной графике сам Бог велел добавить трансформационную анимацию. Однако хоть векторная графика как способ представления изображений существует довольно давно, трансформационная анимация возникла только в последние годы. Фактически трансформационную анимацию создал пакет Flash, если до него и существовали какие-то аналогичные разработки, то они остались неизвестными широкой публике.

**Форматы видеофайлов**

Теперь поговорим о самых распространенных на сегодняшний день форматах видеофайлов. А в описании каждого формата также опишем используемые для этого формата алгоритмы сжатия. Практически все эти форматы обеспечивают сохранение только покадровой анимации, кроме самого формата Shockwave/Flash и его предшественника FutureSplash.

**QuickTim**

Один из самых первых, если не самый первый, формат видеофайлов, получивший широкое распространение. Разработан фирмой Apple в конце 80-х годов, изначально предназначался для использования на компьютерах Macintosh, впоследствии был перенесен в операционную систему Microsoft Windows. Позднее подвергался неоднократным усовершенствованиям. В на­стоящее время последней версией является 5.0.

Позволяет хранить и видео-, и аудиоинформацию в одном файле с расширением mov. Для сжатия данных используется одноименный алгоритм. Степень сжатия довольно велика, но качество получающегося фильма не очень высоко по сравнению с качеством, обеспечиваемым алгоритмами группы MPEG и DivX. Поэтому формат и алгоритм QuickTime в последнее время теряет свои позиции.

За прошедшие десять лет формат QuickTime получил довольно большую популярность, в основном, для создания коротких музыкальных, рекламных или обзорных видеоклипов. Абсолютный лидер на платформе Macintosh. В Web-дизайне применяется крайне редко. Насколько известно автору, для распространения полноразмерных фильмов вообще не применяется, хотя в этом формате часто распространяются музыкальные видеоклипы.

**AVI**

Формат данных AVI (Audio and Video Interlaced - чередующиеся аудио и видео) был разработан фирмой Microsoft в начале 90-х годов для использования в мультимедийном программном пакете Microsoft Video for Windows. Для сжатия как аудио-, так и видеоинформации могли использоваться различные алгоритмы, что обеспечивает этому формату большую гибкость.

**MPEG**

Этот формат был разработан в начале 90-х годов группой MPEG (Motion Picture Encoding Group -- группа кодирования движущегося изображения) для сохранения фильмов, сжатых с использованием алгоритмов, созданных той же группой. Базируется на формате AVI, но лишен его ограничений.

Файл формата MPEG имеет расширение dat, mpg, mpe, mpeg, mpl, mp2 или mp4. Собственно, набор возможных расширений очень велик, но все они однотипны: как правило, используются буквы тир.

Как уже говорилось, группой MPEG было разработано три алгоритма для сжатия видеоданных.

**DivX**

Фактически такого формата не существует. На самом деле, это обычный файл AVI или MPEG, сжатый с использованием алгоритма DivX. Созданный в самом конце 90-х годов на основе алгоритма MPEG 4 группой "независимых программистов" (фактически, хакеров), он обеспечивает еще большую степень сжатия при несколько более низком качестве изображения.

Видеофайлы, сжатые с использованием старых версий этого алгоритма, записывались в формате AVI. Версия 5, появившаяся совсем недавно, также поддерживает формат MPEG. Иногда файлы формата AVI, сжатые этим ал­горитмом, имеют расширение divx.

Формат DivX используется, в основном, для распространения полнометражных фильмов на обычных компакт-дисках (для этого он, собственно, и создавался). Очень часто такие диски содержат пиратские копии фильмов, из-за этого формат DivX считается чуть ли не незаконным, "хакерским", хотя в самом его существовании нет ничего противозаконного. Совсем недавно формат DivX получил признание крупных разработчиков игр, распространивших в этом формате демонстрационные ролики. В Web-дизайне формат DivX не применяется.

**WMV**

Дальнейшее развитие формата AVI. Разработан фирмой Microsoft в самом конце 90-х годов как часть инициативы Windows Media (создание набора аппаратных и программных средств, направленных на улучшение мультимедийных возможностей современных Windows-совместимых компьютеров). Само название этого формата переводится как Windows Media - Video.

Видеофайлы, сохраненные в этом формате, имеют расширение wmv или asf. (Существует особая разновидность формата - WMA (Windows Media -Audio), предназначенная для хранения звука, такие файлы имеют расшире­ние wma.) По сравнению с AVI формат WMV имеет различные дополни­тельные возможности, в частности, средства защиты от несанкционирован­ного копирования.

В настоящее время этот формат стремительно набирает популярность. Но используется он опять же для распространения полноразмерных фильмов и видеоклипов.

**RealMedia**

Разработан фирмой RealNetwork в середине 90-х для распространения видео через Интернет. Вероятно, первый видеоформат, ориентированный на использование в Сети. Подвергался неоднократным усовершенствованиям. В настоящее время последней версией является 8.0.

Позволяет хранить и видео-, и аудиоинформацию в одном файле с расширением U. Для сжатия данных используется одноименный алгоритм. Степень сжатия весьма велика, и достигаемое при сжатии качество фильма также достаточно высоко.

В настоящее время удерживает довольно большую долю рынка "сетевого видео" и терять ее не собирается. В основном, применяется для распространения фильмов различной длительности и назначения и трансляции так называемого "интернет-телевидения". В Web-дизайне применяется крайне редко.

**Глава 13. Покадровая анимация.**

Flash - пакет, предназначенный, прежде всего, для создания именно анимации. Непод­вижные изображения, конечно, с его помощью создавать тоже можно, но профессиональные художники используют его только тогда, когда под рукой нет ничего более подходящего. Для создания неподвижной графики существуют Adobe PhotoShop, Adobe Illustrator, Macromedia Freehand, Macromedia Fireworks и другие программы. Они приспособлены для этого лучше, чем Flash. Имейте это в виду.

Главное преимущество Flash в том, что он предлагает непревзойденное со­четание достаточно богатых возможностей создания неподвижных изобра­жений, анимации и интерактивности. В этом смысле все вышеперечислен­ные программы просто отдыхают. При всех их немалых достоинствах никто из них не может того, что может Flash.

Если кто-то скажет, что это не так, напомните ему статистические данные. То, что проигрыватель Flash установлен на 95% всех компьютеров мира, -правда.

Практическая реализация покадровой анимации очень проста и наглядна, хоть и весьма трудоемка. Таким образом, мы сможем больше узнать о временной шкале, с которой часто будем работать в дальнейшем. Вы ведь помните важнейший принцип познания: от простого - к сложному. Трансформационная анимация, конечно, значительно менее трудоемка, но овладеть ей сложнее, чем покадровой.

Напомним, что представляет собой покадровая анимация. Это последовательность большого количества однотипных кадров, изображающих различные фазы движения. Такая последовательность очень быстро прокручивается перед зрителем, в результате чего он видит непрерывное движение. Как видите, все достаточно просто. Остается добавить, что на принципе покадровой анимации основаны весь кинематограф и все телевидение. Так что с покадровой анимацией вы сталкиваетесь, когда смотрите вечерние новости.

**Глава 14. Трансформационная анимация.**

Как вы узнали из главы 13, покадровая анимация создается очень просто. Нужно только добавить достаточное количество кадров в последовательность, а дальше - рисуй себе их содержимое, используя средства, изученные в части 2. Только вот беда: рисовать придется очень долго...

Но, если вы не хотите рисовать слишком долго, вам на помощь придут средства Flash по созданию трансформационной анимации. Если Flash используется для подготовки Web-графики (элементы оформления Web-страниц, анимированные рекламные байкеры, небольшие фильмы), то зачастую вся анимация ограничивается простейшими движениями какого-либо элемента. А такие движения элементарно реализуются с помощью трансформационной анимации (так называемая трансформация движения). Вам нужно будет только нарисовать два ключевых кадра последовательности - начальный и конечный - и сделать несколько щелчков мышью. Конечно, если ваша анимация сложнее, чем простое движение "из точки А в точку В", вам придется также создать дополнительные ключевые кадры, во всех точках, где движение элемента изменяется.

Более того, с помощью Flash вы можете заставить какой-либо элемент менять свой цвет или прозрачность, последнее нужно, если вы хотите сделать элемент постепенно появляющимся или исчезающим. А еще средствами Flash можно реализовать морфинг - плавное изменение формы любого эле­мента. Этот вид анимации называется трансформацией формы.

Так что средства трансформационной анимации Flash не так уж и бедны, как вы могли подумать, читая главу 12, где подробно описывались оба вида анимации.

Кроме меньшей трудоемкости, трансформационная анимация имеет еще одно достоинство перед покадровой: она занимает меньше места. В самом деле, вместо того, чтобы хранить все кадры фильма, программа записывает в файл только несколько чисел - параметры функции, реализующей эту анимацию. Сравните объем десятка изображений, пусть даже и векторных, и объем, занимаемый этими числами. Даем гарантию - это сравнение будет отнюдь не в пользу покадровой анимации.

Основным недостатком трансформационной анимации является ее бед­ность. Хоть Flash и предлагает нам достаточно мощные средства "оживить" изображения "малой кровью", красивые и сложные мультфильмы вы из одних трансформаций не сделаете. Если вы мечтаете о карьере в художественной анимации, то - увы!.. - готовьтесь опять же рисовать бесчисленные кадры, один за другим, один за другим...

**Глава 15. Слои**

И поговорим о слоях. Поскольку без слоев более-менее сложной трансформационной ани­мации все равно не создашь. Да и при создании покадровой анимации слои могут стать серьезным подспорьем. Если, конечно, научиться ими пользо­ваться...

Слой (в терминологии Flash - layer) можно рассматривать как лист прозрачной пленки, лежащий на рабочем листе Flash. Вы можете рисовать на слое, используя изученные в главе 5 инструменты рисования. Кроме того, вы можете класть поверх этого слоя или под ним другие слои, на которых тоже что-то нарисовано. И, наконец, можно легко переключаться между слоями, чтобы работать с нарисованной на них графикой. Каждый слой имеет уникальное имя, с помощью которого он однозначно идентифицируется.

Выясним теперь преимущества, даваемые слоями.

□ В главе 5 вы познакомились с фрагментацией и слиянием графических фрагментов. В качестве борьбы с этим явлением предлагались группировка, преобразование фрагмента графики в образец и "разнесение" фрагментов по разным слоям. В самом деле, если какие-либо графические фрагменты находятся в разных слоях, то они не будут ни фрагментироваться, ни сливаться. А все потому, что они не соприкасаются друг с другом.

□ Вы уже знаете, что методом трансформации движения можно анимировать только один графический элемент в слое. Но если слоев может быть сколько угодно, то и количество анимированных элементов тоже не ограничено. Каждый анимированный элемент находится в своем слое, не мешая, таким образом, другим анимипоняиным элементам. Единственный недостаток такого подхода: в сложных фильмах слоев может быть очень много.

□ Создание некоторых специальных эффектов, таких, как слои-направляющие и маскирующие слои.

Еще нужно сказать, что слои сами по себе не увеличивают размер файла изображения Flash. Увеличивает его размер только графика, расположенная в этих слоях.

Но мы также знаем, что трансформацию движения можно применить над единственным элементом в слое. Выходит, что нам для достижения своей цели нужно "разбросать" все буквы надписи Flash по разным слоям! Но как это сделать? Неужели придется создавать все пять слоев, рисовать на них буквы и следить, чтобы они находились на своих местах? Нет ли другого способа сделать это?

Есть. Flash содержит встроенные средства для распределения выделенных графических элементов по разным слоям. Таким образом, если вы выберете несколько графических элементов, Flash создаст нужное количество новых слоев и поместит в них эти элементы. Распределенные по слоям элементы пропадут с изначального слоя, где они находились до распределения, остальное же содержимое этого слоя останется нетронутым.

**Управление слоями**

Вы можете добавить новый слой, нажав кнопку Insert Layer, расположенную в нижней части списка слоев. Вы также можете выбрать пункт Layer в меню Insert или пункт Insert Layer в контекстном меню выделенного слоя. Новый слой будет вставлен в список сразу же над выделенным слоем.

Вы можете перемещать слои в списке, меняя порядок их перекрытия друг другом. Для этого просто перетащите нужный слой мышью на новое место.

Есть три способа выделить в списке нужный слой. Во-первых, вы можете щелкнуть по нему мышью в списке слоев, после этого все содержимое этого слоя на рабочем листе будет выделено. Во-вторых, вы можете щелкнуть по любому кадру анимации, созданной в этом слое. В-третьих, вы можете просто выделить на рабочем столе один из графических фрагментов, находящихся в этом слое. (Помните: все слои прозрачны, поэтому сквозь пустое пространство слоев вы можете видеть все, что находится под ними.)

Вы можете также выделять сразу несколько слоев в списке. Если вам нужно выделить непрерывную группу слоев, щелкните по первому слою в группе, нажмите клавишу <Shift> и, удерживая ее, щелкните по последнему слою в группе. Если же вам нужно выделить несколько несвязанных слоев, сначала щелкните по первому, а потом продолжайте щелкать по остальным, удерживая нажатой клавишу <Ctrl>. Как видите, здесь работает та же методика, что и в Проводнике Windows.

Слои можно переименовывать. Для этого дважды щелкните по имени нужного слоя. После этого вместо его имени появится небольшое поле ввода, в котором будет поставлено старое имя слоя. Введите новое имя и нажмите клавишу <Enter>. Если же вы передумали менять имя слоя, нажмите клавишу <Esc>.

Удалить ненужный слой вы можете разными способами. Проще всего выделить слой, который вы хотите удалить, и нажать кнопку Delete Layer (рис. 15.9), расположенную в нижней части списка слоев. Также вы можете перетащить ненужный слой прямо на эту кнопку. Ну и, наконец, вы можете выбрать пункт Delete Layer в контекстном меню выделенного слоя. Учтите только, что при удалении слоя удаляется также вся расположенная в нем графика.

**Слои-направляющие**

При создании анимации очень часто бывает нужно, чтобы какой-либо элемент двигался по некоему пути. Таким путем может быть прямая, кривая или ломаная линия, окружность или сложный контур. Обычными средствами, которые мы изучили в главе 14, направить анимированный элемент по пути невозможно. Для этого нужно использовать специальный слой, называемый слоем-направляющей.

**Маскирующие слои**

Вероятно, вам встречались изображения Flash, по которым "ползает" своего рода "прожектор" - круглое светлое пятно, "высвечивающее" изображение частями. Такие изображения встречаются довольно часто: иногда это карты звездного неба, иногда - карты земной поверхности, а иногда - какие-то "картины реальной жизни". Такое светлое пятно, "высвечивающее" в один момент времени только часть какого-либо лежащего под ним изображения, совпадающую с его размерами, называется маской, слой, на котором оно находится, - маскирующим слоем, а "высвечиваемое" маской изображение - маскируемым изображением. В случае Flash маскируемое изображение находится в маскируемом слое.

Вы можете думать, что маска - это отверстие в непрозрачном маскирующем слое. (Обычно слои, как вы знаете, прозрачны.) Сквозь это отверстие можно видеть все, что лежит под этим слоем. Отверстие может быть любым графическим фрагментом: геометрической фигурой, экземпляром (только не кнопкой) или текстовым блоком. Наконец, это отверстие может быть ани-мировано методами трансформационной анимации.

**Глава 16. Импорт анимации и видео**

Об импорте графики мы уже говорили. Также упоминали и о том, что для каждой задачи существует свой идеальный инструмент. Ну и, конечно, мы достаточно подробно разъяснили, какие задачи решает Macromedia Flash. Все это было описано в главе 8, где рассказывалось об импорте статичной графики, и, пожалуй, повторять все это здесь - значит зря расходовать бумагу.

И все же мы кое-что скажем. Как-никак, статичная графика и анимация - вещи очень разные, а во многом - диаметрально противоположные.

Прежде всего, нужно сказать, что Flash позволяет как внедрять импортированное видео в файл документа Flash, так и создавать ссылку на внешний видеофайл. В первом случае говорят о внедренном в документ Flash клипе, а во втором - о клипе, связанном с документом Flash. Маленькие видеофайлы, как правило, внедряются в документ Flash, а большие - связываются с ним. К несчастью, поддерживается только связывание с файлами формата QuickTime.

Теперь поговорим о собственно импорте внешнего видео. И укажем, какие файлы лучше внедрять, а какие - связывать.

Чаще всего приходится импортировать во Flash фильмы, созданные в самом Flash, но другими художниками. Например, вы можете "позаимствовать" удачные анимированные элементы для своего изображения или фигурные кнопки для своей Flash-программы. (О программировании во Flash см. часть 4 этой книги.) Конечно, эти элементы могут быть помещены в обычные или разделяемые библиотеки Flash, но зачастую они просто "лежат" в Сети "россыпью", в виде отдельных SWF-файлов. Поскольку такие файлы имеют небольшой размер, они внедряются в документ Flash, это позволяет сократить до минимума количество файлов, необходимых для показа изображения.

Также Flash поддерживает импорт "анимации", отдельные кадры которой хранятся в пронумерованных файлах. Такие наборы изображений могут быть созданы разными графическими программами, в частности, самим Flash. Подобные файлы также внедряются в документ Flash.

Видеоклипы, сохраненные в других форматах: QuickTime, MPEG и др. (все поддерживаемые Flash видеоформаты описаны в главе 12), внедряют во Flash значительно реже. Причины этому чисто технические, и заключаются они в том, что, как правило, в таких форматах распространяются полнометражные игровые, документальные и музыкальные фильмы. А файлы, содержащие такие фильмы, занимают очень большой объем, что не может не сказаться на размерах результирующего файла Flash, а значит, на времени его загрузки. Поэтому, такие файлы практически всегда связывают с документом Flash. В результате, внешняя анимация загружается только тогда, когда в ней возникнет надобность.

При импортировании во Flash внешних файлов возникает еще один вопрос - о защите авторских прав их создателей (если, конечно, вы не создали эти файлы сами). В самом деле, воровать нехорошо. Поэтому прежде, чем позаимствовать что-либо чужое для своего блага, обязательно выясните, как к этому отнесется владелец. На Web-сайтах, занимающихся распространением графики, обязательно указывается, что с этой графикой разрешено делать. Поэтому будьте внимательны: всегда читайте то, что написано мелким шрифтом.

Как вы уже знаете, Flash поддерживает импорт "анимации", отдельные кадры которой хранятся в пронумерованных файлах. Если вы импортируете файл растрового изображения (формата BMP, GIF или любого другого), чье имя кончается числом, Flash ищет аналогичные ему файлы. (Подробнее об импорте статичной графики см. главу S.) Если он находит последователь­ность таких пронумерованных файлов, например, Рис1.Ьтр, Рис2.Ьтр, РисЗ.Ьтр и т. д., то предлагает импортировать всю эту последовательность как видеоклип. На экране появляется соответствующее предупреждение, нажмите кнопку Yes (Да) или No (Нет). Если вы нажмете Yes, Flash импортирует последовательность файлов и превратит их в клип, если же вы нажмете No, будет импортирован только выбранный вами файл.

Если вы хотите связать изображение Flash с внешним файлом, учтите что в данной версии Flash (MX) поддерживается только связывание с файлами QuickTime. Если же вы хотите связать изображение Flash с видеофайлом другого формата, то нужно преобразовать последний в файл QuickTime, воспользовавшись программным пакетом Apple QuickTime.

**Импорт видео и работа с ним**

А теперь давайте рассмотрим, как же в изображение Flash импортируется видео, и какие средства предоставляются для работы с ним.

Как вы знаете, все импортированные во Flash видеоклипы преобразуются в образцы-импортированные видео. Так что вы можете создавать на рабочем листе столько экземпляров импортированного клипа, сколько хотите, и размер файла Flash при этом увеличиваться не будет. (Хотя, конечно, при самом импорте клипа он увеличится, и зачастую очень сильно.)

**Внедрение видео**

Для того чтобы внедрить в документ Flash внешний видеофайл, поместив его и в библиотеку, и на рабочий лист, выберите пункт Import в меню File или нажмите комбинацию клавиш <Ctrl>+<R>. На экране появится стандартное диалоговое окно открытия файла Windows. Найдите нужный файл и нажмите кнопку открытия файла этого диалогового окна.

Когда вы хотите импортировать видеоклип только в библиотеку, выберите пункт Import to Library в меню File. На экране появится стандартное диалоговое окно открытия файла Windows. Найдите нужный файл и нажмите кнопку открытия файла.

Если вы собираетесь импортировать файл QuickTime, на экране появится диалоговое окно Import Video (рис. 16.1). В нем находится всего два переключателя. Переключатель Embed video in Macromedia Flash document позволяет вам внедрить видеофайл в документ Flash. А переключатель Link to external video File обеспечивает связывание документа Flash с этим видеофайлом. Поскольку нам нужно внедрить видеофайл в документ Flash, следует включить переключатель Embed video in Macromedia Flash document. И, разумеется, нажать кнопку ОК. Если вы нажмете кнопку Cancel, файл не будет импортирован.

**Глава 17. Работа со звуком**

Когда-то кино было "великим немым". Люди приходили в кинотеатры, чтобы смотреть на крошечном экране черно-белые дергающиеся картинки под музыку разболтанного пианино. Все это было в новинку, как и первые фанерные самолеты, и первые телефоны, в которые нужно было кричать во все горло, как и четыре параллельные линии, показанные на экране первого телевизора. Все это было чудом, недоступным пониманию широких масс, а оттого - вдвойне притягательным.

Сейчас кино перестало быть чудом, также как воздушные сообщения, телефония и телевидение. Чудо превратилось в бизнес, а что может быть прозаичнее. Люди разговаривают по мобильникам, летают на самолетах, смотрят дома телевизоры, а кино да что о нем говорить - кинотеатры практически умирают. Никто не хочет ходить в кино: его заменило домашнее видео, VideoCD, DVD и DivX. (Говорят, в столицах ситуация получше, чем в провинции, но кто его знает.) Настанет день, когда чудо умрет окончательно, уступив место чему?

Единственное, что никогда не умрет, - это магия кинопленки. Магия черно-белых прыгающих лиц тех, кого давно с нами нет...

Когда-то компьютеры были большими и медленными, а перерабатывать умели только одни перфокарты или перфоленты в другие такие же перфокарты или перфоленты. Никто и в мыслях не мог допустить, что через несколько десятков лет их далекие потомки станут игрушками для пресыщенных обывателей и их детишек. Как же - такая большая и сложная техника финансировалась военными, очередь была расписана на недели вперед, какие еще игрушки?!!

Но прошли годы, и компьютеры действительно стали игрушками. В самом деле, если в доме есть компьютер, то обязательно найдутся и CD с игрушками и фильмами (если, конечно, компьютер достаточно мощный, а владелец - не законченный трудоголик). Более того, зачастую этот компьютер используется только для игр и просмотра фильмов. Конечно, владелец пытается заняться чем-то полезным на своем "железном друге", но эти желания так и не претворяются в действительность.

Когда-то Интернет был сплошь текстовым, разбавленным редкими вкраплениями корявых рисунков. В те времена Всемирной Сетью пользовались, в основном, только ученые, военные и государственные чиновники, а что разве нужны этим сухарям дизайнерские изыски? Первым интернетчикам вполне хватало текстовой информации и файлов, никто из них и в страшном сне не мог предвидеть, для чего будет использоваться Сеть через десять лет. Как же можно во время работы качать музыку или смотреть интерактивное видео, когда надо же, елки-палки, работать - за работу казна деньги платит, а не за музыку и видео!

Теперь да что рассказывать об этом - сами зайдите в Сеть или хотя бы вспомните, зачем вы заходили туда последний раз! Даже если у вас и было в мыслях найти что-то действительно для дела, вы точно не забывали и о развлечениях. А какие развлечения сейчас наиболее доступны рядовому пользователю Сети? Конечно, музыка. (К сожалению, видео могут смотреть только обладатели достаточно быстрых каналов, которых пока немного. Но со временем предпочтения большинства могут измениться.)

История повторяется. И кино, и компьютеры, и Интернет из "великих немых" превратились в "великих болтунов".

**Кодирование и хранение звуковых данных**

Для кодирования звука и записи его в файл применяются три различных способа. Все эти способы мы сейчас кратко рассмотрим.

Первый способ - кодирование и запись в файл собственно звука. В этом случае к компьютеру подключается микрофон или иной источник звука, и сигнал с него с помощью специальных программ записывается в файл. Такой способ записи звука применяется в подавляющем большинстве случаев. Например, популярные форматы WAV, MP3, VQF, WMA - все это форматы хранения звуковых данных.

Преимуществом такого подхода к хранению звука является точное его воспроизведение на любом устройстве. Например, вы можете переписать з формат МРЗ любимые кассеты, и ваша запись будет нормально проиграна на любом компьютере или носимом МРЗ -проигрывателе, хотя, конечно, с разным качеством. Также достоинствами являются возможность записи любых звуков (голоса, природных и искусственных шумов, любых музыкальных инструментов) и легкость обработки (подавления шумов, нормализации и др.). Недостаток, пожалуй, один: большой объем получающегося в результате записи звука массива данных.

Чтобы уменьшить массив звуковых данных, применяют различные алгоритмы сжатия. Это может быть сжатие как без потерь, так и с потерями. Как правило, без потерь сжимают звук, предназначенный для дальнейшей обработки, а с потерями - предназначенный для распространения (например, известный формат МРЗ). Иногда звук не сжимают вообще.

Кодирование и запись звука - самый распространенный способ хранения звуковых данных. В частности, он используется для распространения музыки на компакт-дисках и в файлах. На компакт-дисках записывается несжатый звук, для распространения в файлах применяются различные алгоритмы сжатия. Также этот способ записи звука служит для создания звукового сопровождения к видеоклипам.

Второй способ - хранение в файлах только команд для воспроизведения звука. При проигрывании таких файлов эти команды читаются и выполняются либо специальной программой-проигрывателем, либо соответствующим образом оснащенной звуковой картой компьютера. Для создания таких файлов используются специальные программы или музыкальные синтезаторы, опять же в совокупности со специальными программами.

Для воспроизведения собственно звука проигрыватели файлов и звуковые карты имеют примеры звучания реальных инструментов или различных реальных или синтезированных звуков, называемые инструментами или сэмплами. Для удобства загрузки и управления инструменты объединяются в банки инструментов. Некоторые очень старые программы и звуковые карты применяют для вывода звука метод частотного синтеза, но качество такого звука очень плохое.

Достоинством второго способа является компактность получающихся файлов. В самом деле, чтобы записать пару байт, составляющих команду, много места не нужно. Недостатки: невозможность записи любых звуков (только тех, для которых в используемом банке инструментов есть соответствующие сэмплы) и зависимость воспроизводимого звука от используемой программы, звуковой карты и банка инструментов.

Второй способ используется практически для тех же целей, что и первый. С его помощью можно записывать и распространять музыку, хранить репетиционные записи (они занимают немного места), помещать музыкальное сопровождение в различные программы и на Web-сайты. Самые распространенные форматы записи звука вторым способом: MIDI и WАV.

Третий способ - небольшая модификация второго. А именно: в файле сохраняются не только команды для воспроизведения звука, но и используемые для этого сэмплы. Фактически каждый такой файл содержит в себе свой собственный банк инструментов. Такой формат записи данных часто называют гибридным.

Третий способ объединяет достоинства и недостатки первого и второго способов. Файлы, созданные третьим способом, адекватно воспроизводятся на любом оборудовании и любой программой, они имеют размер, меньший чем файлы, созданные первым способом, но больший, чем файлы звуковых команд.

Как ни странно, но третий способ записи звука не снискал особой популярности. В настоящий момент его используют только особые "малобюджетные" программы создания музыки, называемые трекерами. Файлы, создаваемые с помощью трекеров, называются трекерными модулями. Существует целая подпольная трекерная субкультура, живущая своей особой жизнью, но почти не известная "в миру". Часто, говоря о форматах записи звуки и программах для создания музыки, трекеры даже не упоминают.

**WAV**

Формат WAV (от английского wave - волна) был создан фирмой Microsoft в качестве стандартного формата хранения звуковых данных в операционной системе Windows. К настоящему времени получил широчайшее распростра­нение для записи звука, предназначенного для последующей обработки, для хранения сэмплов, помещения звуковых данных в различные программы, на Web-сайты и т. п. Поддерживается абсолютно всеми программами записи и обработки звука, за исключением самых специализированных.

Файл формата WAV имеет "говорящее" расширение wav. Звуковые данные могут быть сжаты с использованием любого алгоритма, для которого в системе установлен кодек, в том числе, популярным алгоритмом MPEG 1 уровень 3, или вообще не сжаты.

Судя по всему, формат WAV будет применяться и в дальнейшем. Пока что никаких причин отказываться от него нет. И все же, для распространения музыки он применяется очень редко.

**МРЗ**

Формат МРЗ был создан для распространения музыкальных файлов, сжатых по алгоритму MPEG 1 уровень 3. В настоящее время, похоже, стал самым популярным форматом звуковых файлов в Интернете. Поддерживается абсолютно всеми профаммами записи и обработки звука, за исключением некоторых узкоспециализированных.

Файл этого формата имеет "говорящее" расширение трЗ. Звуковые данные всегда сжимаются с помощью алгоритма MPEG 1 уровень 3. другие алгоритмы не допускаются. Файл МРЗ также может хранить дополнительные данные, называемые тегами. В частности, с помощью тегов сохраняются имя исполнителя, название композиции, название альбома, год записи альбома и жанр музыки.

Этому формату, похоже, уготована долгая жизнь. Несмотря на то, что многие фирмы-распространители музыкальной продукции настроены против него, МРЗ полностью завоевал Интернет и сдавать завоеванные позиции не собирается. Более того, этот формат не собирается уступать и более новым форматам записи звука (VQF, OGG Vorbis, MS WMA и др.), хотя более новые разработки обеспечивают лучшее качество звука при значительно меньшем размере файла.

**Форматы звука, не поддерживаемые Flash**

Кроме звуковых форматов, поддерживаемых Flash, мы также рассмотрим форматы, которые им не поддерживаются. Это пригодится вам в дальнейшей работе: встретив файл, записанный в одном из таких форматов, вы будете знать, что с ним делать. А именно, перекодировать его в один из поддерживаемых форматов, использовав специальное программное обеспечение.

**WMA**

Разработан фирмой Microsoft в самом конце 90-х годов как часть инициативы Windows Media (создание набора аппаратных и программных средств, направленных на улучшение мультимедийных возможностей современных Windows-совместимых компьютеров). Само название этого формата переводится как Windows Media - Audio. Изначально формат WMA предназначен для постепенной смены форматов WAV и, в идеале, МРЗ, по сравнению с обоими этими форматами WMA обеспечивает лучшее качество звука и меньший размер файла.

Аудиофайлы, сохраненные в этом формате, имеют расширение wma. Существует также особая разновидность п„ формата - WMV (Windows Media Video), предназначенная для хранения видео, такие файлы имеют расширение v,ir . или asf. По сравнению с WAV и МРЗ формат WMA имеет различные дополнительные возможности, в частности, средства защиты от несанкционированного копирования.

В настоящее время этот формат стремительно набирает популярность. Но серьезно поколебать позиции формата МРЗ ему пока не удается. Хотя в будущем, благодаря поддержке крупнейших фирм-распространителей музыки, он вполне может вытеснить МРЗ.

**RealMedia**

Разработан фирмой RealNetwork в середине 90-х для распространения звуковой информации через Интернет. Также часто встречается название RealAudio. Подвергался неоднократным усовершенствованиям. В настоящее время последней версией является 8.0.

Файл формата RealMedia имеет расширение га. (Существует также разновидность этого формата для хранения видео, файлы, записанные в этом формате, имеют расширение .:. или smil.) Для сжатия данных используется одноименный алгоритм. Степень сжатия весьма велика, и достигаемое при сжатии качество звука также достаточно высоко.

В настоящее время удерживает довольно большую долю рынка "сетевого аудио" и терять ее не собирается. В основном, применяется для распространения музыки и трансляции так называемого "интернет-радио".

**MIDI**

Собственно, это не только и не столько формат музыкальных файлов, сколько стандарт обмена информацией между различным музыкальным оборудованием и программным обеспечением. Был разработан в начале 80-х годов, когда получили распространение цифровые музыкальные синтезато­ры, в виде стандарта General MIDI. Используется до сих пор, наряду с раз­личными "расширенными" форматами (MIDI XG и др.).

Файл этого формата имеет расширение mid, midi или nni. Он содержит набор команд воспроизведения звука для программного или аппаратного проигрывателя MIDI (то есть, имеет место второй способ записи звуковых файлов). Из-за этого MIDI-файлы имеют очень маленький размер.

В настоящее время удерживает относительно небольшую нишу - распространение музыки по Сети и создание фонового музыкального сопровождения для Web-сайтов.

**Импорт звука**

Для того чтобы импортировать в документ Flash звуковой файл, выберите пункт Import в меню File или нажмите комбинацию клавиш <Ctrl>+<R>. Вы также можете выбрать пункт Import to Library в меню l'ik> В любом случае, на экране появится стандартное диалоговое окно открытия файла Windows. Найдите нужный файл и нажмите кнопку открытия файла этого диалогового окна. После довольно продолжительного процесса импорта, в течение которого Flash выведет на экран индикатор прогресса, файл будет импортирован.

Импортированные звуковые файлы помещаются в библиотеку в виде образцов-звуков. Таким образом, вы можете создавать сколько угодно экземпляров этого звука, и размер файла Flash при этом не увеличится.

Вместо того чтобы искать нужные звуковые файлы или создавать их самим, вы можете воспользоваться библиотекой общего использования Sounds, поставляемой с Flash. Просто выберите пункт Sounds в подменю Common Libraries меню Window - и в появившемся на экране окне библиотеки вы найдете достаточное количество звуковых образцов для своих первых фильмов.

**Использование звука в фильме**

Импортированный звук используется в фильме точно так же, как и любые другие образцы, содержащиеся в библиотеке. Вы создаете для звука отдельный слой, помещаете в него экземпляр нужного образца-звука, создаете нужной длины последовательность кадров - и все! Возможно также, вы захотите задать для отдельных кадров этой последовательности какие-то эффекты: запустить проигрывание звука с этого кадра или, наоборот, остановить его.

Создание нового слоя мы описывать не будем - об этом достаточно подробно было сказано в главе 15. Желательно, чтобы в рабочем слое не было никакой графики. Вы, конечно, можете поместить несколько экземпляров звука в один слой, но это не рекомендуется. Будет лучше, если вы поместите каждый звук в своем слое: тогда у вас будет больше возможностей по управлению используемыми в фильме звуками.

Поместите в слой экземпляр образца-звука. Создайте последовательность кадров нужной длины. Для этого выделите пустое место на временной шкале, где должна заканчиваться ваша последовательность кадров, и выберите пункт Frame в меню Insert, пункт Insert Frame в контекстном меню выделенного пустого места или просто нажмите клавишу <F5>.

**Правка звука средствами Flash**

Выше мы упомянули, что, выбрав в списке Effect пункт Custom, вы можете задать некоторые параметры звука, в частности то, как будут изменяться его громкость и панорамирование во время проигрывания. Вы также можете сделать это, нажав кнопку Edit, расположенную правее списка Effect. А собственно правка звука выполняется в диалоговом окне Edit Envelope, которое появится на экране после любого из этих действий.

Правка выполняется с помощью двух больших графиков, расположенных соответственно в верхней и нижней частях этого диалогового окна. Верхний график позволяет задать изменение громкости, а нижний "отвечает" за панорамирование. Кроме того, в левом и правом нижнем углах этого окна находятся наборы небольших кнопок, которые весьма помогут вам в работе.

**Глава 18. Создание интерактивных фильмов**

Подавляющее большинство интерактивных элементов в фильмах Flash создаются с помощью клипов, точнее, экземпляров образцов-клипов. (Подробнее о них см. главу 10.) Встроенные клипы могут управляться из сценариев, в отличие от графических экземпляров и обычной графики, ко­торая не имеет соответствующих "рычагов управления". Кроме клипов, управляться из сценариев могут кнопки, и, разумеется, сам фильм.

За доступ к клипу из сценария "отвечает" объект movieClip. Данный объект обеспечивает набор свойств и методов, которые и являются "рычагами управления" клипом.

В отличие от встроенных и пользовательских объектов, экземпляр объекта movieClip не нужно создавать с помощью конструктора. Это выполняет сам

Flash. Также он создает экземпляр объекта movieciip, "отвечающий" за основной фильм (то есть сам документ Flash). Вам остается только использовать нужные свойства и методы для управления соответствующим клипом (или фильмом).

Оператор typeof для клипа возвращает значение "clip".

Зачем нужны встроенные клипы

Встроенные клипы могут использоваться разными путями и для разных целей. Сейчас мы их перечислим.

Во-первых, встроенные клипы применяются для создания сложной трансформационной анимации. Как вы помните, Flash может создать только одно движение в слое. Если же вы хотите создать несколько движений или одно сложное движение, вам придется обращаться к покадровой анимации, либо использовать трансформацию формы или встроенные клипы.

Поясним это на примере. Допустим, вы хотите создать фильм, который бы показывал автомобиль, перемещающийся справа налево по рабочему листу. Причем, его колеса должны вращаться. Обычная трансформация движения, однако, реализует только одно движение в слое: либо перемещение самого автомобиля, либо вращение только одного колеса. Что делать?

Можно, конечно, сделать "классическую" покадровую анимацию, рисуя вручную каждый кадр, но это сложно. Можно создать трансформацию формы, но это еще сложнее. Поэтому лучше всего использовать вложенную анимацию. Порядок действий в этом случае такой:

1. Создаем клип, представляющий собой вращающееся колесо. Экспортируем его в формат Shockwave/Flash.
2. Создаем основной фильм и импортируем в него клип, изображающий вращающееся колесо.
3. В основном фильме создаем графический образец или образец-клип, изображающий наш автомобиль. В него на соответствующие места по­мещаем два экземпляра импортированного клипа-колеса.
4. Помещаем на рабочий лист готовый экземпляр образца-автомобиля. Анимируем его, т. е. заставляем двигаться по листу. Дело сделано.

Как видите, мы создали довольно сложную анимацию средствами трансформации движения.

Во-вторых, встроенные клипы применяются для создания сложной анимации, управляемой сценариями. Только клипы имеют соответствующие "рычаги управления", с помощью которых вы можете воздействовать на их поведение из сценариев. (Об этом, собственно, уже говорилось, но стоит повторить еще раз.)

В-третьих, вы можете использовать встроенные клипы для создания целых Web-сайтов, размещенных во Всемирной Сети или на локальном диске. В самом деле, фильмы Flash можно рассматривать как достойную замену негибкому и порядком устаревшему языку HTML. И такие сайты уже существуют.

Ну, и, в-четвертых, клипы можно использовать для создания элементов управления Flash-приложений.

**Глава 19. Создание приложений Flash**

Знаете ли вы, что отличает простой интерактивный фильм от полноценного приложения Flash? Правильно, наличие пользовательского интерфейса!

Пользовательским интерфейсом называется совокупность средств, предназначенных для организации общения пользователя и программы. В случае хорошо нам знакомой операционной системы Windows, пользовательский интерфейс - это набор окон и элементов управления, которые служат для ввода данных в программу и получения от нее результатов. Любое приложение, созданное во Flash или в другом средстве разработки, должно иметь пользовательский интерфейс, иначе оно не сможет получать и выводить данные. Ведь главное предназначение любой программы - обработка данных. А ф.гп.м ну что фильм - крутится себе на экране, независимо, смотрят его или нет.

Поэтому, если вы собираетесь делать на Flash нечто большее, чем фильмы с ограниченными возможностями "общения" со зрителем, вам обязательно нужно позаботиться о пользовательском интерфейсе. И не просто слепить пару элементов управления и связать их парой строк кода - вам придется сделать его удобным для пользователя. Иначе пользователь предпочтет вашей программе другую, благо сейчас у него есть широчайшие возможности выбора.

Итак, какие же средства предоставляет Flash создателям приложений? Давайте их перечислим и кратко рассмотрим.

Кнопки. Это экземпляры образцов-кнопок, впервые упомянутые в главе 10. Образцы-кнопки не очень отличаются от образцов-клипов, за несколькими исключениями, обусловленными самой их "интерфейсной" природой. Кнопки позволяют привязать обработчики к событию press (щелчок на кнопке) и некоторым другим.

Поля ввода и динамические текстовые блоки. Первые позволят принять какие-либо данные от пользователя, а вторые - вывести результаты. Создание полей ввода и динамических текстовых блоков было описано в главе 7. Запомните, что обычные, статические текстовые блоки не могут управляться из сценариев.

Полноценные элементы управления. Собственно, это обычные клипы, созданные самими разработчиками Flash и включенные в его состав. Среди элементов управления вы можете найти флажки, переключатели, списки и пр., что вам привычно по интерфейсу Windows. Но, в отличие от элементов управления Windows, элементы управления Flash могут менять свой внешний вид.

Пользовательские элементы управления. Это новые элементы управления, не встроенные во Flash, а разработанные вами. Для создания пользовательских элементов управления вы должны использовать встроенные клипы. Как это делается, мы рассмотрим в конце данной главы.

Понятие элемента управления неразрывно связано с понятием компонента. (Поэтому мы будем рассматривать их вместе.) Компонентом во Flash называется своего рода "кусок" графики и кода ActionScript, который вы можете вставлять в свои приложения. Все элементы управления, поставляемые с Flash, реализованы в виде компонентов. Вы также можете писать свои компоненты и потом использовать их где угодно и даже передавать коллегам.

Компоненты - одно из значительных нововведений, появившихся в программировании в последнее время. (Снобы от программирования, правда, так не считают, но нам с ними не по пути.) Они позволяют создавать весьма сложные приложения, просто собирая вместе отдельные их "кирпичики", возможно, написанные другими разработчиками. Количество средств разработки, поддерживающих компонентное программирование, неуклонно растет, и совсем недавно к ним добавился и наш любимый Flash.

**Кнопки**

Здесь мы рассмотрим создание и использование кнопок - простейших элементов управления, предоставляемых Flash. Кнопки используются очень часто, и не только в приложениях, но и и обычных интерактивных фильмах Flash, например, для создания Web-сайтов.

**Создание кнопок**

Чтобы создать образец-кнопку, сделайте следующее. Выберите пункт New Symbol в меню Insert или нажмите комбинацию клавиш <Ctrl>+<F8>. Если у вас открыто окно библиотеки, вы также можете выбрать пункт New Symbol в дополнительном меню этого ок,.а или нажать кнопку, показанную на рис. 10.4, эта кнопка находится в нижнем левом углу окна библиотеки. На экране появится диалоговое окно Create New Symbol, показанное на

**Объект Button**

Объект Button "отвечает" за доступ к кнопке из сценариев ActionScript. Этот объект предоставляет набор свойств и методов, с помощью которых и осуществляется управление кнопкой.

Экземпляры объекта Button создаются самим Flash для каждой кнопки, для которой вы задали имя. Вам самим создавать их не нужно.

Во многом объект Button похож на объект -novieciip. Эти два объекта имеют весьма схожий набор свойств и методов и зачастую ведут себя одинаково. В частности, объект Button поддерживает свойства \_х, \_у, xmouse, ymouse, \_height, \_width, \_alpha И \_rotation, уже знакомые вам ПО объекту movieciip. Однако объект Button не поддерживает методы play, stop, gotoAndPiay, gotoAndstop и подобные им, т. к. назначение его все же иное, чем у объекта movieciip. Так что вы можете рассматривать кнопку как сильно специализированный, "урезанный" вариант клипа.

Ниже будут рассмотрены свойства, которые обязательно пригодятся вам в работе с экземплярами объекта Button. (Многие из них поддерживаются и объектом movieciip.)

Свойство enabled позволяет разрешить или запретить доступ пользователя к кнопке. Оно имеет логический тип: значение true разрешает доступ к кнопке, а значение false - запрещает.

Свойство visiо позволяет сделать кнопку видимой или невидимой. Оно имеет логический тип: значение true делает кнопку видимой, а значение false - невидимой.

Свойство useHandCursor позволяет вам сменить курсор мыши, отображаемый, если мышь поместить над кнопкой. Если задано значение true, то отображается курсор в виде "указующего перста", как над гиперссылкой. Если же задано значение false, то отображается обычная стрелка.

Свойство tabindex задает порядок обхода элементов управления при последовательных нажатиях клавиши <ТаЬ>. (При нажатии комбинации клавиш

<Shift>+<Tab> обход выполняется в обратном направлении.) Значением этого свойства может быть любое целое неотрицательное число, оно задает порядок в очереди элементов управления. Так, сначала фокус ввода переместится на элемент управления со значением порядка обхода - 0, потом - со значением 1 и т. д.

Если ни для одного элемента управления в приложении не задан порядок в очереди обхода (то есть, свойство иох равно undefined), Flash будет применять порядок обхода по умолчанию. Если же вы собираетесь задать порядок обхода, имейте в виду, что значение undefined меньше любого числового. Поэтому элементы управления, для которых свойство tabindex не было задано, будут в очереди обхода первыми.

Свойство tabEnabled позволяет убрать кнопку (и вообще любой элемент управления) из порядка обхода, сделать его недоступным для выбора с клавиатуры, но все же доступным для выбора мышью. Это свойство имеет логический тип: значение true или undefined делает элемент управления доступным для выбора с клавиатуры, а значение false - недоступным.

Объект Button поддерживает большое количество событий, которые вы можете использовать для написания обработчиков.

**Форматирование текста**

Как вы помните из главы 7, имеется возможность форматировать текст, по­мещаемый в поля ввода и динамические текстовые блоки. Вы можете выде­лять текст различными шрифтами, цветом, устанавливать различные виды выравнивания для абзацев, задавать отступы текста и красной строки. Все это можно выполнять как в среде Flash, так и с помощью сценариев.

Управление форматированием текста осуществляется с помощью объекта TextFormat. Он содержит ряд свойств, с помощью которых и задаются параметры форматирования текста.

**Объект Selection**

Flash также предоставляет возможность управления текстовым курсором и выделением текста в полях ввода и динамических текстовых блоках. Для этого предназначен объект Selection, единственный экземпляр которого по имени selection создается самим Flash.

С помощью объекта selection вы можете получить позицию текстового курсора в поле ввода. Для этого вам нужно воспользоваться методом getcaretindex. Если же ни одно поле ввода не имеет фокуса, возвращается -1.

Методы getBeginindex и getEndindex возвращают номера соответственно начального и конечного символа выделенного фрагмента текста. Если ничего не выделено, опять же возвращается -1.

Вы можете выделить нужный фрагмент текста, воспользовавшись методом setselection. Первым параметром этого метода должен быть номер первого символа выделяемого фрагмента, а вторым - номер последнего символа. Запомните, что нумерация символов текста начинается с нуля.

**Selection.setselection(10, 30);**

Вы также можете просто поставить текстовый курсор в требуемую позицию, передав методу setseiection номер нужного символа и в первом, и во втором параметре:

**Selection.setselection(10, 10);**

Метод getFocus возвращает имя переменной, привязанной к полю ввода, имеющему в данный момент фокус ввода. Если ни одно поле ввода не имеет фокуса, возвращается null.

Вы можете дать фокус ввода какому-либо полю. Для этого передайте имя переменной, привязанной к этому полю, в качестве параметра методу setFocus:

Selection.setFocus("Name");

Selection.setFocus("\_root.id");

Точно таким же образом вы можете дать фокус ввода кнопке, передав этому методу путь кнопки:

Selection.setFocus("\_root.btnOK");

Чтобы снять фокус ввода со всех элементов управления, передайте методу setFocus значение null.

Вы можете обрабатывать событие onSetFocus, наступающее, когда поле ввода или динамический текстовый блок получает фокус ввода. Для этого вам будет необходимо использовать объект-перехватчик.

**Элементы управления**

Все, что мы проходили ранее, суть простейшие элементы пользовательского интерфейса, предоставляемые Flash разработчикам. С их помощью вы можете создавать только самые простые приложения, которые и приложениями-то назвать можно с трудом: формы ввода, элементы оформления Web-страниц и пр. Конечно, никто не спорит, что и формы ввода данных, и элементы оформления Web-страниц нужны и важны. Но для их создания достаточно зачастую хватит обычных кнопок, можно обойтись даже без полей ввода.

Высший пилотаж Flash-программирования - это создание полноценных приложений, сходных с приложениями Windows. Это могут быть игры, утилиты, калькуляторы, даже текстовые редакторы. Для создания пользовательского интерфейса таких приложений применяются настоящие элементы управления: флажки, переключатели, списки, меню и т. п.

Что такое элементы управления Flash? Ничего особенного - обычные клипы, только ведущие себя особым образом, благодаря соответствующим сценариям ActionScript. Элементы управления имеют набор особых свойств и методов, позволяющих задавать и получать их состояние и управлять ими. Задавать начальные значения свойств вы можете как в среде Flash, так и в сценарии.

**Глава 20. Работа с внешними приложениями**

Flash MX привносит в создание приложений кое-что новое, а именно, компоненты. Компонентом называется специальный клип, имеющий набор свойств и методов и ведущий себя особым образом. Такие компоненты представляют собой отдельные, независимые графические элементы и фрагменты кода ActionScript, которые можно использовать в любом Flash-приложении. Таким образом, компоненты - это особый вид пользовательских объектов.

Типичный пример компонента - любой встроенный элемент управления Flash. Но компоненты не обязательно должны представлять элементы пользовательского интерфейса. Они могут быть обычными графическими элементами, которые используются в фильмах, или вообще быть невидимыми и служить только для целей программирования. Примерами такого рода компонентов могут быть фигурный курсор мыши и невидимый компонент, отслеживающий нажатия клавиш клавиатуры и перемещающий заданный клип по рабочему столу.

Как правило, любой компонент делается независимым от его окружения. Это значит, что вы можете поместить в свой фильм или приложение некоторый компонент, и он будет работать нормально все зависимости от других компонентов фильма или приложения. Если же один компонент требует наличия на листе других компонентов, то об этом особо указывается в его описании.

**Работа с внешними приложениями**

Когда-то давно, лет двадцать назад, компьютеры были однозадачными. Это означало, что на одном отдельно взятом компьютере в данный момент времени могла выполняться только одна программа. Пользователь запускал эту программу, работал в ней, сохранял результат, завершал программу, запускал другую, опять работал в ней, опять сохранял результат, опять выходил и. т. п. Конечно, это было неудобно, но так в то время работали. И, надо сказать, не очень-то жаловались.

Почему? Во-первых, компьютеры двадцатилетней давности были бесконечно слабее современных и просто не могли потянуть даже две одновременно работающих более-менее серьезных программы. Во-вторых, программное обеспечение было принципиально однозадачным, оно просто не могло Позволить пользователю запустить две программы, за редкими исключениями. В-третьих, кому это было нужно: техника делает вид, что работает, пользователи делают вид, что довольны, а значит, зачем напрягаться...

Конечно, существовали способы запускать на тогдашней технике сразу несколько программ, но они налагали на программы очень серьезные ограничения. Было даже что-то подобное настоящим многозадачным операционным системам (например, первая версия Microsoft Windows). Но все это так и не получило широкого распространения. Почему? Ответ на этот вопрос содержится в предыдущем абзаце.

Потом появилась Windows 3.0, за ней - 3.1, 3.11, 95... Компьютеры существенно прибавили в мощности, а программисты наконец-то поняли, что принцип "один компьютер - одна программа" не так уж и хорош. Многозадачная операционная система MS Windows вытеснила старую однозадачную MS-DOS. И каждый пользователь достаточно мощного компьютера смог уподобиться Юлию Цезарю.

Многозадачность принесла еще одну выгоду: теперь разные приложения могли взаимодействовать друг с другом, обмениваясь данными. Первой попыткой реализовать такой обмен данными был DDE (Dynamic Data

Exchange - - динамический обмен данными), реализованный еще в Windows 3.0. Данные передавались от приложения к приложению небольшими порциями и весьма неспешно, да и надежность DDE была невысока. Вторая попытка - OLE (Object Linking and Embedding - связывание и внедрение объектов) - была несравнимо удачнее и применяется до сих пор, став частью технологии COM (Component Object Model - компонентная объектная модель). Существует еще несколько способов передачи данных от приложения к приложению, но они не столь распространены.

А потом получили широкое распространение компьютерные сети, и в мир пришел Его Величество Интернет. Появились серверные приложения, работающие на серверах, не имеющие пользовательского интерфейса и общающиеся с пользователем по сети. Этим они принципиально отличаются от клиентских приложений, работающих на компьютерах пользователей и взаимодействующих с пользователями напрямую. Самые распространенные и наверняка вам знакомые серверные программы - это сервер почты, Web-сервер, FTP-сервер и сервер баз данных.

Пользователь никогда не работает с серверной программой напрямую, а вместо этого использует особые клиентские программы, называемые клиентом или клиентской частью соответствующего сервера. Такие программы формируют запрос, основываясь на введенных пользователем данных, и отправляют его серверному приложению. В ответ последнее возвращает результат обработки этих данных, который клиентское приложение выдает пользователю. Так, для Web-сервера клиентом является Web-обозреватель, а для сервера данных клиентом может выступать любая современная настольная СУБД.

В этой главе мы поговорим о том, как осуществить взаимодействие приложения Flash с внешними по отношении к нему программами и выполнить обработку внешних данных. Вы, как открыть Web-страницу в Web-обозревателе, как посылать команды проигрывателю Flash из сценария на ActionScript. Вы научитесь общаться с серверными приложениями, посылать им запросы и принимать от них результаты. И напоследок вы узнаете, как обрабатывать во Flash-приложениях данные, написанные на новомодном языке описания и форматирования данных XML.

**Управление внешними приложениями**

Здесь мы выясним, как из приложения Flash можно управлять другими программами, установленными на том же компьютере. Также мы узнаем, как можно управлять проигрывателем Flash, в котором работает это приложение.

**Загрузка Web-страницы**

Для загрузки и отображения Web-страницы вам следует воспользоваться действием getuRL. Первым параметром этого действия передается интернет-адрес страницы, которую нужно отобразить, в строковом формате. Вторым параметром может быть передано имя окна Web-обозревателя или фрейма, в котором будет показана страница.

**Управление проигрывателем Flash**

Как вы знаете, все фильмы, и приложения Flash отображаются в особой программе, называемой проигрывателем Flash. Этот проигрыватель может быть как отдельной, независимой программой, так и встраиваемым модулем для Web-обозревателя. (Также само приложение Flash может быть сохранено в виде исполняемого файла, содержащего проигрыватель, но и в этом случае оно считается отдельным приложением.) И вы можете управлять им из сценариев ActionScript, конечно, в некоторых пределах.

**Взаимодействие со сценариями JavaScript**

Web-страница, в которую внедрен фильм Flash, может содержать сценарии JavaScript. Вы можете вызывать эти сценарии и выполнять с их помощью какие-то действия над страницей, пользуясь уже знакомым вам действием FSCommand. Для этого нужно просто следовать несложным соглашениям об именовании сценариев JavaScript, а остальное берет на себя Flash.

**Использование внешних данных**

Мы выяснили, каким образом приложение Flash может управлять внешними программами (а также как внешние программы могут управлять этим приложением). Теперь пришла пора выйти за пределы клиентского компьютера и обратиться к серверным программам.

Как вы уже знаете, серверные программы работают на удаленных компьютерах под управлением программы Web-сервера, с пользователями напрямую не взаимодействуют, а "общаются" с ними только по сети. По сети они принимают от пользователей запросы и так же высылают им ответы: тексты писем, счета, данные из баз и пр. Стало быть, клиентские приложения, "общающиеся" с серверными, должны уметь преобразовывать запросы пользователей в поддерживаемый серверными программами формат. Ну и, конечно, они должны выполнять обратное преобразование, иначе пользователь ничего не сможет разобрать в возвращенном результате.

Flash, как вы уже знаете, делает за вас очень много работы. Он сам кодирует данные, чтобы их можно было нормально передать по протоколу HTTP (HyperText Transfer Protocol - протокол передачи гипертекста), используемому при передаче Web-страниц от сервера к клиенту. (Протоколом называется набор правил обмена информацией по сети, которых должны придерживаться приложения.) Вам остается только правильно сформировать нужный запрос и получить и обработать ответ.

**Использование данных XML**

Интернет-общественность нашла себе новую игрушку. Это широко разрекламированный язык описания данных XML (extensible Markup Language -расширяемый язык разметки), который служит для структурирования данных. Считается, что он должен покончить с неразберихой, связанной с существованием множества несовместимых форматов хранения данных, привести всю информацию, накопленную трудолюбивым человечеством, в строгий порядок. Сбудутся ли эти прогнозы, оправдает ли XML надежды страждущих? Кто знает...

Сейчас мир охватила мода на XML. Очень и очень многие программы спешно обзаводятся поддержкой этого языка, невзирая, нужна она там реально или нет. Не стал исключением и Flash. Уже предыдущая, пятая версия поддерживала данные, отформатированные с использованием этого языка. A Flash MX эту поддержку развил и углубил.

**Глава 21. Средства отладки сценариев ActionScript**

В мире нет ничего совершенного. Даже компьютеры - и те несовершенны, постоянно "зависают", ломаются, "глючат" по любому поводу, а то и без повода. И это понятно: ведь компьютеры - творение людей, а люди да что и говорить!., люди есть люди...

Программы содержат ошибки. Не все, конечно, а те из них, что состоят более чем из двух строк кода. (Функционального кода, который что-то реально делает.) Чем больше и сложнее программа, тем больше (теоретически, по крайней мере) в ней ошибок. Вы и сами, конечно, это знаете: уже всем надоели истории об ошибках в операционных системах Microsoft Windows, да и другие большие программные пакеты не лучше. Производители ПО, разумеется, пытаются с этим бороться различными способами, но пока что толку особо не видно. А проистекает все это безобразие опять же оттого, что программы пишутся людьми. А люди есть люди...

Разумеется, ошибки необходимо исправлять. (Если вы думаете, что их нужно смывать кровью, попытайтесь вспомнить все совершенные вами ошибки и прикиньте, хватит ли у вас крови все их смыть.) Для этого используются мощные программные отладчики, организуются специальные, весьма дорогостоящие мероприятия, выпускаются бесконечные пакеты обновления и т. д. и т. п. И что в результате? Как говорят злые языки, "исправляются старые ошибки и добавляются новые". Не хотелось, конечно, чтобы это было на самом деле, но люди есть люди!

Людям свойственно ошибаться. Хорошо еще, что хоть компьютеры ошибаться не могут в принципе. (Конечно, имеются в виду исправные компьютеры.) Компьютеры лишены свободы воли, они только выполняют программный код, созданный людьми. Для них он всегда правилен.

Но хватит философских отступлений. Давайте поговорим о "вылавливании" ошибок, допущенных вами в сценариях ActionScript, и их исправлении. Одним словом, поговорим об отладке сценариев.

Но прежде - небольшое теоретическое введение. Рассмотрим средства, которые может использовать Flash-программист, чтобы найти ошибки в своих сценариях.

**Как выявить ошибки**

Ошибки, встречающиеся в сценариях ActionScript и программах вообще, можно разделить на два принципиально разных вида. Это ошибки синтаксические и логические.

Синтаксические ошибки - это ошибки и неточности в написании самого кода. Скажем, если вы написали вместо действия else что-то похожее на elswe. Flash предупредит вас о синтаксической ошибке. В самом деле, действия eiswe в языке ActionScript нет, и предупреждение Flash вполне резонно.

Мы рассмотрели простейшую синтаксическую ошибку, которая, что называется, бьет в глаза. Более сложная синтаксическая ошибка - вызов несуществующего метода объекта. Тут сразу трудно понять, что вызывает ошибку. Хорошо, что Flash во многих случаях весьма точно дает знать, что ему не нравится в вашем коде.

Когда вы вводите код в обычном режиме панели Actions, Flash сам следит за правильностью написания кода. Если вы сделаете что-то не так, он подсветит некорректный фрагмент кода красным. Вам будет нужно исправить его, пользуясь соответствующим элементом управления.

**Использование отладчика Flash**

Отладчик Flash активизируется только при проигрывании фильма в так называемом режиме отладки. Во время обычного проигрывания и рисования фильма он недоступен.

Чтобы запустить проигрывание фильма в отладочном режиме, выберите пункт Debug Movie меню Control или нажмите комбинацию клавиш <Ctrl>+<Shift>+<Enter>. После этого фильм будет экспортирован и открыт в отдельном окне Flash, но проигрывание фильма будет приостановлено. Кроме того, на экране появится само окно отладчика. Чтобы запустить проигрывание фильма, щелкните кнопку Continue, расположенную в верхней части этого окна.

Окно отладчика очень похоже на панель Actions. И там, и здесь в правой части находится текстовая область, где отображается отлаживаемый сценарий. Сами сценарии выбираются в раскрывающемся списке, находящемся прямо над этой текстовой областью. В левой части окна отладчика, в отличие от панели Actions, находятся сразу три списка, которые мы рассмотрим далее.

Разделяет эти две части окна отладчика довольно толстая серая полоса, перемещая ее мышью, можно изменять их относительные размеры. На этой полосе также имеется небольшая кнопка, щелкая которую, можно убирать последовательно то одну, то другую часть окна отладчика. Вместо щелчков по кнопке вы можете делать двойные щелчки по самой этой серой линии.

**Удаленная отладка фильмов Flash**

Отладчик Flash предоставляет еще одну интересную возможность - удаленную отладку фильмов. При удаленной отладке фильм загружается не с локального диска, а с Web-сервера. Таким образом, вы можете отлаживать чужие фильмы и приложения, и другие разработчики могут отлаживать ваши (пресловутое разделение труда).

Чтобы удаленная отладка стала возможной, вам следует поместить на Web-сервер вместе с файлом Shockwave/Flash, содержащим фильм, еще один особый файл, содержащий отладочную информацию. Этот файл имеет расширение swd, такое же имя, как у файла фильма, и формируется Flash при экспорте. Если же Flash не найдет на сервере SWD-файл, отладчик не будет работать правильно; в частности, вы не сможете ставить точки останова и трассировать код. Сейчас мы рассмотрим, как создать такой файл и как запустить удаленную отладку.

Сначала нужно экспортировать фильм, задав параметры, разрешающие удаленную отладку. Для этого, прежде всего, откройте нужный документ. В диалоговом окне Publish Settings (см. рис. 19.1), на вкладке Flash включите флажок Debugging Permitted. Как только вы включите этот флажок, станет доступно поле ввода Password, где вы сможете ввести пароль. После этого любой, кто захочет отлаживать ваш фильм, должен будет ввести этот пароль. Используйте его, чтобы не давать просматривать ваши сценарии случайным людям.

После этого опубликуйте или экспортируйте фильм. Выложите на сервер сформированные файлы swf и swd. Все, подготовка к удаленной отладке завершена.

Теперь расскажем, как выполняется удаленная отладка фильма. Предположим, кто-то попросил вас выловить ошибки в своем приложении. Также предположим, что все шаги по подготовке приложения Flash к удаленной отладке сделаны правильно.

Прежде всего, вам нужно включить удаленную отладку в самом Flash. Для этого запустите Flash, откройте любой документ (можно, в принципе, оставить пустой, созданный при запуске) и запустите его отладку. Когда на экране появится окно отладчика, проверьте, включен ли пункт-выключатель Enable Remote Debugging дополнительного меню. Если он отключен, включите его.

Теперь, применив все знания, редактируйте.

**Заключение**

Мы рассмотрели все возможности Flash, описанные в поставляемом с ним электронном руководстве. Мы упомянули также те возможности, которые почему-то в руководстве не были описаны, вероятно, его авторы куда-то торопились и не доделали свою работу. Чтобы раздобыть эту "секретную" информацию, нам пришлось порыться на Web-сайте Macromedia - уж там-то есть все. И, разумеется, пришлось многое пробовать "методом научного тыка", а иначе нельзя узнать программный продукт, тем более, такой сложный.

Но очень многое осталось "за кадром". Мы не говорили о расширениях Flash - дополнительных модулях, подключаемых к основной среде и выполняющих какие-либо специальные задачи. Мы не упоминали о тонкостях работы Flash на компьютерах Apple Macintosh. Мы не описывали многие частные проблемы, с которыми вы вполне можете столкнуться при работе с Flash, так как они появляются достаточно редко, но все-таки появляются. Мы не рассматривали дополнительные программы, поддерживающие формат Shockwave/Masn, в том числе и выпущенные самой фирмой Macromedia. Мы, в конце концов, не говорили о создании серверных приложений и тонкостях языка HTML. Мы о многом не говорили. Ибо невозможно объять необъятное.

Macromedia Flash MX - мощный программный продукт, который еще не раз преподнесет сюрпризы пользователям. Чтобы овладеть им в полной мере, вам также могут понадобиться дополнительные знания по компьютерным сетям, Интернету, серверному программированию, языку HTML и прочим Web-технологиям.

**Литература.**

1.Дронов В. А. Macromedia Flash MX. - СПб.: БХВ-Петербург, 2003. - 848 с.: ил.

2.Гурвиц, Майкл, Мак-Кейб, Лора. Использование Macromedia Flash MX. Специальное издание.: Пер. с англ. — М.: Издательский дом "Вильяме", 2003. — 704 с.: ил.

3.Уотролл Э., Гербер Н. Эффективная работа: Flash MX (+CD). — СПб.; Питер; Киев: BHV, 2003, — 720 с: ил.