

Гэри Розенцвейг

Adobe FLASH

создание аркад, головоломок и
других игр с помощью ActionScript

УДК 004.4'27

ББК 32.973.26 - 018.2

Р64

Розенцвейг Г.

Р64 Adobe Falsh. Создание аркад, головоломок и других игр с помощью ActionScript. Пер. с англ. – М.: ДМК Пресс. – 590 с.: ил.

ISBN 978-5-94074-485-6

Данная книга посвящена программированию игр с помощью ActionScript. Здесь вы найдете подробные указания, необходимые для создания самых разных игр – аркад, головоломок, загадок и даже игровых автоматов. В тексте приведены исходные коды программ и детальные, доступно изложенные инструкции. Базовые принципы программирования ActionScript рассматриваются на примере игр, однако вы без труда сможете применить полученные знания и для разработки неигровых проектов, таких как Web-дизайн и реклама. Рекомендации Гэри Розенцвейга помогут вам не только придумывать занимательные игры и размещать их на Web-сайте, но и оптимизировать скорость их работы, а также защищать свои творения от несанкционированного копирования. Представленный в книге код несложно изменить для использования в других программах.

Книга предназначена для широкого круга читателей – создателей анимационных роликов, художников-оформителей, программистов и разработчиков Web-сайтов. Издание может также выступать в качестве практического пособия по изучению ActionScript.

Authorized translation from the English language edition, entitled Macromedia Flash 8 ActionScript for Fun and Games, published by Que Corporation.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

RUSSIAN language edition published by DMK Press [translator]

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельца авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно остается, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможный ущерб любого вида, связанный с применением содержащихся здесь сведений.

Все торговые знаки, упомянутые в настоящем издании, зарегистрированы. Случайное неправильное использование или пропуск торгового знака или названия его законного владельца не должно рассматриваться как нарушение прав собственности.

ISBN 0-7897-2799-4 (англ.)

Copyright © QUE

ISBN 978-5-94074-485-6

© Перевод на русский язык, оформление
ДМК Пресс

Содержание

Введение	22
1. Инструменты Flash, используемые для создания игр	31
Интерфейс Flash	32
Главное окно Flash	32
Панели	33
Пункты меню	36
Создание кнопок	42
Временная шкала кнопки	43
Пример кнопки	44
Создание текста	45
Свойства текста	45
Динамический текст	46
Редактируемый текст	47
Импорт мультимедийных элементов во Flash ...	47
Растровые изображения	47
Звуки	49
Организация ролика	51
Библиотека	51
Кадры	52
Слои	52
Сцены	53
Нерабочая область	54
Подготовка ролика к публикации	54
Список заключительных проверок	55
Публикация	55
Размещение Flash-ролика в сети	58
2. Введение в ActionScript	61
Урок 1. Что такое код ActionScript	
и где его размещать	62
Сценарии кадра	63
Сценарии для кнопок	63
Сценарии клипов	63

Урок 2. Использование окна Actions	64
Урок 3. Ваша первая программа на ActionScript	66
Урок 4. Контроль воспроизведения ролика	68
Урок 5. Создание кнопок для пользователя	71
Урок 6. Анимация при помощи Action Script	73
Урок 7. Возможность управления клипами пользователем	75
Урок 8. Использование переменных	77
Урок 9. Выполнение операций	78
Урок 10. Условные выражения	80
Урок 11. Циклы	82
Урок 12. Текст и строки	83
Урок 13. Создание функций	85
Урок 14. Массивы	87
3. Элементы ActionScript для создания игр	91
Урок 15. Контроль воспроизведения клипа ...	92
Урок 16. Управление свойствами клипа	94
Урок 17. Применение метода “Перетащи и положи” к клипам	96
Урок 18. Клипы и уровни	98
Урок 19. Копирование клипов	101
Урок 20. Управление несколькими клипами ..	104
Создание клипа сценария	104
Обработчики событий	107
Урок 21. Пересечение объектов	108
Урок 22. Изменения содержимого экрана с помощью клипов	110
Урок 23. Ввод команд с клавиатуры	111
Урок 24. Воспроизведение звука	114
4. Процесс создания игры	117
Flash и игры	118
Недостатки	118
Достоинства	119

Элементы игры	119
Окно загрузки	120
Титульное окно	120
Правила	121
Фон	121
Окно завершения игры	122
Программирование	122
Анализ задачи	123
Правила программирования	124
Отладка	125
Тестирование	125
Вопросы авторских прав	126
Изменение игр, рассматриваемых в данной главе	127
Изменение графики	127
Изменение кода ActionScript	127
Объединение игр	128
Распространенные ошибки	128
Имена клипов	128
Свойства связи	128
Уровни клипа	128
Нижнее подчеркивание	129
Splice и Slice	129
Глобальные и локальные переменные	129
5. Создание визуальных эффектов	131
Увеличение кнопок	132
Задача проекта	132
Подход	133
Подготовка ролика	133
Создание кода	133
К сведению	136
Другие возможности	136
Вращающаяся и изменяющая цвета спираль	137
Задача проекта	137
Подход	137

Подготовка ролика	142
Смена цветов и вращение	142
К сведению	144
Другие возможности	145
Случайность и эффект старого кино	145
Задача проекта	145
Подход	145
Подготовка ролика	147
Создание кода	148
К сведению	150
Другие возможности	150
Перемещение букв	151
Задача проекта	151
Подход	152
Подготовка ролика	152
Создание кода	152
К сведению	154
Другие возможности	155
След от курсора	155
Задача проекта	155
Подход	156
Подготовка ролика	156
Создание кода	156
К сведению	158
Другие возможности	158
Эффект трехмерности	158
Задача проекта	159
Подход	159
Подготовка ролика	159
Создание кода	159
Другие возможности	167
6. Игрушки и безделушки	169
Волшебный хрустальный шар	170
Задача проекта	170
Подход	171

Подготовка ролика	171
Создание кода	173
К сведению	176
Другие возможности	177
Музыкальный автомат	177
Задача проекта	177
Подход	178
Подготовка ролика	179
Создание кода	180
Другие возможности	181
Аналоговые часы	181
Задача проекта	182
Подход	182
Подготовка ролика	184
Создание кода	184
Другие возможности	187
Лавовая лампа	187
Задача проекта	187
Подход	187
Подготовка ролика	188
Создание кода	189
Другие возможности	192
Игра “Жизнь”	192
Задача проекта	192
Подход	193
Подготовка ролика	193
Создание кода	194
К сведению	197
Другие возможности	198
7. Игрушки-конструкторы	199
Составь лису методом “Переташи и положи”	200
Задача проекта	200
Подход	200
Подготовка ролика	201
Создание кода	201

Создай лису методом “Нажми и измени”	205
Задача проекта	205
Подход	205
Создание кода	206
К сведению	207
Другие возможности	208
Альбом для рисования	208
Задача проекта	208
Создание простой программы рисования	208
Создание полной программы рисования	211
Добавление кнопок Clear и Print	215
К сведению	216
Другие возможности	217
Музыкальный миксер	217
Задача проекта	217
Подход	218
Подготовка ролика	218
Создание кода	219
К сведению	221
Другие возможности	222
Танцующая лиса	222
Задача проекта	223
Подход	223
Подготовка ролика	224
Создание кода	224
К сведению	226
Другие возможности	227
8. Игры типа “Найди и нажми”	229
Найди лису	230
Задача проекта	231
Подход	231
Подготовка ролика	232
Создание кода	232
К сведению	234
Другие возможности	234

Поймай лису	234
Задача проекта	235
Подход	236
Подготовка ролика	236
Создание кода	237
К сведению	239
Другие возможности	239
Подстрели лису	240
Задача проекта	241
Подход	241
Подготовка ролика	241
Создание кода	242
К сведению	245
Другие возможности	246
Охота на лис	246
Задача проекта	246
Подход	246
Подготовка ролика	247
Создание кода	248
К сведению	250
Другие возможности	250
9. Игры типа “Поймай или пропусти”	251
Поймай яблоко	252
Задача проекта	253
Подход	253
Подготовка ролика	254
Создание кода	254
К сведению	260
Другие возможности	261
Поймай хорошее яблоко	261
Задача проекта	262
Подход	262
Подготовка ролика	262
Создание кода	262
К сведению	266
Другие возможности	266

Сплав на байдарке	266
Задача проекта	266
Подход	267
Подготовка ролика	267
Создание кода	268
К сведению	273
Другие возможности	273
Гонки	274
Задача проекта	274
Подход	275
Подготовка ролика	276
Создание кода	276
К сведению	281
Другие возможности	281
10. Игры типа “Прицелься и выстрели”	283
Стрельба по воздушным шарам	284
Задача проекта	284
Подход	285
Подготовка ролика	285
Создание кода	286
К сведению	293
Другие возможности	294
Стрельба с прицелом	295
Задача проекта	295
Подход	296
Подготовка ролика	296
Создание кода	296
К сведению	300
Другие возможности	300
Шарики-захватчики	300
Задачи проекта	301
Подход	301
Подготовка ролика	301
Создание кода	302
К сведению	305
Другие возможности	305

11. Игры с мячом и битой	307
Стена и мяч	308
Задача проекта	308
Подход	309
Подготовка ролика	310
Создание кода	310
К сведению	313
Другие возможности	314
Мяч и бита	314
Задача проекта	315
Подход	315
Подготовка ролика	316
Создание кода	316
К сведению	320
Другие возможности	321
Бита и блоки	321
Задача проекта	322
Подход	322
Подготовка ролика	323
Создание кода	324
К сведению	329
Другие возможности	329
Бита и блоки: три измерения	330
Задача проекта	330
Подход	331
Подготовка ролика	331
Создание кода	332
К сведению	338
Другие возможности	338
12. Викторины и загадки со словами	339
Викторина во Flash	340
Задача проекта	340
Подход	341
Подготовка ролика	341
Создание кода	342

К сведению	349
Другие возможности	349
Викторина с учетом времени	349
Задача проекта	350
Подход	350
Подготовка ролика	351
Создание кода	352
К сведению	356
Другие возможности	357
Игра “Виселица”	357
Задача проекта	357
Подход	358
Подготовка ролика	358
Создание кода	359
К сведению	362
Другие возможности	362
Криптограмма	363
Задача проекта	363
Подход	363
Подготовка ролика	365
Создание кода	365
К сведению	370
Другие возможности	370
13. Мозаика	371
Пазл	372
Задача проекта	373
Подход	373
Подготовка ролика	374
Создание кода	375
Другие возможности	376
Пятнашки 377	
Задача проекта	378
Подход	378
Подготовка ролика	378
Создание кода	379

К сведению	382
Другие возможности	382
Игра совпадений	383
Задача проекта	383
Подход	384
Подготовка ролика	384
Создание кода	385
К сведению	388
Другие возможности	388
Найди картинку	388
Задача проекта	389
Подход	389
Подготовка ролика	390
Создание кода	391
К сведению	394
Другие возможности	394
14. Головоломки	395
Игра на развитие памяти	396
Подход	397
Подготовка ролика	397
Создание кода	398
К сведению	402
Другие возможности	402
Дедукция	402
Задача проекта	403
Подход	403
Подготовка ролика	405
Создание кода	405
К сведению	410
Другие возможности	410
Йога	410
Задача проекта	411
Подход	411
Подготовка ролика	412
Создание кода	412
Другие возможности	417

Рекурсивные блоки	417
Задача проекта	418
Подход	418
Подготовка ролика	419
Создание кода	419
К сведению	424
Другие возможности	424
15. Казино и карточные игры	425
Игровой автомат	426
Задача проекта	426
Подход	427
Подготовка ролика	428
Создание кода	428
К сведению	433
Другие возможности	434
Видеопокер	434
Задача проекта	434
Подход	435
Подготовка ролика	436
Создание кода	437
К сведению	445
Другие возможности	446
Игра в очко, или двадцать одно	446
Задача проекта	446
Подход	447
Подготовка ролика	448
Создание кода	449
К сведению	455
Другие возможности	455
Пасьянс “Пирамида”	456
Задача проекта	457
Подход	457
Подготовка ролика	458
Создание кода	458
К сведению	465
Другие возможности	465

16. Аркады	467
Космический бой	468
Задача проекта	468
Подход	469
Подготовка ролика	469
Создание кода	470
К сведению	476
Другие возможности	476
Астероиды	477
Задача проекта	477
Подготовка ролика	478
Создание кода	478
К сведению	487
Другие возможности	487
Погоня в лабиринте	488
Задача проекта	488
Подготовка ролика	488
Создание кода	490
К сведению	497
Другие возможности	497
Луноход	498
Задача проекта	498
Подготовка ролика	499
Создание кода	501
К сведению	506
Другие возможности	506
Платформенный скроллер	506
Задача проекта	507
Подход	508
Подготовка ролика	509
Создание кода	509
К сведению	519
Другие возможности	519

17. Сложные приемы	521
Выполнение и оптимизация	522
Использование альтернативных возможностей	522
Установление контрольных точек	523
Выбор лучшей альтернативы	525
Оптимизация в ActionScript	526
Экран загрузки	528
Создание полосы загрузки	528
Сложные экраны загрузки	530
Другие типы экранов загрузки	532
Защита от копирования	533
Как и почему воруют игры	533
Как предотвратить воспроизведение игры с другого компьютера	534
Как обойти защиту от копирования	536
Другие методы защиты	537
Список лучших игроков	537
Как работает список лучших игроков	537
Программы на языке Perl	539
Демонстрация списка лучших игроков	544
Добавление списка лучших игроков в игру	546
Сложности применения списка лучших игроков	548
Определение версии Flash-плагина	549
Как узнать версию плеера	550
Требуемые версии	551
Возможности Flash 8-плеера	552
Приложения	
А. Ресурсы сети Internet для Flash-разработчика	555
В. Краткий экскурс в историю игр	559
Предметный указатель	570

Прежде чем приступить к изучению того, как создаются Flash-игры, давайте попробуем взглянуть на это с точки зрения человека, работающего с ActionScript (язык программирования, используемый в программе). Данная книга не является пособием для начинающих, поэтому в ней не излагаются основы, тем более что в пакет Flash входят специальные обучающие программы. Вместо этого в книге подробно рассматриваются различные инструменты Flash 8 и способы их использования в играх, написанных на ActionScript.

Задача первых четырех глав – развить ваши навыки и тем самым подготовить к изучению примеров, содержащихся в остальных главах книги.

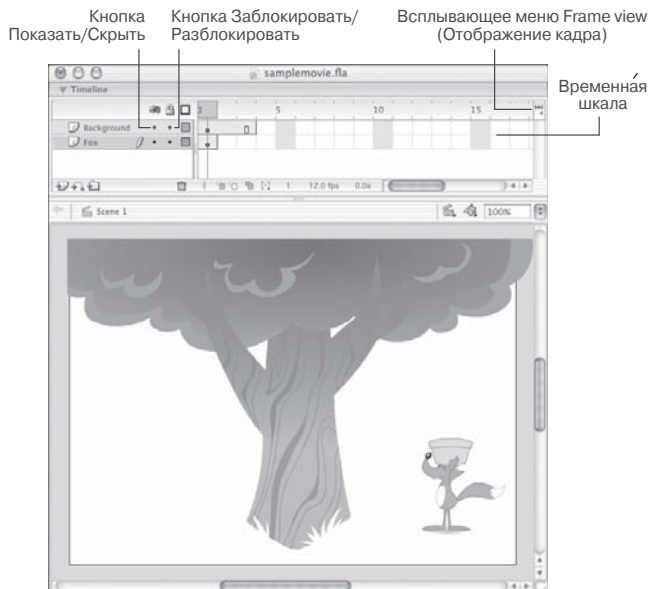
Интерфейс Flash

Прежде чем создавать сложные программы на ActionScript, вам необходимо хорошо изучить интерфейс Flash, который включает множество окон, панелей и меню.

Главное окно Flash

Основой интерфейса Flash является главное окно и его важнейший элемент – *рабочее поле*. На рис. 1.1 показано рабочее поле, которое содержит изображение большого дерева и лисы, держащей корзину.

Рисунок 1.1
Главное окно –
основной элемент
интерфейса Flash



Для программиста на ActionScript одни элементы главного окна более важны, чем другие. В верхней части расположена временная шкала, в которой некоторые кадры помечены номерами 1, 5, 10 и 15. Сбоку временной шкалы находится панель управления слоями. В данном примере показаны два слоя: «Background» (Фон) и «Fox» (Лиса).

Вы наверняка заметили, что кадры на временной шкале намного шире, чем обычно. Это связано с тем, что поле представления кадров было расширено при помощи кнопки Frame view (Отображение кадра), показанной на рис. 1.1. Существует еще несколько опций по управлению представлением кадров, например можно регулировать высоту каждого слоя (находится в свойствах слоя). Поэкспериментируйте с ними самостоятельно.

Область рабочего поля главного окна может содержать любой элемент вашего ролика: графический образ, кнопку или клип.

Изображение, показанное на рабочем поле главного окна, можно увеличить или сжать. Для этих целей предназначено всплывающее меню Scale (Масштаб), расположенное в левом нижнем углу главного окна, но для изменения масштаба намного удобнее пользоваться «горячими» клавишами. Нажатие клавиш Cntrl+2(PC) или Command+2(Mac) масштабирует рабочее поле так, чтобы оно вплотную занимало все свободное место в главном окне. Использование сочетания клавиш Cntrl+3 или, соответственно, Command+3 масштабирует рабочее поле так, что в главном окне максимально распаивается область, содержащая все изображения, в том числе те, которые находятся на серой области вне пределов рабочего поля. Сочетание Cntrl+1 (Command+1) возвращает рабочему полю масштаб 100%. И наконец, использование командных клавиш в сочетании с + и – позволяет изменять текущий масштаб в два раза.

Последнее, о чем хотелось бы здесь сказать, – кнопки Show/Hide (Показать/Скрыть) и Lock/Unlock (Заблокировать/Разблокировать). Эти функции чрезвычайно полезны тем, что позволяют отобразить отдельные слои или определенный набор слоев. Блокировка слоев позволяет без труда выбрать нужный вам объект, не внося при этом случайных изменений в другой слой.

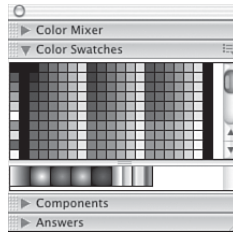
Панели

Хотя главное окно программы Flash и является основным элементом, оно содержит очень мало подробной информации о вашем ролике и его частях. Детали отображаются в небольших окнах панелей управления. При запуске они появляются в правой и нижней частях главного окна, однако затем их можно перемещать куда угодно.

Эти панели могут быть приведены к любому виду. Панели находятся каждая в своем окне, но можно отобразить сразу несколько из них в одном окне в виде закладок. Например, на рис. 1.2 показано окно, включающее панели Color Mixer (Миксер цветов), Color Swatches (Образцы цвета), Components (Компоненты) и Answers (Ответы). Только панель Color Swatches доступна к использованию.

Рисунок 1.2

Это окно
содержит четыре
панели



Вы можете изменить состав окна панелей, а также создать новое окно. Это осуществляется перетаскиванием панелей за левую, помеченную характерной “дырчатой” текстурой, часть титульного поля панели. Для того чтобы научиться перемешать панели, требуется немного практики. У вас всегда есть возможность вернуть исходные настройки, выбрав команду Window ⇒ Panel Sets ⇒ Default Layout (Окно ⇒ Наборы панелей ⇒ Расположение по умолчанию).



СОВЕТ

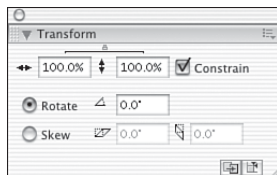
Необязательно перемешать панели между окнами панелей только потому, что у вас имеется такая возможность. Я нахожу расположение по умолчанию самым лучшим вариантом, которого и придерживаюсь. Это также удобно, если вам придется работать на другом компьютере, в этом случае вам не надо будет привыкать к какому-либо нестандартному расположению. Вы можете сохранить собственную конфигурацию панелей с помощью команды Window ⇒ Save Panel Layout. Сохраненные конфигурации доступны с помощью команды Window ⇒ Panel Sets.

Панель Info, которую можно выбрать командой Window ⇒ Info, необходима для точного размещения объектов на рабочем поле. Выбрав объект, вы можете задать значения его координат (X и Y), а также его ширину и высоту (W и H).

Панель Transform (рис. 1.3) можно выбрать командой Window ⇒ Transform. Эта панель позволяет изменять масштаб выделенного объекта и поворачивать его.

Рисунок 1.3

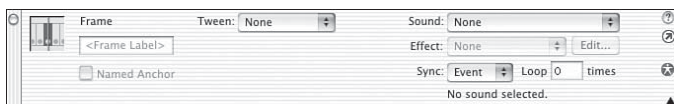
Панель *Transform* позволяет изменять масштаб выделенного объекта и поворачивать его



Еще одна панель, которую постоянно используют программисты на ActionScript, – *Properties* (Свойства). Эта панель (рис. 1.4) отображает информацию о выбранном в данный момент объекте: графическом символе, кнопке или клипе. С помощью этой панели вы можете менять различные свойства выбранных объектов. Например, ставить метки выделенным кадрам и присваивать имена клипам. Для того чтобы можно было программно обращаться к клипам, им необходимо присвоить имена.

Рисунок 1.4

Панель *Properties* позволяет присваивать имена клипам

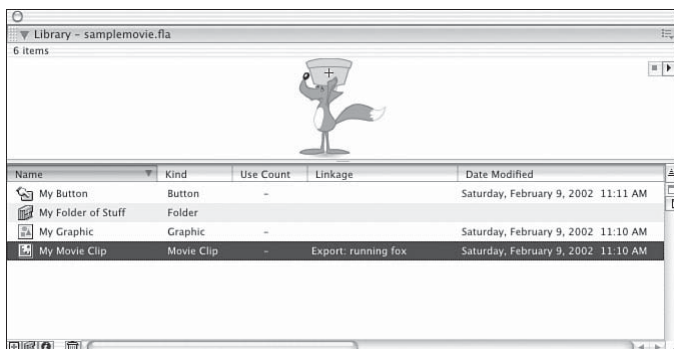


Панель *Properties* (Свойства) позволяет присваивать имена (метки) кадрам. Как ни странно, вы можете сделать это в любом слое. Например, вы можете присвоить имя первому кадру, выделив его в верхнем слое, затем присвоить имя второму кадру, выделив его в следующем слое. Неправда ли, сбивает с толку? Я советую пометать все кадры в одном слое (обычно так и делают, называя этот слой, например, “labels”).

Наиболее важным окном после рабочего поля как для программиста на ActionScript, так и для художника-графика, несомненно, является палитра

Рисунок 1.5

Палитра *Library* показывает, из каких элементов состоит ваш ролик



Library (Библиотека). Панель на рис. 1.5 содержит растровое изображение, кнопку, папку, графический символ и клип.

Палитра Library показывает список различных элементов, составляющих ваш ролик. На первый взгляд элементы списка кажутся выбранными наугад. Например, графические образы, которые были нарисованы непосредственно на экране, не отображаются в палитре, даже если они были сгруппированы. Если же вы сохраните их как графический символ, они будут показаны.

Логика здесь такая: библиотека хранит многократно используемые объекты. Любой символ¹, будь то клип, графический символ или кнопка, могут использоваться в одном или нескольких местах вашего ролика. Однако набор линий и заливок, нарисованных непосредственно на рабочем поле, может применяться только в кадре или наборе кадров на временной шкале.

Включение эталона в библиотеку не означает, что он будет экспортирован вместе с роликом. Например, вы импортируете растровое изображение или звуковой файл, но не помещаете его на рабочее поле. Или же создаете клип и не используете его. В таком случае умная программа не включает этот объект в конечный сжатый swf-файл, что способствует уменьшению размера файла и его быстрой загрузке.

Что делать, если элемент библиотеки не выложен на рабочее поле, а ваша программа его использует? Flash не будет отслеживать это и не включает данный элемент в конечный файл ролика.

Для того чтобы заставить Flash включить в конечный файл кажущийся ему ненужным элемент библиотеки, установите связь с этим элементом в диалоговом окне Symbol Linkage Properties (Свойства связи символа) – см. рис. 1.6. Диалоговое окно можно вызвать из всплывающего меню Options (Параметры), расположенного в правом верхнем углу палитры Library (рис. 1.5).

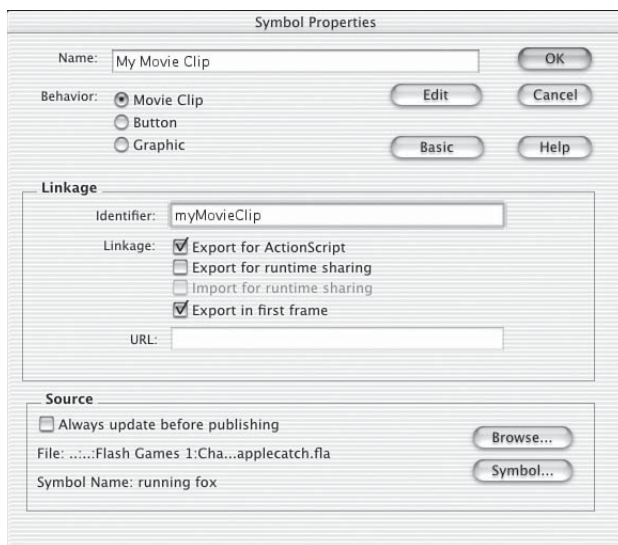
Пункты меню

Выбор пунктов меню столь же богат, как и выбор панелей. Однако регулярно используется лишь небольшая их часть. Программисту на ActionScript, помимо расположения самих пунктов меню, необходимо знать сочетания “горячих” клавиш для выполнения наиболее часто используемых команд.

¹ Преобразовав изображение в символ, вы тем самым создаете экземпляр этого символа в библиотеке. Перевод термина “Symbol” как “эталон” или “образец” подразумевает, как правило, что речь идет о библиотечном объекте, используемом как образец для создания символов непосредственно на рабочем поле. – *Прим. науч. ред.*

Рисунок 1.6

В диалоговом окне *Symbol Properties* можно указать, что объект должен быть включен в библиотеку для последующего экспорта в Action Script

**Меню File**

Меню File (Файл) включает стандартный набор команд, знакомый пользователям Macintosh и Windows. Наряду с командами Open (Открыть), Save (Сохранить) и Save As (Сохранить как) здесь присутствует также команда Import (Импорт), позволяющая импортировать в программу растровые изображения, векторную графику и звуковые файлы.

Наиболее важной в меню File является простая команда Save, которой соответствует сочетание клавиш Command+S (Macintosh) и Ctrl+S (Windows). Советуем вам почаще ее использовать. Если после многочасовой работы без сохранения программа или компьютер дадут сбой, вы пожалеете о том, что не нажимали данное сочетание клавиш каждые пять минут.

**СОВЕТ**

К сожалению, Flash время от времени имеет обыкновение давать сбой, что приводит к потере всей вашей работы со времени последнего сохранения. Компания Macromedia регулярно поставляет новые, исправленные и усовершенствованные версии Flash. Обратной стороной такого стремительного развития программы является ее нестабильность. Тем не менее, если вы не будете забывать почаще сохранять вашу работу, возникающие сбои не испортят вам настройки.

В меню File также находятся команды Publish (Публикация) и Publish Settings (Параметры публикации). Подробнее об этих командах вы узнаете в конце данной главы.

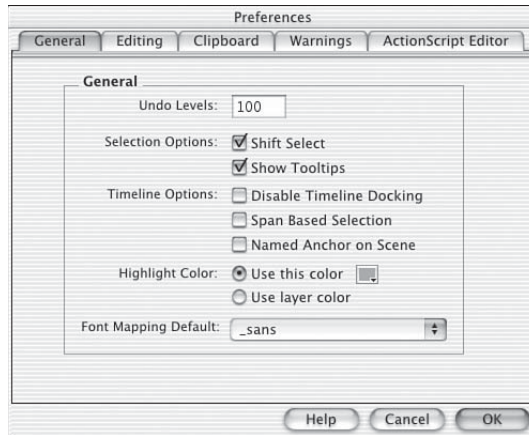
Меню Edit

Меню Edit (Правка) включает обычные команды: Copy (Копировать), Cut (Вырезать) и Paste (Вставить). Команда Undo (Отменить) особенно полезна, так как с ее помощью вы можете отменить 100 последних действий, что является довольно значительной цифрой. Вы можете задать количество отменяемых действий, выбрав команду Edit ⇒ Preferences (Правка ⇒ Настройка).

При выборе команды Preferences на экране появится диалоговое окно, показанное на рис. 1.7. Здесь вы можете не только указать число отменяемых действий, но и задать целый ряд настроек, упорядоченных в три отдельные вкладки. Что касается меня, я стараюсь придерживаться настроек по умолчанию, однако иногда не помешает знать, какие опции вам предлагаются. Обычно я меняю предлагаемый по умолчанию режим Normal (Обычный) на режим Expert (Эксперт), что позволяет мне набирать команды ActionScript непосредственно в окне редактирования сценария.

Рисунок 1.7

Для того чтобы вызвать диалоговое окно Preferences, используйте основное меню, но его расположение зависит от вашей операционной системы



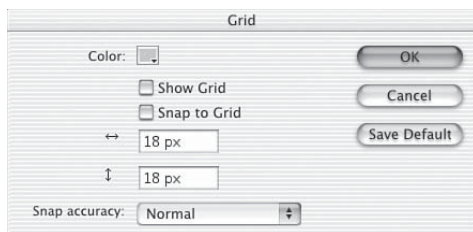
Меню View

Меню View (Вид) позволяет изменять масштаб отображения рабочего поля и степень детализации отображаемых на нем графических элементов во время работы над роликом, а также использовать сетки и направляющие.

Сетка пригодится для быстрого выравнивания элементов на рабочем поле в том случае, если вам важнее составить программу на ActionScript, а не найти лучший способ расположения графических объектов. Чтобы вызвать диалоговое окно Grid (Сетка), показанное на рис. 1.8, надо выбрать команду View ⇒ Grid ⇒ Edit Grid (Вид ⇒ Сетка ⇒ Редактировать сетку). Здесь вы можете задать масштаб сетки и ее цвет, указать, будет ли сетка отображаться на экране и будут ли к ней привязываться объекты. Диалоговое окно Grid позволяет быстрее задавать и отменять данные настройки, чем команды Show Grid (Показать сетку) и Snap To Grid (Привязать к сетке).

Рисунок 1.8

В диалоговом окне Grid задаются цвет и другие настройки сетки



Меню View включает команду Hide Panels (Скрыть панели), которую можно также выполнить простым нажатием на клавишу табуляции. При этом все панели, включая палитру Tools (Инструменты), будут моментально скрыты или отображены. Это удобно, если некоторые часто используемые панели закрывают рабочее поле, и вы хотите их скрыть, но так, чтобы в следующий раз они появлялись на том же самом месте экрана.

Меню Insert

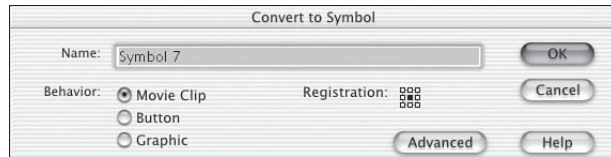
Меню Insert (Вставка) содержит две функции, полезные для программиста на ActionScript. Первая из них – команда Convert to Symbol (Преобразовать в символ), которая преобразует выделенные вами элементы в графический объект, клип или кнопку. Вновь созданный эталон включается в библиотеку, а также остается на рабочем поле на месте исходных элементов.

При вызове данной команды на экране появляется диалоговое окно Symbol Properties (Свойства символа), в котором вы должны задать тип символа и назначить ему имя (рис. 1.9).

Команда Convert to Symbol позволяет создавать графические объекты, клипы и кнопки непосредственно на рабочем поле и затем добавлять их в библиотеку. Вы также можете использовать эту команду для создания символов внутри символов. Например, вы можете создать объект, похожий на кнопку, при помощи команды Convert to Symbol превратить его в кнопку, а затем, повторно используя данную команду, поместить кнопку в клип.

Рисунок 1.9

Диалоговое окно *Symbol Properties* появляется на экране при выборе команды *Insert* ⇒ *Convert to Symbol*. Это же окно появится, если вы захотите изменить свойства уже существующего символа



Второй важной функцией меню *Insert* является набор команд, позволяющих добавлять кадры в главную временную шкалу. Чтобы добавить новый кадр, используйте команду *Insert* ⇒ *Frame* (Вставить ⇒ Кадр).

Для добавления нового ключевого кадра во временную шкалу предназначены команды *Insert* ⇒ *Keyframe* (Вставить ⇒ Ключевой кадр) и *Insert* ⇒ *Blank Keyframe* (Вставить ⇒ Пустой ключевой кадр). В первом случае будет создана копия текущего ключевого кадра, во втором – новый ключевой кадр, не содержащий никаких объектов на рабочем поле. При помощи команды *Insert* ⇒ *Remove Frames* (Вставить ⇒ Удалить кадры) можно удалить выбранные в слое кадры, а используя *Insert* ⇒ *Clear Keyframe* (Вставить ⇒ Очистить ключевой кадр), удалить ключевой кадр, не удаляя простых кадров слоя.

Команды для работы с кадрами могут немного сбить с толку тех, кто никогда до этого не создавал анимации во Flash. Лучший способ научиться работать с этими командами – попробовать их применить. Создайте ролик и потренируйтесь добавлять и удалять кадры и ключевые кадры. Тогда вы начнете понимать, как работают эти команды.



СОВЕТ

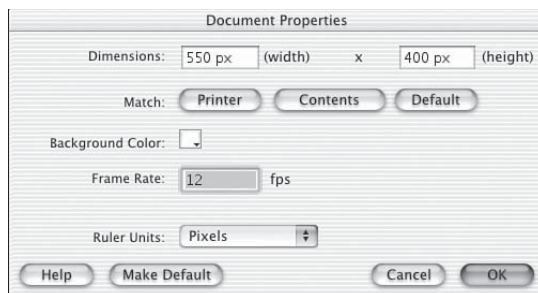
Разница между кадром и ключевым кадром состоит в следующем: ключевой кадр – это точка временной шкалы, определяющая точное положение всех объектов на рабочем поле данного слоя. Они не будут менять своего положения до следующего ключевого кадра. Однако аниматоры могут выбрать, например, анимацию движения между ключевыми кадрами, при этом объекты рабочего поля будут постепенно передвигаться из позиции, определяемой первым ключевым кадром, к позиции, определяемой следующим ключевым кадром. Более подробно о ключевых кадрах вы узнаете из урока анимации, поставляемого с программой Flash 8.

Меню Modify

Меню Modify предоставит вам еще несколько способов быстрого доступа к панелям. Команда Modify ⇒ Document открывает окно свойств документа (рис 1.10). Диалоговое окно Document Properties (Свойства документа) позволяет изменить размеры ролика, количество кадров в секунду и цвет фона.

Рисунок 1.10

В диалоговом окне Document Properties задаются наиболее важные свойства ролика



Также заслуживает упоминания набор команд подменю Transform (Трансформация). Эти команды служат для изменения масштаба, вращения и транспонирования графических объектов на рабочем поле.

Далее в книге рассказано, как выполнять подобные действия при помощи ActionScript, но некоторыми графическими изображениями и символами невозможно управлять при помощи ActionScript, поэтому эти команды надо обязательно знать.

Следует также остановиться на команде Group (Группировать). В случае наложения друг на друга графических объектов линии и заливка объекта, расположенного снизу, могут оказаться скрытыми. Например, если вы нарисуете большой синий круг, в центре этого круга нарисуете маленький красный круг, а затем удалите красный круг, то в центре первого круга, на том месте, где был расположен красный, получится дырка. Если же вы сгруппируете синий круг перед тем, как рисовать красный, то первый не окажется частично стертым, а останется в своем первоначальном виде под вторым. Подробнее о группировании вы узнаете из урока по рисованию, поставляемого с пакетом Flash 8.

Меню Text

Включенные сюда команды обеспечивают быстрый доступ к опциям панелей Text (Текст). Более подробно о работе с текстом мы расскажем в разделе “Создание текста” данной главы.

Меню Control

Меню Control (Управление) включает постоянно используемую команду Test Movie (Пробное воспроизведение ролика). Данная команда, выполняемая также нажатием клавиш быстрого доступа Command+Return (Macintosh) или Ctrl+Enter (Windows), преобразует ваш текущий ролик в конечный swf-файл и затем запускает его в окне предварительного просмотра.

Ролик воспроизводится аналогично тому, как он бы воспроизводился в проигрывателе Flash или в Web-браузере пользователя. Данная команда отличается от команды Play (Воспроизведение) из меню Control, которая не использует никаких сценариев ActionScript, находящихся в ролике. Команда Play совершенно бесполезна для программиста, в то время как команда Test Movie бесценна.

Меню Window

С помощью меню Window (Окно) вы можете открыть все панели, а также все другие окна. Сочетания клавиш для быстрого доступа к наиболее часто используемым инструментам приведены в таблице:

Для доступа...	Нажмите на Macintosh	Нажмите на Windows
Properties Panel	Command+F3	Cntr+F3
Info Panel	Command+I	Ctrl+I
Library	Command+L	Cntrl+L
Reference Panel	Shift+F1	Shift+F1
Actions Panel	F9	F9

Окно справки

В окне Reference (Справка) содержится словарь ActionScript программы Flash. Даже самый опытный программист ActionScript не будет знать всех свойств, применений и отличительных особенностей синтаксической структуры Flash. Это меню позволит вам быстро получить необходимую информацию об определенной команде или функции.

Создание кнопок

Во Flash существует три типа символов: графические объекты, кнопки и клипы. В отличие от графических объектов кнопки и клипы могут иметь закрепленные за ними сценарии ActionScript. Программисту на ActionScript приходится создавать и использовать множество кнопок и клипов, поэтому давайте рассмотрим, начиная с кнопки, как это сделать.

Временная шкала кнопки

Кнопку можно создать следующими способами: выбрать команду Insert ⇒ New Symbol (Вставить ⇒ Новый символ) или выделить ряд объектов на рабочем поле и выбрать команду Insert ⇒ Convert to Symbol (Вставить ⇒ Преобразовать в символ). Вы также можете выбрать команду New Symbol из всплывающего меню палитры Library.

Создав кнопку, вы можете редактировать прямо на рабочем поле, выбрав ее двойным щелчком мыши, или отдельно, в главном окне Flash, два раза щелкнув по ее имени в окне палитры Library.

В обоих случаях в верхней части главного окна Flash появится специальная временная шкала, которая содержит четыре изображения, необходимые для описания кнопки: Up, Down, Over и Hit.

Вы можете добавлять в кнопку сколько угодно слоев. При отображении кнопки будут использоваться только первые четыре кадра, показанные на рис. 1.11. Вот для чего они предназначены:

- Up – обычное изображение кнопки;
- Over – вид, который принимает кнопка при помещении над ней курсора мыши;
- Down – вид кнопки, когда пользователь уже нажал, но еще не отпустил ее;
- Hit – определяет область, реагирующую на курсор.

Вначале лучше сосредоточить свое внимание на создании кадра Up. Затем создать варианты кадров Over и Down. В кадре Over кнопка должна каким-то образом выделиться, в то время как в кадре Down ее следует

Рисунок 1.11

Временная шкала, расположенная над кнопкой, содержит предопределенный набор из четырех кадров



изобразить нажатой или выделить каким-то другим способом. В следующем разделе вы увидите несколько примеров дизайна кнопки.



СОВЕТ

Вы можете создать кнопку, включающую только кадр Up. Если остальные три кадра останутся пустыми и будет отсутствовать даже ключевой кадр, кадр Up будет применен для всех четырех состояний кнопки.

Отличительной чертой кадра Hit является то, что он никогда не виден. Также не важно, какого цвета расположенный в кадре Hit объект; важна лишь форма. Кадр Hit вместе с кадрами Over и Down вообще можно не создавать. В этом случае все эти кадры заменит кадр Up.

Кадр Hit можно представить как невидимую фигуру под кнопкой, определяющую область нажатия кнопки мышью. Эта область может быть больше или меньше видимого графического изображения кнопки.



СОВЕТ

Вы можете создать невидимую кнопку, оставив кадры Up, Over и Down пустыми, а кадр Hit заполненным. При этом в главном окне Flash кнопка будет выглядеть как светло-голубая фигура. При воспроизведении ролика ее не будет видно вообще. Несмотря на это, на кнопку можно нажимать, а ее сценарий будет выполняться. Пользователь поймет, что это кнопка, так как при наведении на эту область курсор примет специальный вид. Такой способ применяется для создания различных эффектов.

Пример кнопки

🌐✂️ Исходный файл: [Samplebuttons fla](#)

Давайте рассмотрим пример кнопки. На рис. 1.11 показана кнопка прямоугольной формы с надписью. Она была нарисована непосредственно на рабочем поле при помощи инструмента Rectangle (Прямоугольник) и затем преобразована в кнопку при помощи команды Insert ⇒ Convert to Symbol. Ролик с кнопкой находится на Web-сайте в файле Samplebuttons fla в папке данной главы.

Процесс создания данной кнопки начался с выбора инструмента Rectangle. Два раза щелкнув по инструменту, вы получаете возможность задать значение радиуса для скругленного угла прямоугольника. В нашем примере было использовано значение 2. Для линий был задан черный цвет, для заливки – голубой.

После того как прямоугольник был нарисован, я выбрал инструмент Text (Текст) и напечатал “PRESS ME!” (НАЖМИ МЕНЯ!). После центровки текста получилось изображение, показанное на рис. 1.11.

После использования команды Преобразовать в символ, я получил кнопку и присвоил ей имя “My Button” (Моя кнопка). Новый символ был включен в библиотеку и остался на рабочем поле на прежнем месте.

Два раза щелкнув кнопкой мыши по символу в библиотеке, вы можете отдельно редактировать кнопку на рабочем поле. В верхней части главного окна Flash появятся четыре предустановленных кадра, при этом все кадры, кроме кадра Up, будут пустыми. Кнопка будет работать отлично, однако станет использовать одно и то же изображение для всех своих состояний. Форма области в кадре Hit будет определяться самим изображением кнопки.

Чтобы внести разнообразие, необходимо создать ключевой кадр для каждого из четырех имеющихся кадров. Для этого выберите кадр на временной шкале и нажмите клавишу F6 (вставка ключевого кадра). Затем вернитесь к кадрам Over и Down и внесите в них какие-нибудь изменения. В нашем примере была изменена заливка каждого из кадров. Для кадра Hit текст не обязателен, значение имеет лишь форма изображения.

Создание текста

Во Flash имеется три различных типа текстовых окон. Первое, статический текст, – если рассматривать его с точки зрения программиста на ActionScript, – это не более чем графические объекты в форме текста. Другие два, динамический и редактируемый тексты, могут быть изменены при помощи ActionScript.

Свойства текста

При выделении текстового поля окно Properties будет отображать свойства текста (рис. 1.12). Наиболее важные свойства находятся в верхнем левом углу панели Properties. На рис.1.12. выбран тип Static Text; вы можете изменить это свойство на Dynamic Text или Input Text.

Рисунок 1.12
Окно Properties
позволяет
изменять свойства
текста

