

Macromedia Flash 5

Дизайн и программирование в одном флаконе

В свое время программисты проделали долгий путь от командной строки и текстовых редакторов к визуальным средам разработки приложений. Сегодня почти все ведущие производители средств программирования выпекают визуальные среды типа Delphi и Borland C++ Builder. Я — старый программист, и не знаю слов любви, но когда у меня впервые появилась одна из таких систем — Borland C++ Builder — волна восторга и воодушевления охватила меня с ног до головы. Наконец-то можно было заняться просто программированием, не путаясь в многочисленных нюансах и технических деталях интерфейса.

Одним из важных преимуществ визуальных сред было то, что они освободили программистов от рутины по созданию внешнего вида программы.

Borland C++ Builder, к примеру, имел библиотеки объектов, реализующих готовые элементы интерфейса — кнопки, меню, текстовые поля ввода и многое другое. Было так удобно бросить пару готовых кнопок на окно создаваемой программы и заняться отладкой ее внутренних алгоритмов.

Немного истории

Примерно в то же время, в конце 90-х гг. прошлого века, развивался и рос Web, словно продукт творчества каких-то электронных пауков, решивших оплести нашу планету глобальной системой коммуникации. Главным в Web был... нет, не контент, как считают некоторые. Главным в Web и в первых браузерах стал удобный интерфейс, позволявший одним нажатием мышки переходить куда вздумается. HTML и браузеры избавили пользователей от не-

обходимости пользоваться командной строкой. В 1996 г. я читал перевод книги про Интернет какого-то зарубежного автора. Книга была очень толстой, на обложке был изображен то ли тушканчик, то ли суслик. Но не это главное. Главное, что из всей массы (страниц 500) только процентов 10 посвящалось WorldWideWeb и привычным ныне страничкам. Все остальное в книге долго и вкусно рассказывало о программах, работающих с командной строки: Telnet, FTP, Gopher и других. Кто-то не зна-»

» ет, что такое командная строка? Вы меня удивляете — это же просто черный экран с текстом! К счастью, Web навсегда похоронил командную строку для массового пользователя, оставив ее сугубо для профессионалов. И правильно сделал: именно благодаря этому привлекательность Интернета взлетела на недосягаемую высоту.

Именно природа Web как интерфейса толкала всех на часто необдуманные эксперименты с HTML-страничками. Интернет быстро перешел от текстового вида к мультимедийному наполнению. Разные производители в своих программных продуктах предлагали решения, пытались привить мировой паутине графику, музыку, видео и, наконец, интерактивное взаимодействие с пользователем. И на гребень волны взлетел формат, предложивший все это вместе взятое плюс легкость в создании, привычный для дизайнеров стиль работы и малые размеры файлов.

Речь идет об очень популярном формате Flash от Macromedia. Назвать его просто анимацией или векторной графикой не поворачивается язык. Flash! Как много в этом звуке для web-дизайнера слилось. Стартовав как формат векторной анимации для Web, Flash превратился в мультимедийный язык программирования с самой широкой сферой применения — от онлайн до офлайн, от интернет-страничек до компьютерных игр и рекламных роликов для телевидения.

Последняя на момент написания статьи версия Flash 5 — это, безусловно, самая настоящая визуальная среда программирования, кое в чем даже превосходящая своих старших братьев.

Программировать — это легко

В Flash 5 вновь улучшен встроенный язык ActionScript. Обычно его сравнивают с JavaScript, а я сравню его с C++. Анимационный ролик необязательно может двигаться кадр за кадром, анимации может вообще не быть. Мощь ActionScript позволяет легко освоить программирование даже тем, кто никогда и не мечтал об этом. Почему? Потому что редактор ActionScript поддерживает и направляет неопытного программиста, аки пастух заблудшего ягненка. Сле-

зуется, ввести их с клавиатуры. Программирование больше не выглядит как безостановочный долбеж по клавиатуре. Все очень просто. Создаем объект или выбираем кадр в Macromedia Flash, вызываем окно редактора с помощью команды Window -> Actions, выбираем из меню нужную команду и нужные параметры. Ни строчки кода вручную. Причем сделать шаг влево или вправо от «правильного» кода очень трудно, что чрезвычайно облегчает задачу написания корректно работающей программы. Это справедливо для режима Normal, в котором редактор ActionScript работает по умолчанию. Впрочем, тот, кто считает себя экспертом, может выбрать режим Expert: в нем можно набивать команды вручную и писать что

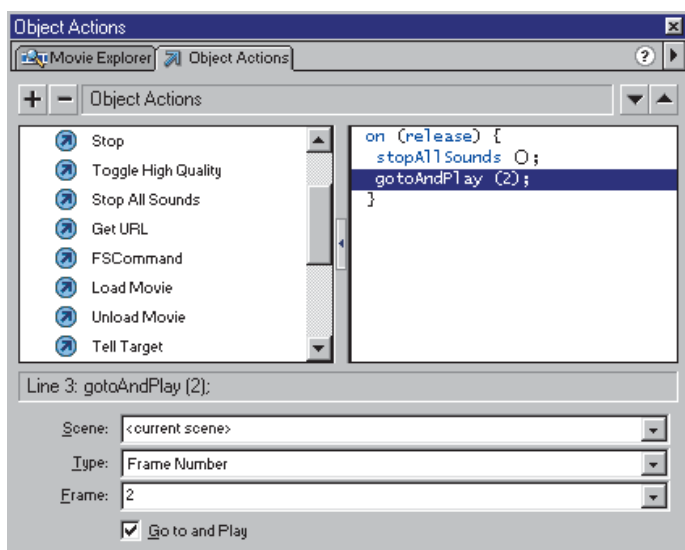
ActionScript содержит в себе возможности настоящего объектно-ориентированного программирования (ООП)

ва — список доступных в языке команд, операторов и функций. Справа — непосредственно сам код ActionScript, то есть программа. Внизу, в зависимости от выбранной строки, располагаются параметры команды.

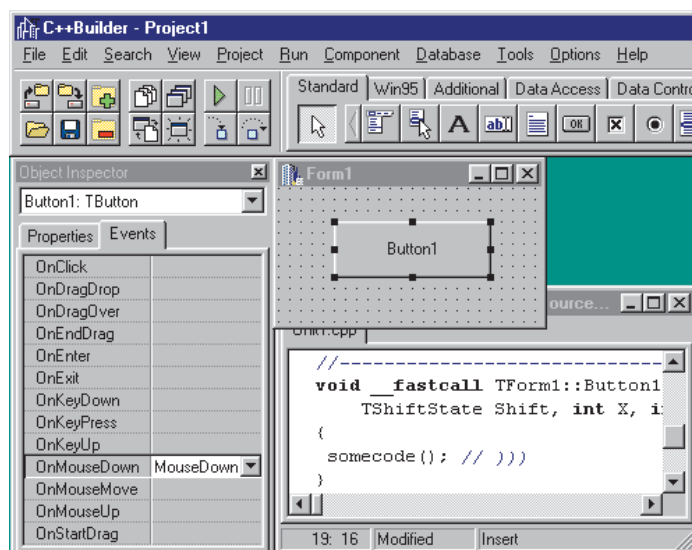
Таким образом, нужные команды выбираются из меню. Редактор автоматически создает корректный код в текстовом виде. После этого при щелчке по созданному коду параметры функции можно выбрать из выпадающих меню, а можно, если это

удобно. Режим вставки команд и операторов из меню, однако, сохраняется.

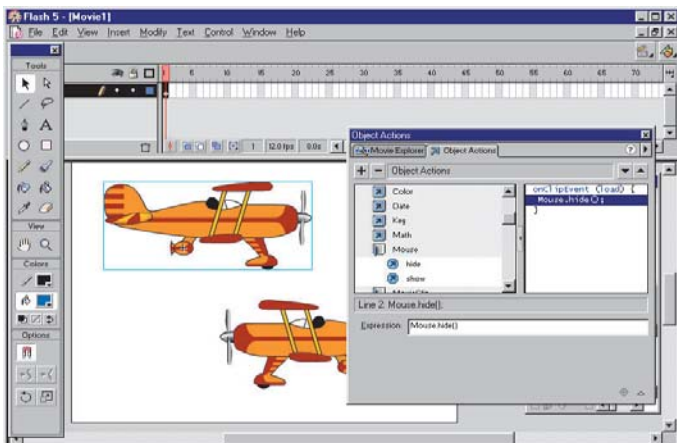
Я уже несколько раз восторженно произнес волшебное слово — «программирование». Настало время пояснить его использование применительно к Flash. При этом мне придется произнести еще несколько магических заклинаний вроде «объектно-ориентированное программирование», «условия разветвления» или «определение функций». Но все по порядку.



▲ Редактор ActionScript в режиме Normal. Слева — окно с выбором доступных команд и операторов. Справа — создаваемый код программы. Внизу — доступные параметры выделенной строки



▲ Среда программирования Borland C++ Builder. Вверху слева видны библиотеки элементов, справа — окно обработчиков событий. В центре — окно создаваемой программы, под ним — окно с исходным текстом на языке C++



▲ Macromedia Flash 5 с открытым редактором ActionScript



▲ Игра на Flash. Звездолет расстреливает летающие астероиды

» Почему программирование?

Действительно, почему? Да потому, что любой объект в ролике Flash имеет свои свойства. Я сейчас говорю о видимых разработчику объектах вроде кнопок, символов, текста или рисунка. У них есть свойства: цвет, видимость, размеры и т. д. Используя отрывки код на языке ActionScript, можно изменять свойства объектов во время проигрывания ролика. Причем эти отрывки, или Actions, могут выполняться при наступлении определенных условий, скажем, действия зрителя по ту сторону экрана. Как создать новый курсор для мышки? Очень просто. Выберите привлекательный графический объект, сделайте его кнопкой и назначьте ему следующий Action:

```
on (press) {
    Mouse.hide();
    startDrag ("");
}
```

В результате наступления определенных событий (нажатия пользователем на кнопку) программка на ActionScript прячет курсор мышки и отдает приказ перетаскивать объект за мышкой. Таким образом мы получаем произвольный курсор.

Итак, первый момент истины. Мы использовали так называемый обработчик событий (первая строка приведенного кода). Обработчики событий — это важнейшие элементы для всех современных программ с графическим интерфейсом. Фактически любая современная программа сначала ждет событий — действий пользователя, и только потом начинает действовать. На иллюстрации, изображающей среду программирования Borland C++ Builder в действии, прекрасно

виден список обработчиков событий для выбранного объекта.

Второе — мы использовали функции для выполнения определенных действий. Кроме стандартных мы можем определять свои собственные функции. В пятой версии Flash созданные пользователем функции могут получать и возвращать данные с помощью оператора return, чего нельзя было делать раньше. Прогресс налицо. Теперь для часто используемых операций можно писать подпрограммы, как в обычном языке программирования:

```
function sqr (x){
    return x*x;
}
```

Лично для меня приведенный участок кода сильно напоминает C, так как с JavaScript я работал редко. А сами разработчики из Macromedia наирают на то, что ActionScript очень близок именно к JavaScript, хотя и не является его диалектом.

Функции получают данные, обрабатывают их и изменяют. Данными могут быть как свойства объектов Flash-ролика, так и созданные нами переменные: они в ActionScript делятся на несколько типов в зависимости от данных — числовые, булевы (логические) и строковые (или текстовые переменные). Мы можем громоздить свои переменные в массивы и другие сложные структуры для хранения данных, вроде стеков и очередностей.

Спокойное течение команд может встретить на своем пути условия разветвления типа if... else... (если налево пойдешь, коня потеряешь, иначе смотри следующую строчку). Можно организовать повторяющийся

цикл команд с помощью операторов while, do...while, for и for...in. Главное, чтобы при этом ваш будущий зритель не попал в ситуацию мальчика из анекдота («Мальчик, хочешь я расскажу тебе сказку... Мальчик, хочешь я расскажу тебе сказку...»).

ООП...с!

Но это еще не все. Фактически ролик Flash уже содержит в себе ряд готовых к употреблению объектов, не визуальных, а объектов в программном смысле. К примеру, объект Mouse. Выше было показано, как используется метод этого объекта. В пятой версии Flash язык ActionScript перешел на адресацию объектов с помощью точек, как это принято в C++. Раньше использовалась слэш-адресация, как в web-адресах.

```
Mouse.show();
```

Всего имеется 14 predefined объектов. Объект Array позволяет организовывать массивы данных и работать с ними. Здесь массив данных — это не жесткая структура с заданным количеством членов, а скорее динамический список с переменной длиной. Члены массива можно добавлять или удалять после его создания.

Объект Boolean — это простая оболочка для имитации булевого типа данных с теми же возможностями, что и стандартный Boolean-объект в языке JavaScript.

Объект Color организует чтение и установку цветовых параметров клипа Flash. Объект Date позволяет читать текущую дату, время по Гринвичу или на локальном компьютере, где идет ролик. Очень интересным является объект Key: он позволяет отслежи-

» вать клавиши, которые нажимает пользователь компьютера. Математические же функции интегрированы в объекте Math.

Объект Mouse используется только для того, чтобы спрятать или показать курсор мышки, когда она находится над Flash-роликом. Объект MovieClip — это общий тип объектов для всех анимационных клипов. Точнее сказать, все созданные клипы сразу являются и объектами MovieClip. Методы и свойства этого типа объектов позволяют управлять проигрыванием клипов.

Number, как и Boolean, всего лишь оболочка для работы с числовыми данными. А вот объект под гордым названием Object — это корневой объект всех остальных объектов в ActionScript. Это, кстати, одна из самых забавных вещей в ООП: все объекты происходят от одного общего прародителя, обладающего, как одноклеточный организм, только минимумом функций и свойств.

Объект Selection работает только с выделением в редактируемом текстовом поле. А вот объект Sound куда более интересен, потому что управляет звуком в отдельно выделенном клипе. С помощью методов Sound можно менять стереобаланс звука, его громкость, ну и, само собой, управлять его проигрыванием.

Объект String управляет работой с текстовыми строками; он может поделить, добавить текст, преобразовать все в верхний регистр и т. д. Самый загадочный объект — это XML, который может загружать, обрабатывать, строить и манипулировать документами XML. Даже не просто документами, а деревьями документов (ветвящиеся структуры данных в программировании). А XMLSocket позволяет локальному компьютеру, на котором запущена анимация, связываться с удаленным сервером, идентифицируемым либо именем, либо IP-адресом.

Объекты, как и полагается, создаются с помощью конструктора, имеют свои свойства и методы. Ряд predeterminedных объектов не требует конструктора, они как бы уже созданы. На основе имеющихся объектов можно создавать новые объекты, при этом методы и свойства «родительских» объектов переходят к новым объектам. Наследование, однако. Кроме наследования, никаких особых супервозможностей ООП не используется. Нет разделения на приватные и публичные функции, нет виртуальных методов. Но не слишком ли многого я хочу? В конце концов, это же задумывалось как пакет для со-

здания интерактивной анимации, а не как конкурент монстров программостроения.

Честное слово, когда я углубился в дебри изучения ActionScript, забыл, что нахожусь в программе с дружелюбным интерфейсом, которая называется графическим редактором! Казалось, я снова вернулся на несколько лет назад, когда я был еще программистом и продирился сквозь дебри C++! Мое дежавю было вызвано очень похожим стилем работы с визуальными и программными объектами, языком и концепциями, которые использовались в Borland C++ Builder.

Десерт для программиста

Одна из особенностей Flash сегодня — это возможность вложить один анимационный клип в другой, причем неоднократно. Для программиста, пишущего компьютерные игры, это означает встроенную поддержку спрайтов. Если мы делаем анимацию с летящим самолетом, то можно либо нарисовать все на одном уровне, либо разложить на составляющие. Последнее предпочтительнее, так как использованные запчасти можно будет употребить потом на постройку других самолетов с улучшенными характеристика-

Все программы ActionScript ограничены в действиях границами Flash-ролика. Хотя в языке и есть возможности обмена информацией с удаленным сервером, компания Macromedia приложила все усилия, чтобы для локального компьютера Flash-ролики были максимально безопасными. В этом и заключается огромное преимущество технологии. Программа ActionScript как бы выполняется внутри ящика с песком, предохраняющего компьютер пользователя от взрыва. Но в этом, конечно, и недостаток. Ни тебе файл с диска пользователя прочитать, ни записать таблицу рекордов в игре.

Первым делом, первым делом...

Первым делом все-таки остается дизайн. Не обязательно использовать ActionScript, чтобы создавать крутые ролики. Не обязательно даже анимация, вполне возможно делать только статические сцены (к примеру, иллюстрации). Основное предназначение пакета Macromedia Flash — по-прежнему дизайн, хотя дизайн скорее динамический, а не статический.

Дизайнерские корни Macromedia Flash 5 позволяют создавать самый невообразимый интерфейс, который и не снился обычным программистам

ми. Мы можем сначала создать короткий мультфильм, изображающий вращающийся пропеллер. Потом нарисовать корпус самолета и добавить к нему клип с пропеллером как составную часть. Мультфильм в мультфильме — и вот уже готов сложный ролик.

Кроме удобств конструирования мы получаем дополнительную гибкость в поведении объектов. Допустим, у нас в ролике две копии анимационного клипа с самолетом. Напишем для каждого самолета программки ActionScript, проверяющие код на столкновение с другими объектами; добавим для одного из них реакцию на нажатие клавиш, два клипа с движущейся землей и облаками — и простейшая компьютерная игра готова.

Многочисленные «прибамбасы» Flash, ориентированные именно на внешнюю сторону, на визуальный интерфейс Flash-клипа, делают такие фокусы очень легкими. В отличие от традиционных сред программирования дизайнер Macromedia Flash отталкивается от графической части.

В этом Flash-художники сродни аниматорам-трехмерщикам. Только флэшеры создают мультфильмы в привычном смысле, тогда как трехмерщики все чаще переключаются на «большое кино».

Мультимедиа-редактор Flash сегодня смело может использоваться для производства самой разнообразной продукции: компьютерных игр, презентаций, электронных учебников. А ограничение действия ActionScript пределами только Flash-ролика делают пакет безопасным средством для натаскивания начинающих программистов. Мне нравится само название языка — ActionScript. Оно говорит само за себя. Больше действия, больше взаимодействия, больше жизни!

■ ■ ■ Сергей Токарев

Macromedia Flash 5

Разработчик ▶ Macromedia, Inc

Сайт разработчика ▶ www.macromedia.com

Условия распространения ▶ shareware