ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 1759408E48E0BF168748ZA088ED99D3E
Владелец: Евсюкова Наталья Юрьевна
Действителен: с 03.03.2022 до 27.05.2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»
ВОЛЖСКИЙ ФИЛИАЛ

Кафедра гуманитарных и естественнонаучных дисциплин

Методические указания к лабораторным работам
по дисциплине

FLASH ПРОГРАММИРОВАНИЕ

Направление подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль, специализация) образовательной программы

«Автоматизированные системы обработки информации и управления»

Квалификация

бакалавр

Чебоксары
2019

СОДЕРЖАНИЕ

ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ 3

ТЕМАТИКА ЛАБОРАТОРНЫХ РАБОТ 5

Лабораторная работа №1. Элементы управления Macromedia Flash MX и Dreamviewer. 5

Лабораторная работа №2.1. Этапы проектирования. 10

Лабораторная работа №2.2. Этапы проектирования. 12

Лабораторная работа №3.1. Управление слоями. Понятие слоя 17

Лабораторная работа №3.2. Управление слоями. Порядок управления слоями. 25

Лабораторная работа №3.3 Управление слоями. Операции и методы управления слоями. 26

Лабораторная работа №4.1. Язык управления сценариями Action Script. ..... Ошибка! Закладка не определена.

Лабораторная работа №4.1 Язык управления сценариями Action Script. Методы сохранения обновления и отладки сценариев 32

Лабораторная работа №4.3 Язык управления сценариями Action Script. Библиотеки сценариев. 35

ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Все лабораторные работы курса “Flash программирование” выполняются в едином порядке, в соответствии с едиными требованиями.

Порядок выполнения работы.

1. Изучить "Краткие теоретические сведения".
2. Получить у преподавателя условия задач (номер варианта).
3. Решить задачи (см. п.
4. **Порядок решения задачи.**).
5. Показать работу программ преподавателю.
6. Распечатать тексты полученных программ.
7. Оформить отчет (пояснительную записку) (см. п.
8. **Требования к отчету**).
9. Сдать отчет (пояснительную записку) преподавателю.
10. Подготовиться к ответам на контрольные вопросы.
11. Защитить работу.

Порядок решения задачи.

1. Изучить условие задачи.
2. Разработать алгоритм решения задачи.
3. Согласовать алгоритм решения задачи с ведущим преподавателем
4. Разработать порядок работы с программой.
5. Написать и ввести программу.
6. Отладить программу.
7. Скорректировать алгоритм и порядок работы программы по результатам отладки.
8. Ответить на вопросы задания (если есть).

Общие требования к программам.

- Программа должна выводить на терминал реквизиты авторов (фамилию, имя и группу).
- Программа, использующая ввод с клавиатуры, должна подсказывать пользователю, что ему делать.

Требования к отчету.

Требования к оформлению.

1. Отчет выполняется на листах формата А4 с использованием любого текстового процессора и распечатывается на принтере.
2. Титульный лист отчета выполняется по стандартной форме (см. Рисунок 1).
3. Рамка на последующих листах отчета необязательна.

4. Листы отчета необходимо скрепить.
5. Отчет должен быть подписан исполнителем.
6. Тексты программ должны содержать комментарии к использованию переменных и работе программы.
7. Тексты программ распечатываются на принтере.

Содержание отчета.

1. Титульный лист (см. Рисунок 1).
2. Цель работы.
3. Описание решений задач (см. п. 0).
4. Тексты программ (распечатки).

Порядок описания решения задачи.

Решение каждой задачи описывается в следующем порядке;

1. Условие задачи.
2. Физическое и математическое решение задачи.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»
Волжский филиал

Факультет очный
Кафедра математики и информатики
Направление подготовки Автоматизированные системы обработки информации и управления

О Т Ч Е Т

к лабораторной работе №1

«Элементы управления Macromedia FX»

Дата сдачи «__»_____20__г
Зачтено «__»_____20__г

Выполнил: студент гр. ОП-14,
Петров И.В.
Преподаватель: Семенов Б.И.

Чебоксары 2015

Рисунок 1. Пример оформления титульного листа.

ТЕМАТИКА ЛАБОРАТОРНЫХ РАБОТ

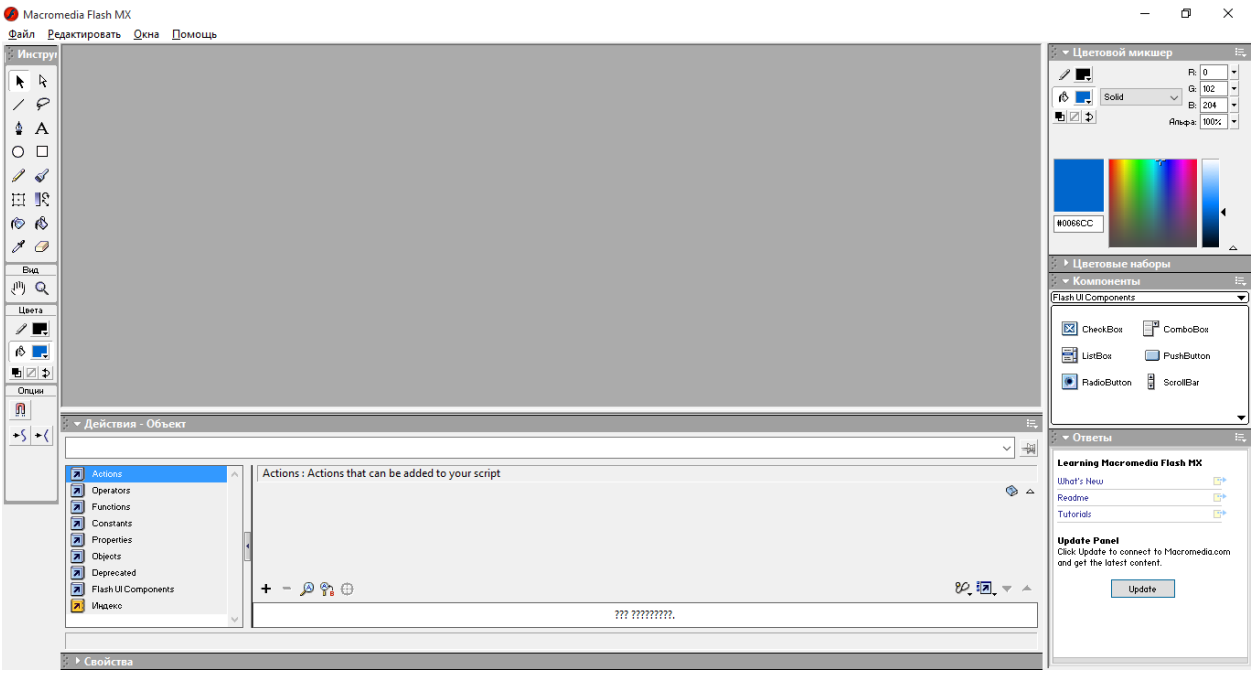
Лабораторная работа №1. Элементы управления Macromedia Flash MX и Dreamviewer.

Основные сведения

1. Понятие элемента управления

Что такое Macromedia Flash-MX

освоение базовых понятий и методов компьютерной графики; изучение графической программы «Macromedia Flash-MX». В результате обучения обучающиеся смогут получить опыт: проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; коллективной реализации информационных проектов, информационной деятельности в различных сферах, востребованных на рынке труда; эффективного применения информационных образовательных ресурсов в учебной деятельности, в том числе самообразовании; эффективной организации индивидуального информационного пространства.



1. Панель инструментов "Инструменты".
2. Временная диаграмма "Временная шкала".
3. Окно библиотеки "Библиотека"
4. Стол "Сцена".

Стол отображает кадр, выбранный в окне временной диаграммы. Работая с изображениями на Столе, вы создаете фильм.

Библиотека предназначена для хранения так называемых символов, которыми могут быть изображения, кнопки, фрагменты фильмов,

Временная диаграмма отображает процесс изменения содержания фильма во времени. Для организации содержания используются уровни.

Панель Toolbox предлагает инструменты для создания и обработки текста и графики. В нижней части панели инструментов появляются кнопки модификаторов активного (выбранного) инструмента.

Чтобы максимально расширить рабочую зону Стола, окно библиотеки, окно временной диаграммы и набор инструментов можно перетащить в любую часть экрана, даже разместив их вне окна приложения.

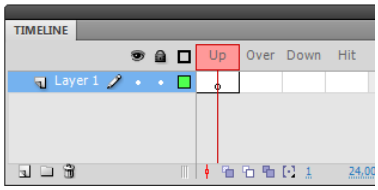
Создание кнопок

Кнопки — это экземпляры символов класса `Button`. Кнопки должны храниться в библиотеке.

Чтобы создать новую кнопку в меню Insert выберите пункт **New Symbol**… или на панели **Library** нажмите кнопку (**New Symbol**). В меню **Behavior** (Type) появившейся панели **Create New Symbol** выберите пункт **Button**. В результате откроется шаблон новой кнопки.

В рабочем поле находится только черный крестик — точка фиксации символа (начало его системы координат). Вокруг этой точки будет происходить поворот символа. Сместить ее нельзя, надо разместить относительно ее сам графический образ.

Временные диаграммы клипов и кнопок существенно отличаются. Кадры кнопок по умолчанию отображаются более крупными, их нумерация отсутствует.



Кадры временной диаграммы кнопки соответствуют ее четырем возможным состояниям:

♦ Up — стандартный вид кнопки. Объекты, размещенные на данном кадре, отображаются, если указатель мыши располагается вне кнопки (или ее активность отключена при помощи свойства enabled);

♦ Over—вид кнопки при наведении на нее указателя мыши. Этот режим показывает пользователю, что кнопка готова для нажатия. Дополнительным его признаком является смена формы курсора со стрелки на очертания руки;

♦ Down—вид кнопки при ее нажатии. Переход в данный режим осуществляется, если при нажатии левой клавиши мыши ее указатель располагался в активной зоне кнопки;

♦ Hit — особый кадр, служащий для задания активной зоны кнопки. Активная зона — это область, чувствительная к наведению курсора мыши и ее щелчку. Определяют ее объекты, расположенных на этом кадре. Сами объекты не отображаются, а только определяют фактические границ кнопки.

Активная зона и кнопка как видимый объект могут не совпадать. Однако обычно кнопка располагается в центре активной зоны. Если на кадре Hit не имеется объектов, то для формирования активной зоны используются объекты последнего заполненного кадра.

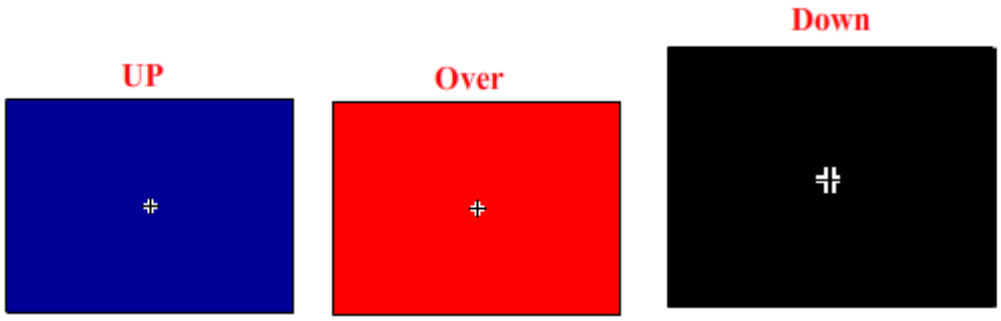
Чаще всего явно задавать активную зону необязательно. Это стоит делать, если кнопка имеет сложную форму или отверстия, а также если ее очертания в разных режимах сильно различаются. Отдельный случай— кнопки в виде текста. Для них просто необходимо задать активную зону, иначе их нажатие будет происходить лишь при "прямом попадании" курсора по букве.

Если вид кнопки во всех режимах одинаков, то создать ее можно более просто. Для этого необходимо выделить изображающую ее графическую фигуру и нажать <F8>. При этом появится панель **Convert to Symbol**. После задания в ней нужных настроек новый символ будет добавлен в библиотеку, а фигура будет преобразована в экземпляр. На временной оси этой кнопки будет заполнен только один кадр UP.

Экземпляры кнопок, подобно клипам, могут иметь имена. Прописываются они в поле Instance Name панели Properties. Для возможности работы с мастером завершения кода имена кнопок должны иметь суффикс _btn, например left_btn.

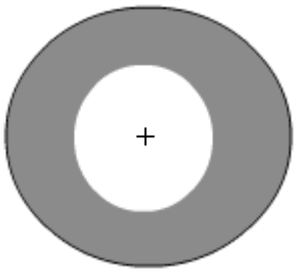
Список индивидуальных заданий

1. Ролик в формате SWF для двух кнопок из набора стандартных.
2. Ролик в формате SWF с анимированным текстом
3. Создать новый файл и в его библиотеке кнопку со следующими кадрами Up, Over и Down..
5. Создать экземпляр кнопки в рабочей области.



Протестировать кнопку во всех состояниях (`Up`, `Over`, `Down`). Какая зона является активной (`Hit`).

1. Создать в библиотеке дубликат первой кнопки. Определить для дубликата область `Hit` следующим образом



2. Добавить в кадр экземпляр второй кнопки. Протестировать ее работу. Сохранить файл под именем `Lab1_1.fla`.

Лабораторная работа №2.1. Этапы проектирования.

Цель работы: выработать практические навыки работы с Flash в Macromedia FX изучить объекты. научиться создавать, вводить в компьютер, выполнять и исправлять простейшие проекты на языке Flash в Macromedia FX в режиме диалога, познакомиться с диагностическими сообщениями программы об ошибках при выполнении роликов Flash.

Общие сведения:

Технология Flash появилась в середине 1990-х годов и быстро нашла применение в Интернете. Она позволила решить острую на тот момент проблему кроссбраузерной и кроссплатформенной совместимости. Это достигалось путем встраивания в браузер специальной программы (плагина), называемой — флеш-плейером.

Так как дизайн веб-страниц стал приобретать первостепенное значение, то флеш-анимаций стали эффектным и эффективным инструментом его реализации. Флеш-ролики стали активно применяться для повышения интерактивности пользовательского интерфейса, для создания броской рекламы, для разработки элементов обучающих программ и т.п..

Динамичность и интерактивность графики в технологии Flash обеспечивается за счет использования объектно-ориентированного языка программирования ActionScript, основанного на стандарте ECMAScript 262 (другой популярный его диалект — Javascript). Вместе с объектной моделью документа и продуманной событийной архитектурой этот язык программирования обеспечивает высокую интерактивность пользовательского интерфейса.

В последние годы стало распространяться мнение, что актуальность flash-технологии осталась в прошлом. Это связано, в первую очередь, с отказом от поддержки проигрывания flash-роликов на некоторых мобильных платформах. Возникает вопрос об актуальности предлагаемого учебного пособия.

В роли альтернативы флеш-технологии выступает совокупность спецификаций, объединенных понятием HTML 5. Действительно, в них появились возможности для реализации простых анимаций, которые реализуется стандартными средствами браузеров. Например, некоторыми стилевыми свойствами CSS3.

Однако, на современном этапе (и на ближайшую перспективу) выразительные возможности новых технологий заметно уступают Flash, и в создании сложных анимаций конкурировать не могут. Нужно отметить также, что компания Adobe — владелец технологии Flash, разработала для

создания анимаций, реализуемых средствами HTML 5, новый инструмент Adobe Edge, который в основном повторяет среду разработки Adobe Flash.

Пособие построено так, что может применяться не только бакалаврами и магистрантами в рамках основного учебного процесса, но и для самостоятельного изучения технологии.

Свойства кнопок и фрагментов ролика (MovieClip)

а) **_enabled**

За активность кнопки отвечает свойство **enabled**. Если оно равно true,

```
left_btn._enabled=true;
```

то кнопка реагирует на мышку (принятое по умолчанию значение). Если же ему присвоить `false`, кнопка станет неактивной.

Это означает следующее.

- При наведении на кнопку указателя мыши она не будет переходить в состояние `Over`. Аналогично, при попытке нажать кнопку перехода в режимDown не будет.
- При попадании в активную зону кнопки вид указателя мыши не будет меняться со стрелки на руку.
- Событий кнопки происходить не будет (кроме `onSetFocus` и `onKillFocus`).
- Если отключенная кнопка располагается над активной, то последняя будет способна к восприятию событий.

б) **_visible**

Будет ли кнопки видна или нет определяет свойство `visible`. Если оно равно `true`,

```
left_btn._visible =true;
```

то кнопка видна (принятое по умолчанию значение). Если же ему присвоить `false`, кнопка станет невидимой.

Координаты и размеры символов

Координаты символов определяются значениями свойств `_x` и `_y`. Для символов, которые находятся в кадре основной временной оси, `_x` — расстояние по горизонтали от левого верхнего угла кадра и `_y` — расстояние по вертикали от левого верхнего угла кадра в пикселах .

Если символ входит в другой символ (родительский), то координаты дочернего символа определяются в системе координат родительского символа. Для перемещения символа `m_mc` в точку с координатами `(200;125)` надо выполнить следующий код

```
m_mc._x=200;
m_mc._y=125;
```

Размеры символов определяются значениями свойств **_height** (высота) и **_width** (ширина) в пикселах. Следующий код сделает размеры символа m_mc 100x200 пикселов

```
m_mc._ height=100;
m_mc._ width =200;
```

Изменение размера можно задавать и в процентах к первоначальному значению. Чтобы увеличить высоту на 20% и ширину на 10%, надо выполнить следующий код

```
m_mc._ xscale=110;
m_mc._ yscale+=20;
```

Список индивидуальных заданий

1. Создайте новый документ и назовите его слой "Небо". Сохраните файл под именем "Lab1_2.fla".
2. Создайте символ звезды (тип — MovieClip). Экземпляр звезды назовите s_mc. Создайте луну как символ.
3. В кадре создайте "небо" с одной звездой, которой будем управлять с помощью кнопок.
4. Создайте слой ”Кнопки”. Для размещения кнопок создайте "пульт управления" в виде серого прямоугольника в нижней части рабочего поля. Создайте и запрограммируйте серию кнопки для управления звездой **s_mc**.

Лабораторная работа №2.2. Этапы проектирования.

Цель работы: выработать практические навыки работы с ASP.NET, научиться создавать, вводить в компьютер, выполнять и исправлять простейшие программы в режиме диалога, познакомиться с

диагностическими сообщениями компилятора об ошибках при выполнении программ.

Общие сведения:

Методы управления проигрыванием флеша

По умолчанию флеш автоматически проигрываться начиная с первого кадра. Но иногда требуется запускать или прерывать выполнение анимации по требованию пользователя, просматривать по кадрам или вернуть на несколько кадров назад. Такой режим используется, в частности, при создании флеш-презентации.

Методы клипа (`movie clip`), позволяющие управлять проигрыванием его анимации:

`play()` — начинает или возобновляет воспроизведение анимации клипа;

`stop()` — останавливает воспроизведение анимации клипа;

`gotoAndPlay()` — переходит на определенный кадр и продолжает воспроизведение;

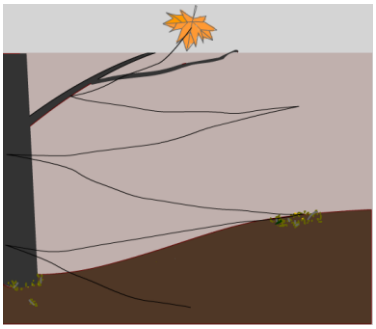
`gotoAndStop()` — переходит на определенный кадр и останавливает воспроизведение;

`nextFrame()` — переходит на следующий кадр и останавливает воспроизведение;

`prevFrame()` — переходит на предыдущий кадр и останавливает воспроизведение.

Список индивидуальных заданий

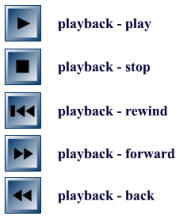
1. Создать новый документ и в нем анимацию падения листа по заданной траектории.



2. Создать новый слой ”`Actions`”. Остановить автоматическое проигрывание анимации. Для этого создать сценарий первого кадра слоя ”`Actions`”. В сценарии всего одна команда — **`stop();`**

Для размещения кода создать новый слой "Actions", выделить его первый кадр и записать в поле кода панели Action комманду. Слой "Actions" должен быть самым первым.

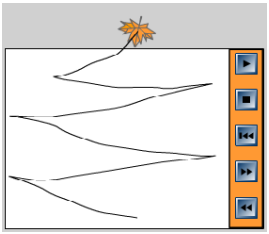
2. Чтобы анимация проигрывалась, надо создать кнопку, запускающую проигрывание. В общей библиотеке кнопок Flash в папке Playback есть кнопки, имеющие стандартные обозначения. Имена кнопок говорят о их назначении.



Последние две кнопки осуществляют переход на один кадр вперед и назад.

Кнопки предлагаются без кода, т.е. их действие не запрограммировано!

Создайте новый слой "Кнопки". Разместите кнопки в рабочей области нового слоя. Т.к. кнопки, "болтающиеся" в пространстве не очень хорошо выглядят, подложите под них "пульт" — прямоугольник.



3. Теперь надо кнопки запрограммировать. При их освобождении должны выполняться простые комманды:

```
play();
stop();
gotoAndPlay(1);
nextFrame();
prevFrame();
```

4. Можно дорисовать осенний пейзаж, сделав сероватый фон, коричневую землю и дерево на заднем плане (на каком слое его надо рисовать?).

5. Протестируйте флеш.

6. Обозначения на кнопках стандартные, но, возможно, не все их знают. Поэтому сделайте всплывающие подсказки.

При наведении курсора мышки на кноки рядом с ними должны появляться надписи "Play", "Stop", "Rewind", "Next frame", "Prev frame".

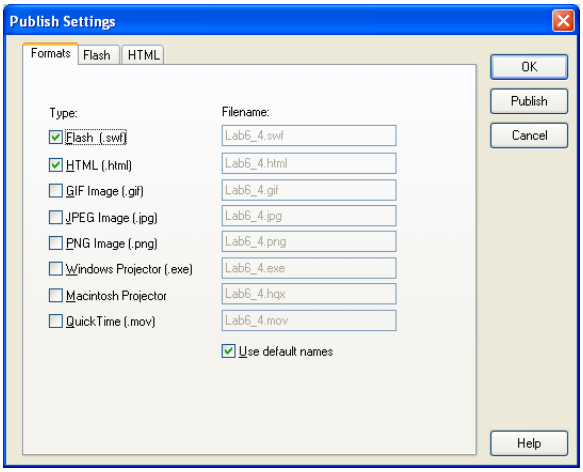
Для создания всплывающей подсказки:

- двойным щелчком левой кнопкой мышки на кнопке войдите в режим редактирования,
- в кадре "Over" добавьте надпись в нужном месте,
- выйти из режима редактирования.

7. Протестируйте флеш.

Встраивание ролика в html-страницу

В меню File выберите пункт Publish Settings…



В окне Publish Settings отметьте флажки Flash и HTML.

а) Нажмите кнопку Publish. б) Закройте окно Publish Settings. в) Просмотрите созданную html-страницу.

9. Откройте созданную html-страницу в любом простом текстовом редакторе (Блокнот, Far или др.).

```
<HTML>
<HEAD>
<meta http-equiv=Content-Type content="text/html;
charset=">
<TITLE>Lab7_4</TITLE>
</HEAD>
<BODY bgcolor="#FFFFFF">
<!-- URL's used in the movie-->
<!-- text used in the movie-->
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
```

```
codebase="http://download.macromedia.com/pub/shockwave
/cabs/flash/swflash.cab#version=6,0,0,0"
 WIDTH="550" HEIGHT="400" id="Lab7_4" ALIGN="">
<PARAM NAME=movie VALUE="Lab7_4.swf"> <PARAM
NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED src="Lab7_4.swf" quality=high bgcolor=#FFFFFF
WIDTH="550" HEIGHT="400" NAME="Lab7_4" ALIGN=""
TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/go/getflashplay
er"></EMBED>
</OBJECT>
</BODY> </HTML>
```

Изучите, как флеш вставляется в веб-страницу.

1. Вставьте флеш задания №1 в html-страницу.
2. Вставьте на страничке перед флешем несколько строчек, например:

```
<H1>Управление выполнением флеша  </H1>  <br>
<H3> Выполнил студент(ы) Фамилия Имя </H3>
<br> <br> <br>
```

3. Чтобы вокруг флеша была рамка, после <OBJECT > вставьте

```
border = 1
```

4. Сохраните страничку и просмотрите ее.

Лабораторная работа №3.1. Управление слоями. Понятие слоя

Цель работы: научиться правильно использовать слои. Получить навыки управление слоями.

Общие сведения.

Временная шкала

Временная шкала - основной инструмент при работе с анимацией во Flash. На ней отображается информация о слоях, о том какие кадры являются ключевыми, а какие генерирует Flash. С помощью временной шкалы можно понять, какие кадры содержат действия или метки. Она позволяет перемещать ключевые кадры и целые куски анимации.

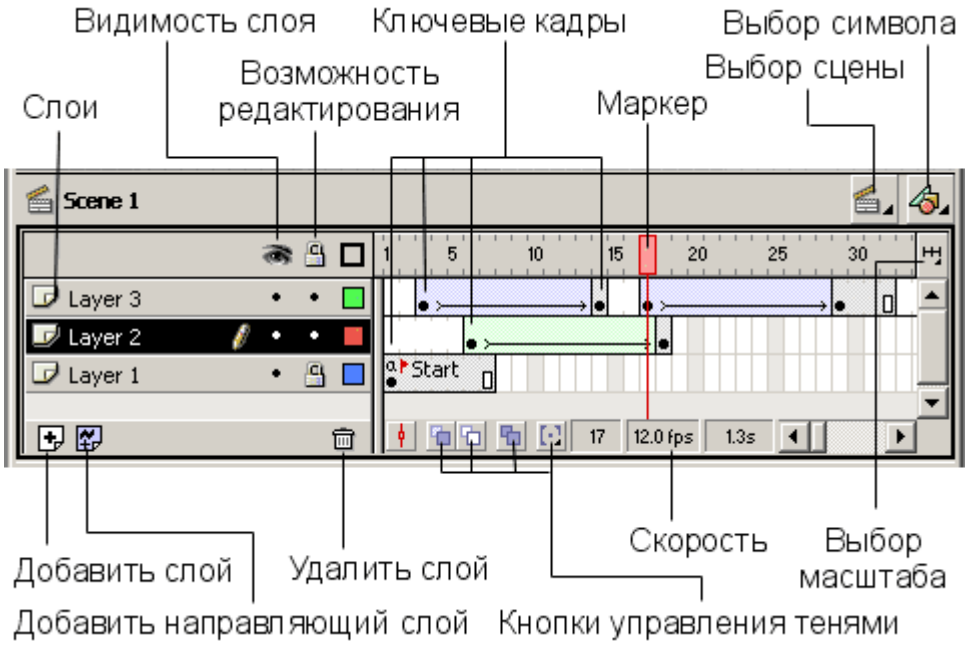


Рис.1_2 - Временная шкала

Маркер - указывает на текущий кадр, отображаемый в окне. При клике на какой-либо кадр, маркер автоматически перемещается на него.

Слои - слева находится перечень слоев. Под ним существуют кнопки, позволяющие добавлять и удалять слои. Каждый слой можно сделать невидимым и запретить его для редактирования.

Шкала кадров - поле, где вы можете добавлять и удалять простые и ключевые кадры. Если вызвать контекстное меню (нажать на правую клавишу мыши) на каком-либо кадре, вы увидите перечень действий,

которые можно совершить. На шкале отображается информация о кадрах, которые являются ключевыми (такие кадры помечаются черными кружочками), содержат действия (буковка "a" над кружочком) или метку (красный флажок, после которого идет название метки). Цвет тоже говорит о типе кадров. Серый цвет - это кадры, которые в точности повторяют ключевой кадр (Keyframe). Синеватая или зеленоватая подсветка говорит о том, что кадры сгенерированы Flash. Белое или "пустое" полосатое пространство говорят о том, что на этих кадрах ничего нет.

Кнопки управления тенями - это кнопки, позволяющие отображать соседние кадры как бы через кальку, чтобы видеть разницу между предыдущими и последующими кадрами. Можно задавать глубину такого отображения по обе стороны от маркера.

Перекрывание фигур во Flash

Когда вы проводите линию поперек другой линии или закрашенной фигуры инструментами Pencil, Line, Oval, Rectangle или Brush, накладывающиеся линии разделяются на сегменты (сегментируются) в точках пересечения (рис. 3.2). С помощью инструмента Arrow (Стрелка) можно выделить и переместить или изменить каждый сегмент отдельно от других.

Рис. 3.2. Разделение фигуры на сегменты

Пересекающиеся линии, созданные инструментом Реп, не сегментируются в точках пересечения, но остаются неизмененными. Об этом рассказано в разделе "Использование инструмента Реп" данной главы.

Когда вы закрашиваете верхнюю часть фигуры или линии, то нижняя часть принимает тот же цвет, что и верхняя. Области, закрашенные одинаковым цветом, сливаются имеете; различно закрашенные области остаются разделенными. Можно использовать эти особенности, чтобы создавать, маски, высечки и другие инверсные изображения.

Например, весечка, приведенная на рис. 3.3, создана перемещением несгруппированного изображения бумажною змея на зеленую форму, отменой выделения бумажного змея и затем перемещением закрашенной части бумажного змея за пределы зеленой фигуры.

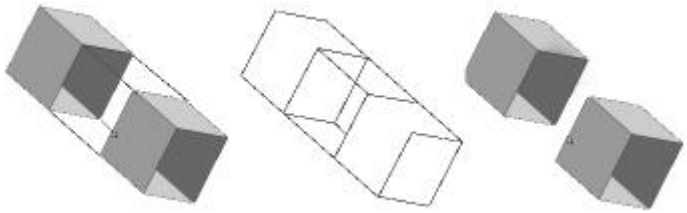


Рис. 3.3. Высечка

Во избежание случайного изменения фигур и линий при наложении, группируйте фигуры или используйте слои, чтобы разделить их.

Линии и фигуры рисуются во Flash инструментом Pencil (Карандаш) почти так же, как это делается реальным карандашом. Чтобы применить к линиям и фигурам скругление или выпрямление, выберите необходимый режим рисования для инструмента Pencil.

Чтобы рисовать инструментом Pencil (Карандаш):

1. Выберите инструмент Pencil (Карандаш).
2. Выберите цвет контура, толщину и стиль линии.
3. В нижней части панели Tools (рис. 3.4) выберите режим рисования:

- Straighten (Спрямление), чтобы рисовать прямые линии и преобразовывать неточно нарисованные треугольники, овалы, круги и прямоугольники в правильные геометрические фигуры;
- Smooth (Сгладить), чтобы рисовать гладкие изогнутые линии;
- Ink (Чернила), чтобы рисовать произвольные линии без автоматической их модификации.

- 4. Перейдите на Стол, чтобы рисовать инструментом Pencil (Карандаш), Нажав и удерживая клавишу <Shift>, вы сможете рисовать строго вертикальные или строго горизонтальные линии.

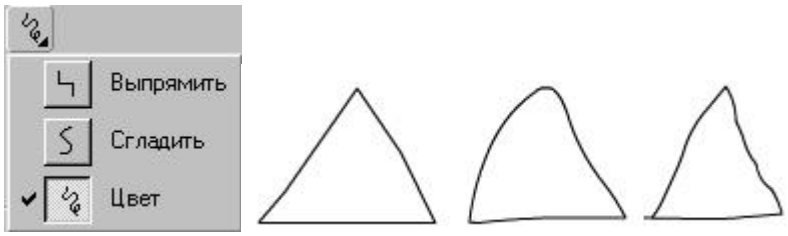


Рис. 3.4. Выбор режима рисования

Рисование прямых линий, овалов и прямоугольников

Инструменты Line (Линия), Oval (Овал) и Rectangle (Прямоугольник) позволяют рисовать соответствующие геометрические фигуры. Инструменты Rectangle и Oval создают оконтуренные и заполненные фигуры. С помощью инструмента Rectangle можно создавать прямоугольники с прямыми или скругленными углами.

Чтобы нарисовать прямую линию, овал или прямоугольник:

1. Выберите инструменты Line, Oval или Rectangle

Рис. 3.5. Инструменты Line, Oval и Rectangle

2. Выберите атрибуты для контуров и закрашивания.
3. Для инструмента Line атрибуты закрашивания недоступны.

4. Для инструмента Rectangle определите скругление углов, щелкну и на модификаторе Round Rectangle (Округление углов) и указав значение радиуса скругления. Значение "О" (ноль) задает прямые углы.

5. Перейдите на Стол. Если вы работаете с инструментом Rectangle, нажмите клавишу <Up> или <Down>, чтобы отрегулировать радиус скругленных углов. Нажмите и удерживайте клавишу <Shift>, чтобы нарисовать инструментами Oval или Rectangle круг или квадрат, соответственно. Работая инструментом Line, нажмите и удерживайте клавишу Shift, что- бы рисовать линии под углами, кратными 45 градусам.

Использование инструмента Pen

Инструмент Pen (Перо) позволяет рисовать точные траектории как прямые или сглаженные линии. Можно создавать прямые или изогнутые сегменты линий и регулировать величину угла, длину прямых сегментов и наклон изогнутых сегментов. Рисуя инструментом Реп, можно щелчком мыши создавать точки на прямых линиях, а щелкая и перетаскивая - точки на изогнутых линиях. С помощью таких точек регулируются прямые и изогнутые сегменты линии, причем можно преобразовывать кривые линии в прямые, и наоборот. Таким же образом можно создавать точки па линиях, созданных другими инструментами рисования Flash: Pencil (Карандаш), Brush (Кисть), Line (Линия), Oval (Овал) или Rectangle (Прямоугольник) и регулировать эти линии.

Настройка параметров инструмента Pen:

Можно настроить вид указателя инструмента Pen для просмотра сегментов линий в ходе рисования или для появления выделенных узловых точек. Выделенные сегменты линии и узловые точки отображаются цветом контура того слоя, на котором находятся эти линии и точки.

Чтобы установить параметры инструмента Pen:

1. Выберите команду Edit (Правка) Þ Preferences (Параметры) и перейдите на вкладку Editing (Редактирование).

2. Установите для инструмента Pen следующие настройки:

- Show Pen Preview (Предварительный просмотр инструмента Pen) - позволяет предварительно увидеть сегментирование линии. Flash отображает возможное сегментирование линии по мере того, как вы перемещаете указатель по Столу, и до того, как укажете щелчком точку ' сегментирования. Если эта настройка не выбрана, то Flash не показывает сегмент до тех пор, пока точка сегментирования не будет установлена;
- Show Solid Points (Показать основные точки) - при этом невыделен ные узловые точки будут отображаться как жирные точки, а выбран ные узловые точки - как полые точки. Эта настройка указана по умолчанию. Отмените данную настройку, чтобы гтказывать невыде ленные узловые точки как полые точки, а выбранные узловые точки как жирные точки;
- Show Precise Cursors (Показать курсор точно) - при этом указатель инструмента Pen отображается в виде перекрестья для более точного размещения линий. Отмените эту настройку, чтобы отображать для инструмента Pen установленный по умолчанию значок.

Нажатие клавиши <Caps Lock> позволяет переключаться между курсорами. 3. Нажмите кнопку ОК.

Рисование прямых линий инструментом Pen

Чтобы нарисовать инструментом Pen (Перо) ломаную линию, создайте узловые точки на линии, которые определят длину каждого ее отрезка.

Чтобы нарисовать прямую линию инструментом Pen (Перо):

1. Выберите инструмент Pen.
2. Выберите атрибуты сглаживания и заполнения.

3. Поместите указатель на Столе в том месте, где будет начинаться прямая линия, и щелчком установите первую узловую точку.

4. Щелкните снова там, где хотите завершить первый отрезок прямой линии. Нажмите и удерживайте клавишу <Shift>, чтобы рисовать линии под углами, кратными 45 градусам.

5. Таким же образом - указывая щелчками узловые точки - создайте следующие отрезки (рис. 3.6).

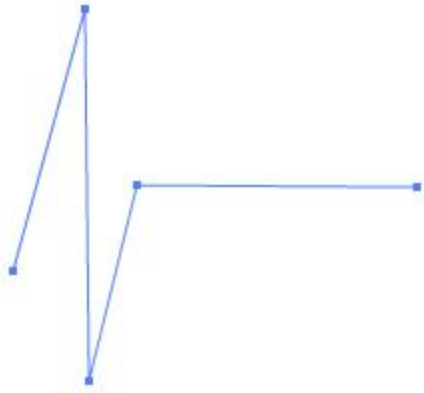
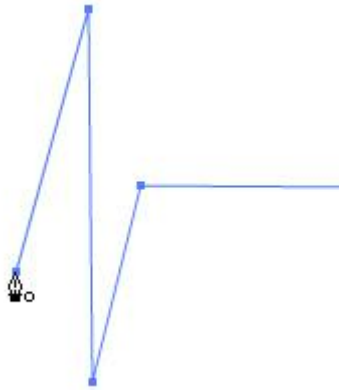


Рис. 3-6. Ломаная линия, созданная инструментом Pen

6. Чтобы закончить линию как замкнутый или незамкнутый контур, выполните один из следующих шагов:

- чтобы закончить линию, не замыкая ее, дважды щелкните на последней точке или на инструменте Pen (Перо) в панели Tools или щелкните, нажав и удерживая клавишу <Ctrl> (Windows) или <Cmd> (Macintosh), где-нибудь вне линии;
- чтобы замкнуть линию, наведите инструмент Pen (Перо) на первую узловую точку. Если вы все сделали правильно, то при этом рядом с указателем появится маленький круг (рис. 3.7). Щелкните на точке или потяните ее, чтобы замкнуть линию;



чтобы завершить линию в том виде, как она есть, выберите команду Edit (Правка) Þ Deselect All (Отменить все выделенное) или выберите иной инструмент в панели Tools.

Рис. 3.7. Чтобы замкнуть линию, наведите инструмент Pen на первую узловую точку

Рисование кривых инструментом Pen

Кривую линию можно нарисовать, перемещая инструмент Pen (Перо) в нужном направлении, создать первую узловую точку, а затем создать вторую узловую точку.

При рисовании инструментом Pen (Перо) изогнутого отрезка узловые точки представляются в виде линейных тангенс-маркеров. Наклон и длина каждого тангенс-маркера определяет наклон и высоту (или глубину) кривой. Перемещение тангенс-маркера изменяет вид кривой.

1. Выберите инструмент Pen (Перо).
2. Укажите инструментом Pen (Перо) на Столе начальную точку кривой, нажав и удерживая кнопку мыши. При этом появится первая узловая точка, и курсор примет вид стрелки.
3. Протяните курсор туда, куда хотите направить кривую. По мере его движения появится и будет динамически изменяться тангенс-маркер. Нажмите и удерживайте клавишу <Shift>, чтобы создать угол, кратный 45 градусам.

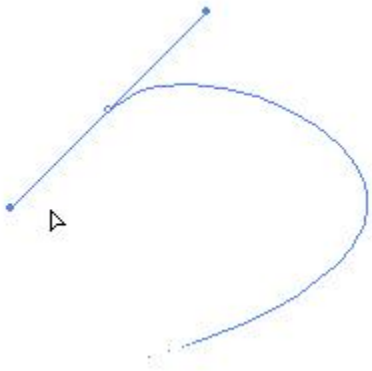


Рис. 3.8. Кривая, созданная при помощи инструмента Pen

4. Отпустите кнопку мыши.

5. Длина и наклон тангенс-маркера определяют форму сегмента кривой. Можно переместить тангенс-маркер, чтобы изменить кривую.

6. Поместите указатель там, где хотите завершить сегмент кривой, нажмите кнопку мыши и протяните курсор в противоположном направлении, чтобы завершить сегмент. Нажмите и удерживайте клавишу, чтобы создать угол, кратный 45 градусам.

7. Чтобы нарисовать следующий сегмент кривой, поместите указатель там" где хотите завершить следующий сегмент.

Контрольные вопросы.

1. Методы клипа (`movie clip`), позволяющие управлять проигрыванием его анимации.
2. Создание всплывающихъ подсказок.
3. Как встроить флеш в html-страничку?
4. Управление проигрыванием основного ролика.
5. Управление проигрыванием анимации символа типа `MovieClip`.
6. Создание сценария кадра.
7. Вопросы по коду заданий.

Лабораторная работа №3.2. Управление слоями. Порядок управления слоями.

Цель работы: закрепить практические навыки работы со слоями, научиться правильно использовать различные способы внедрения текста в разные слои, научиться составлять таймлайн и проводить операции с кодом .

Общие сведения.

Чтобы создавать текст в программе Flash, выберите инструмент ***Text Tool*** (Текст) щелкните на сцене и введите текст. Выберите инструмент ***Text Tool*** (Текст) (T).

На панели ***Properties*** (Свойства) выберите из открывающегося списка ***Text Type*** (Тип текста) пункт ***Static Text*** (Статический текст). Если выбрать пункт ***Dynamic Text*** (Динамический текст), то текст не появится в клипе. (Изменения динамического текста основываются на языке ActionScript. С динамическим текстом мы будем работать в уроке 10.)

Щелкните кнопкой мыши на сцене. Введите текст.

Выберите в меню команду ***Edit ♦ Select All*** (Правка ♦ Выделить все), чтобы выделить весь текст. На панели ***Properties*** (Свойства) выберите из открывающегося списка ***Font*** (Шрифт). Щелкните на значке ***Edit Format Options*** (Правка параметров формата), где вы сможите отформатировать текст.

Лабораторная работа №3.3 Управление слоями. Операции и методы управления слоями.

Цель работы: научиться правильно описывать операции управления слоями сохранять их и проект; научится генерировать сценарии роликов из исходного кода.

Общие сведения:

Рисование овалов

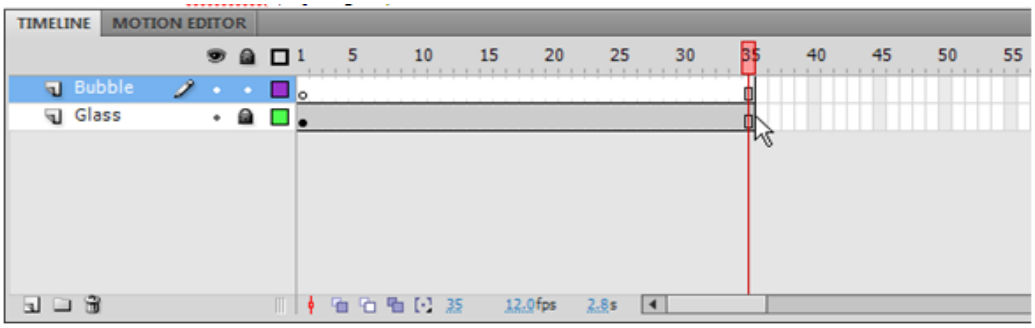
Инструмент ***Oval Tool*** (Овал) подобен инструменту *Rectangle Tool* (Прямоугольник) кроме того, конечно, что он рисует овалы. Вы используете инструмент ***Oval Tool*** (Овал), чтобы нарисовать пузырек в стакане для питья.

Чтобы вы могли легко анимировать пузырек позже, вы создадите это в новом слое.

Заблокируйте слой ***Glass***, чтобы случайно не изменить его во время работы с пузырьком.

Выберите команду меню ***Insert ♦ Timeline ♦ Layer*** (Вставить ♦ Шкала времени ♦ Слой), чтобы добавить слой для пузырька.

Назовите слой *Bubble* (Пузырек).



Выберите инструмент ***Oval Tool*** (Овал), который сгруппирован с инструментом ***Rectangle Tool*** (Прямоугольник).

Удерживайте нажатой клавишу ***Shift*** и нарисуйте кружок в стакане. Если вы нажимаете клавишу ***Shift***, инструмент ***Oval Tool*** (Овал) рисует правильный круг.

Протащите инструмент ***Selection Tool*** (Выделение) вокруг всего чтобы выбрать как заливку, так и обводку. В панели ***Properties*** (Свойства) установите настройки Stroke Color (Цвет обводки) в значение ***No Color*** (Без цвета) и ***Fill Color*** (Цвет заливки) - белый (#FFFFFF).

Выберите инструмент ***Free Transform Tool*** (Произвольная трансформация) чтобы изменить размеры пузырька.

Удерживайте нажатой клавишу ***Shift***, чтобы сохранить пропорции пузырька, и перетащите маркер угла внутрь, чтобы сделать пузырек меньше. Измененный в размерах кружок должен быть приблизительно 8 пикселов в ширину и 8 пикселов в высоту. Вы можете найти, что проще изменить размеры пузырька, если увеличить масштаб его отображения. (Панель ***Properties*** (Свойства) отображает высоту и ширину пузырька.)

По умолчанию объект масштабируется относительно левого верхнего угла. Если вы нажимаете клавишу Alt (Windows) или Option (Mac OS), объект масштабируется относительно его точки трансформации, представленной значком круга. Вы можете перетащить точку трансформации куда-нибудь в объекте или даже вне объекта.

Создание простой анимации

Более поздние уроки освещают анимацию более подробно, но для этого проекта вы будете использовать приращение движения, чтобы создать простую покадровую анимацию. Вы преобразуете пузырек в эталон, примените приращение движения и затем скопируете слой, чтобы создать дополнительные анимированные пузырьки.

Преобразование фигуры в эталон

Эталон — графический символ, кнопка или киноклип, который вы создаете в программе Flash один раз и затем используете столько раз, сколько вам нравится, сокращая размер вашего файла кино. Кроме того, существуют приемы, в которых вы можете использовать эталоны и не можете использовать другие фигуры. Вы узнаете больше об эталонах в уроке 3.

Теперь вы преобразуете пузырек в эталон, чтобы можно было его анимировать.

С помощью инструмента ***Selection Tool*** (Выделение) выделите пузырек.

Выберите команду меню ***Modify ♦ Convert to Symbol*** (Модификация ♦ Преобразовать в эталон).

Назовите эталон ***bubble*** и включите переключатель ***Graphic*** (Графический) для выбора типа символа. Щелкните мышью на кнопке ОК, чтобы закрыть диалог ***Convert to Symbol*** (Преобразовать в эталон).

Анимация объекта

Пузырек должен подниматься в напитке и изменять размеры на протяжении 20 кадров. Вы определите конечный размер пузырька и положение в кадре 20, а затем примените расчетную анимацию движения. Расчетная анимация движения автоматически создает кадры между началом и концом анимации.

Выберите пузырек на сцене и переместите его в нижнюю половину стакана.

Выберите кадр 20 в слое ***Bubble*** и нажмите клавишу F6, чтобы вставить ключевой кадр.

Выберите пузырек на сцене.

В панели ***Properties*** (Свойства) введите следующие значения:

H (Высота): 12 пикселов.

W (Ширина): 12 пикселов.

Установите значение Y на 90 пикселов ниже его текущего значения. Например, если текущее значение Y — 100, замените его на 10. Нажмите клавишу Enter или Return, чтобы применить значение.

Больший пузырек перемещается к вершине стакана.

Выберите любой кадр между 1 и 20 в слое ***Bubble***.

Щелкните правой кнопкой на выбранном кадре и выберите из выпадающего меню ***Create Classic Teween***.

Чтобы увидеть анимацию пузырька, выберите команду меню ***Control*** ♦ ***Test Movie*** (Управление ♦ Тестирование фильма). Пузырек всплывает к вершине стакана и выходит из него, а затем анимация повторяется.

Режимы рисования программы Flash

Программа Flash предоставляет три режима рисования, которые определяют, как объекты взаимодействуют друг с другом на сцене и как вы можете их редактировать. По умолчанию программа Flash использует для рисования режим слияния, но вы можете включить режим объекта или использовать инструменты ***Rectangle Primitive Tool*** (Прямоугольный примитив) или ***Oval Primitive Tool*** (Овальный примитив), чтобы использовать режим примитивов.

Режим слияния

В этом режиме программа Flash соединяет нарисованные фигуры, такие как прямоугольники и овалы, где они накладываются так, что множество фигур кажется единственной фигурой. Если вы перемещаете или удаляете фигуру, которая была объединена с другой, перекрывающаяся часть всегда удаляется.

Режим объектов

В этом режиме программа Flash не объединяет нарисованные объекты; они остаются отдельными и разделенными, даже когда перекрываются. Чтобы включить рисовальный режим объектов, выберите инструмент, который вы хотите использовать, а затем щелкните мышью на значке ***Object Drawing*** (Рисование объектов) в области настроек панели инструментов.

Чтобы преобразовать объект в фигуры (режим слияния), выделите его и нажмите комбинацию клавиш ctrl+ B (Windows) или Command+B (Mac OS). Чтобы преобразовать фигуру в объект (режим объекта), выделите его и выберите команду меню ***Modify ♦ Combine Objects ♦ Union*** (Модификация ♦ Объединить объекты ♦ Слияние).

Режим примитивов

Режим примитивов является новым для программы Flash CS3. Если вы используете инструменты ***Rectangle Primitive Tool*** (Прямоугольный примитив) или ***Oval Primitive Tool*** (Овальный примитив), программа Flash рисует формы как отдельные объекты. В отличие от обычного объекта, однако, вы можете изменить радиус углов прямоугольных примитивов, а также начальный и конечный углы и внутренний радиус овальных примитивов, используя панели ***Properties*** (Свойства).

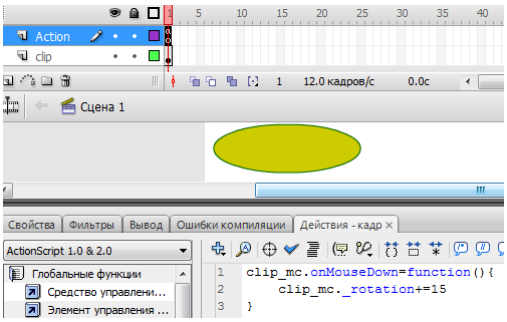
Лабораторная работа №4.1 Язык управления сценариями Action Script. Методы сохранения обновления и отладки сценариев.

Цель работы: Научиться обрабатывать события; выработать навыки работы со сценариями ActionScript 2.0.

Общие сведения

Методы обработки событий ActionScript 2.0

Хотя ActionScript 2.0 и поддерживает сценарии клипов и кнопок, но предпочтительнее создавать сценарии кадров, которые помогают централизовать код. Можно собрать весь код обработчиков событий кнопок и клипов в одном сценарии ***ключевого*** кадра. Обработчики событий объекта в сценарии кадра создаются как метобы объекта.



Методы обработки событий ActionScript 2.0 и обработчики ActionScript 1.0

Метод ActionScript 2.0	Обработчик ActionScript 1.0	Применение для клипов	Применение для кнопок
`onPress`	`on(press)`	Да	да
`onRelease`	`on(release)`	Да	да
`onReleaseOutside`	`on(releaseOutside)`	Да	да
`onRollOver`	`on(rollOver)`	Да	да
`onRollOut`	`on(rollOut)`	Да	да
`onDragOver`	`on(dragOver)`	Да	да
`onDragOut`	`on(dragOut)`	Да	да
`onKeyDown`	`on(keyPress "Name")` `onClipEvent(keyDown)`	да да	да нет
`onKeyUp`	`onClipEvent(keyUp)`	Да	нет

onSetFocus	Нет	Да	нет
onKillFocus	Нет	Да	нет
onLoad	onClipEvent(load)	Да	нет
onUnload	onClipEvent(unload)	Да	нет
onEnterFrame	onClipEvent(enterFrame)	Да	нет
onMouseDown	onClipEvent(mouseDown)	Да	нет
onMouseUp	onClipEvent(mouseUp)	Да	нет
onMouseMove	onClipEvent(mouseMove)	Да	нет

Создание методов-обработчиков

Обработчик чаще всего является анонимной функцией, ссылка на которую присваивается методу клипа.

```
clip_mc.onPress = function() {
this._x+=10; //при нажатии клип перемещается горизонтально
}
```

Можно создать именованную функцию и присвоить ее имя методу-обработчику.

```
function horizont _move() { this._x+=10;    }
clip_mc.onPress = horizont _move;
```

Функция-обработчик может ссылаться на вызвавший ее объект через ссылку this. Если именованная функция вызывается непосредственно, то this будет указывать на хозяина временной оси, но которой находится сценарий кадра.

Абсолютная и относительная адресации в сценарии кадра используются так же, как и в сценарии кнопки. Обработчики, как и остальной код, желательно размещать на первом кадре основной временной шкалы.

Когда обработчик события выполнит свою задачу и станет не нужен, его можно удалить помощи оператора `delete`:

```
delete clip.onEnterFrame;
```

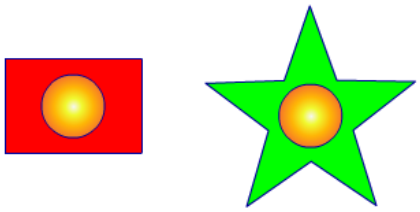
В случае часто происходящих событий (`onEnterFrame`, `onMouseMove`) это позволит освободить оперативную память и несколько уменьшить нагрузку на процессор.

Обработчики, созданные в сценарии клипа, удалить нельзя.

ЗАДАНИЕ №3

Создайте 2 символа:

- прямоугольник, в центре которого символ-круг (`circle`);
- звезда, в центре которого находится такой-же символ-круг.



При вхождении указателя мышки в область круга символ, содержащий этот круг поворачивается на 15 градусов.

Возможны 2 варианта эффективного написания кода:

1. На временной оси основного ролика создать именованную функцию поворота родительского объекта и присвоить ее обработчикам события RollOver обоих желтых кругов.

2. При создании символа-круга circle на его временной оси создать сценарий кадра

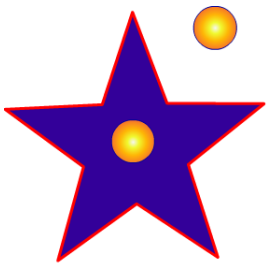
```
this.onRollOver=function(){
    this._parent._rotation+=15;
}
```

Выполнить оба варианта и добавить в рабочее поле еще один круг. Проанализировать поведение дополнительных кругов.

Задание №4

1. Создать новый fla файл. На его единственный кадр "перетащить " из библиотеки файла задания №3 символ circle. Протестировать файл.

2. Создать символ звездочку в центре которой экземпляр circle. Разместить в кадре экземпляр звездочки. Протестировать файл.



В чем удобство символов с кодом?

Контрольные вопросы к лабораторной работе №4

1. Создание методов-обработчиков событий.
2. Удаление методов-обработчиков событий.
3. Использование именованных функций для создания методов-обработчиков событий.
4. Использование указателя `this` в функциях и обработчиках событий.
5. Размещение кода на временной оси символа.
6. Объяснить код, использованный для выполнения задания.

Контрольные вопросы

1. Какие важные специфические характеристики присущи любому проекту?
2. Чем отличается проектное управление от традиционного менеджмента? Почему традиционный менеджмент можно назвать «рутинным управлением», а управление проектами — нет?
3. Перечислите принципы, какими необходимо руководствоваться при разработке методов и стандартов управления проектами.
4. Перечислите ключевые международные стандарты управления проектами. На решение каких задач направлено создание каждогостандарта?

Лабораторная работа №4.3 Язык управления сценариями Action Script. Библиотеки сценариев.

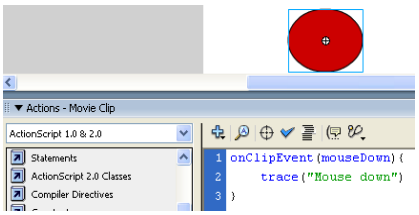
Цель работы: познакомиться с понятиями "Сервис" и "Сервис как класс" в проекте, рассмотреть их сходства и различия, закрепить практические навыки работы с системой Google analytics на примере реализации сервисов.

Общие сведения

ActionScript 2.0 поддерживает создание сценариев клипов и кнопок (стиль ActionScript 1.0).

Сценарии кнопок

Создание сценариев кнопок рассматривалось в предыдущей лабораторной работе.



При обращении к свойствам основного ролика (`_root`), в сценарии кнопки, расположенной непосредственно в `_root`, имя объекта можно опускать

```
on(press){
   _xscale+=10;    //эквивалентно   _root._xscale
}
```

Если же кнопка, для которой пишется сценарий, находится внутри другого объекта, то отсутствие имени объекта будет означать обращение к родительскому объекту.

```
on(press){
   _xscale+=10;    //эквивалентно   _parent._xscale
}
```

Указатель **`this`** является указателем на родительский объект.

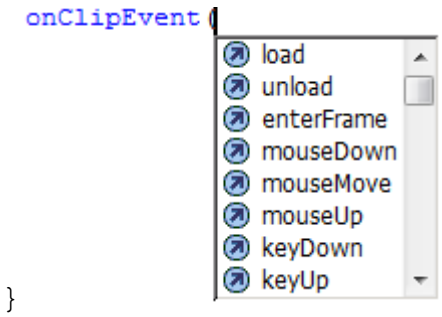
При ссылке на объекты можно опускать имя родительского объекта, общего для кнопки со сценарием и этого объекта. Например, объект `clip` и кнопка `b_btn` находятся внутри клипа `m_mc`. В обработчике кнопки `b_btn` к объекту clip можно обратиться `m_mc.clip` и просто `clip`.

Сценарии клипов

Код сценария клипа размещается в самом клипе и имеет вид

```
onClipEvent(событие){
    действия
}
```

Для открытия окна сценария клипа надо выделить клип и открыть окно Actions.



В его заголовке появится `Actions – Movie Clip`.

Кроме этого можно применять обработчики событий ***on***, которые применяются для кнопок.

```
on(событие){
```

```
        действия
    }
```

При обращении к свойствам фрагмента ролика, для которого пишется сценарий, имя объекта можно опускать.

```
onClipEvent (mouseDown) {
  _x+=10    //"хозяин" сценария перемещается вправо
}
```

Указатель `this` является указателем на фрагмента ролика, для которого пишется сценарий.

При ссылке на вложенные объекты надо опускать имя родительского объекта или использовать указатель `this`. Например, фрагмента ролика clip_mc находится внутри фрагмента ролика m_mc.

В сценарии фрагмента ролика `m_mc` к объекту `clip_mc` можно обратиться `this.clip_mc`, `_root.m_mc.clip_mc` или просто `clip`. При обращении `m_mc.clip_mc` на самом деле будет подразумеваться объект `m_mc.m_mc.clip_mc`.

События мыши

Событие `press` наступает при нажатии кнопки мыши *над* объектом, а событие `mouseDown` — при нажатии кнопки мышки в любом месте.

Событие `release` наступает при отпускании кнопки мышки над объектом, `releaseOutside` — при отпускании кнопки мышки вне объекта, а `mouseUp` — при отпускании кнопки мышки в любом месте. Событие `rollOver` наступает при вхождении указателя мышки в область объекта, `rollOut` — когда указатель мышки выходит за пределы объекта.

Событие `rollOver` наступает при вхождении указателя мышки в область объекта, `rollOut` — когда указатель мышки выходит за пределы объекта.

Если нажать кнопку мышки в области объекта и вывести указатель мышки за пределы объекта, то наступает событие `dragOut`. Если после этого ввести указатель в область объекта (не отпуская кнопку мышки), то для объекта наступит событие `dragOver`.

Свойства объектов MovieClip

Свойства **`_x`** и **`_y`** определяют координаты объекта в системе координат родительского клипа.

Ось **`X`** клипа направлена вправо, а ось **`Y`** — вниз. Первоначально направление осей вложенного и родительского клипов совпадают. При повороте клипа поворачиваются оси его локальной системы координат

относительно осей координатной системы родительского клипа. Угол поворота определяется свойством клипа **_rotation**.

Размеры символов определяются значениями свойств **_height** (высота) и **_width** (ширина) в пикселах. Следующий код сделает размеры символа **m_mc** 100x200 пикселов

```
m_mc._ height=100;
m_mc._ width =200;
```

Размеры символов определяются в системе координат самого символа.

Изменение размера можно задавать и в процентах к первоначальному значению. Чтобы увеличить высоту на 20% и ширину на 10%, надо выполнить следующий код

```
m_mc._ xscale=110;
m_mc._ yscale+=20;
```

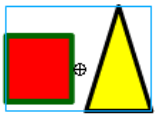
Контрольные вопросы

Что такое web-сервис?

1. Как интерпретируется ссылка this в сценарии кнопки?
2. Как интерпретируется ссылка this в сценарии символа MovieClip?
3. Как интерпретируется ссылка this в сценарии кадра?
4. Если перед именем свойства не указано имя объекта, то к какому объекту будет относится это свойство в сценарии символа MovieClip?
5. Если перед именем свойства не указано имя объекта, то к какому объекту будет относится это свойство в сценарии кнопки?
6. Если перед именем свойства не указано имя объекта, то к какому объекту будет относится это свойство в сценарии кадра?
7. Когда для объекта наступает событие press и когда событие mouseDown?
8. Изменяются ли координаты родительского объекта при изменении координат дочернего объекта?
9. Изменяются ли координаты дочернего объекта при изменении родительского координат объекта?
10. Изменяется ли свойство _rotation родительского объекта при изменении его у дочернего объекта?
11. Изменяется ли свойство _rotation дочернего объекта при изменении его у родительского объекта?
12. Изменяются ли размеры родительского объекта при изменении размера дочернего объекта?
13. Изменяются ли размеры дочернего объекта при изменении размера родительского объекта?
14. Вопросы по коду задания.

Список индивидуальных заданий

1. Создайте два символа типа MovieClip в виде квадрата с стороной 50 пикселов и треугольника с нижним основанием 50 и высотой 75. Точка регистрации обоих символов в центре.



2. Затем создайте новый символ, содержащий экземпляр квадрата с именем rec_mc и экземпляр треугольника с именем tri_mc. Координата _y внутренних объектов rec_mc и tri_mc должна равняться нулю. Координата _x у квадрата равна -30, а у треугольника +30.

Создайте экземпляр нового символа с координатами (300;200) и назовите outer_mc.

3. Создайте кнопку, при нажатии которой выполняются следующие действия:

- вывод координаты клипа outer_mc и вложенного в него rec_mc;
- перемещение outer_mc на 50 пикселов по горизонтали и по вертикали;
- вывод координаты клипов outer_mc и rec_mc после перемещения;

```
    // вывод координат родительского outer_mc и
    //дочернего rec_mc клипов
trace("Начальные координаты");
trace("_root.outer_mc._x=" + _root.outer_mc._x +
      "  _root.outer_mc._y=" + _root.outer_mc._y);
trace("_root.outer_mc.rec_mc._x="+_root.outer_mc.rec_m
c._x +  "  _root.outer_mc.rec_mc._y=" +
_root.outer_mc.rec_mc._y);
        // перемещение клипа outer_mc
_root.outer_mc._x+=50;
_root.outer_mc._y+=50;
       //вывод координат клипов после перемещения
trace("Координаты после перемещения outer_mc");
trace("_root.outer_mc._x=" + _root.outer_mc._x +
      "  _root.outer_mc._y=" + _root.outer_mc._y);
trace("outer_mc.rec_mc._x=" + outer_mc.rec_mc._x +
      "  _root.outer_mc.rec_mc._y=" +
_root.outer_mc.rec_mc._y);
```

4. При тестировании флеша, в окне Output появится:

```
Начальные координаты
```

```
_root.outer_mc._x=300      _root.outer_mc._y=200
_root.outer_mc.rec_mc._x=-30  _root.outer_mc.rec_mc._y=0
Координаты после перемещения outer_mc
_root.outer_mc._x=350      _root.outer_mc._y=250
_root.outer_mc.rec_mc._x=-30  _root.outer_mc.rec_mc._y=0
```

Координаты внешнего объекта `outer_mc` задаются относительно основного клипа `_root`, поэтому при перемещении `outer_mc` его свойства `_x` и `_y` изменились. Расположение внутренних объектов `outer_mc` относительно его точки регистрации при перемещении всего `outer_mc` не изменилось, следовательно локальные координаты внутреннего клипа `rec_mc` не должны измениться.

К заданию №1 добавьте кнопку, нажатие которой вызывает перемещение внутреннего объекта `rec_mc` по горизонтали вправо на `50` пикселов и вывод координаты.

```
_root.outer_mc.rec_mc._x+=50;
trace("Коорд. после перемещения внутрен. объекта rec_mc")
trace("_root.outer_mc._x=" + _root.outer_mc._x +
      "  _root.outer_mc._y=" + _root.outer_mc._y);
trace("_root.outer_mc.rec_mc._x="+_root.outer_mc.rec_mc._x
+"  _root.outer_mc.rec_mc._y=" +
_root.outer_mc.rec_mc._y);
```

5. После тестирования увидим в окне `Output` следующее:

```
Координаты после перемещения внутренего объекта rec_mc
_root.outer_mc._x=300  _root.outer_mc._y=200
_root.outer_mc.rec_mc._x=20  _root.outer_mc.rec_mc._y=0
```

Так как точка регистрации `outer_mc` не изменилась, координаты родительского объекта остались прежними. Смещение внутреннего объекта привело к изменению его локальных координат.

Посмотрим, что происходит с координатами при повороте (свойство `_rotation`).

Добавим кнопку, нажатие которой вызывает поворот внешнего объекта `outer_mc` на 45^{o} и вывод координат `outer_mc` и `rec_mc`.

```
_root.outer_mc._rotation+=45;// поворот по часовой
стрелке
trace("Координаты после поворота outer_mc");
trace("_root.outer_mc._x=" + _root.outer_mc._x +
      "  _root.outer_mc._y=" + _root.outer_mc._y);
trace("_root.outer_mc.rec_mc._x=" +
_root.outer_mc.rec_mc._x +
      "  _root.outer_mc.rec_mc._y=" +
_root.outer_mc.rec_mc._y);
```

В окне `Output` увидим, что не изменились координаты ни `outer_mc`, ни `rec_mc`.

```
Координаты после поворота outer_mc
_root.outer_mc._x=300  _root.outer_mc._y=200
_root.outer_mc.rec_mc._x=20  _root.outer_mc.rec_mc._y=0
```

Добавим кнопку для поворота квадрата rec_mc на 45^o . Вывести координаты `outer_mc` и `rec_mc` до и после поворота.

Координатами объекта являются координаты его точки регистрации. Поворот осуществляется вокруг точки регистрации, она же является и началом координат. Поэтому координаты объектов не изменяются при повороте.

Размеры клипа. Свойства _height, _width, _xscale, _yscale

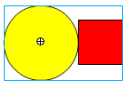
Высота клипа определяется его свойством `_height`, а ширина — `_width`. Значения этих свойств определяются в системе координат родительского клипа и зависят от масштаба ее осей.

Узнать или изменить масштаб координатных осей клипа можно с помощью свойств `_xscale` и `_yscale`. Измеряются значения этих свойств в процентах по отношению к масштабу осей недеформированного клипа.

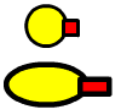
Ширина и высота клипа не могут принимать отрицательные значения.

Масштаб `_xscale` и `_yscale` может быть и отрицательным! При отрицательном масштабе происходит изменение направления соответствующих осей на противоположное.

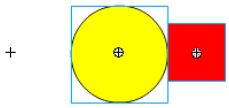
1. Создать символ `clip`, который содержит два внутренних символа: круг с именем circle и квадрат с именем square. Начало его системы координат расположить в центре круга.



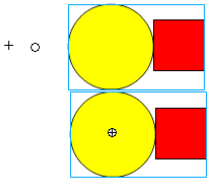
2. Создать кнопку, нажатие которой выводит в Output координаты и размеры родительского клипа clip и дочернего sguare. Затем в два раза растягиваем родительский клип по оси X и повторяется вывод.



3. Создать дубликат символа clip. Сместить круг и квадрат таким образом, чтобы координата x круга была равна 100.



Разместить экземпляры символов clip и его отредактированного дубликата друг над другом. Провести через центры кругов линию.



Создать кнопки, нажатие которых сжимает оба экземпляра по горизонтали в два раза, растягивает в два раза и восстанавливает первоначальные размеры.