ЯЗЫК ПРОГРАММИРУЕМОЙ ГРАФИКИ *ACTIONSCRIPT*

Понятие о сценарии ActionScript

- Набор инструкций (actions), которые осуществляют программное управление элементами фильма
- Обрабатываются интерпретатором ActionScript
- Могут быть непосредственно встроены в фильм, либо размещаться во внешнем текстовом файле с расширением *.as
- Могут быть рассредоточены по разным элементам структуры фильма

Запись сценариев ActionScript

- 3 объекта рабочей среды документа Flash могут содержать сценарий AS:

  * *ключевой кадр* основной монтажной линейки или монтажной линейки символа Movie Clip (кадр имеет метку «a») – *Frame Action*

  * экземпляр символа типа типа Button – *Button Action*

  * экземпляр символа типа Movie Clip – *Movie Clip Action*

- Важно убедиться, что сценарий размещен именно в том объекте, для которого предназначен:

  сценарий для кнопки в кадре вызовет ошибку при компиляции

Воспроизведение сценариев AS

- Сценарии выполняются только при наступлении 2 категорий событий:

  * *системные события System events* – генерируются автоматически при воспроизведении фильма

  * *пользовательские события User events* – генерируются пользователем (мышь, клавиатура)

- Сценарий кадра выполняется автоматически при достижении этого кадра при воспроизведении фильма

- Сценарий кнопки или клипа должен содержать явное указание на событие:

  используется инструкция – обработчик событий Event handler

Функции управления кадра

Остановка/запуск воспроизведения монтажной линейки:

- ***stop()***
- ***play()***

Перевод в указанный кадр и остановка/воспроизведение монтажной линейки:

- ***gotoAndStop([scene],frame)***
- ***gotoAndPlay([scene],frame)***
  функция ***gotoAndPlay(1)*** – в последнем кадре вызывает зацикливание

Безусловный переход:

- ***nextFrame(), prevFrame()***
- ***nextScene(), prevScene()***

Вывод в окно Output результата вычисления аргумента y:

- ***trace(y)***

Программирование кнопок

- Сценарий кнопки всегда размещается внутри обработчика события on():

  on(событие) { текст сценария; }

- Обработчик on() может одновременно перехватывать несколько различных событий:

  on(press, keyPress “a”) { текст сценария; }

- Сценарий кнопки может содержать несколько обработчиков событий

  on(press) { текст сценария1; }
  on(keyPress “a”) { текст сценария2; }

- Областью видимости обработчика событий on() является монтажная линейка, содержащая экземпляр кнопки со сценарием

События обработчика on()

Нажатие левой кнопки мыши:

- press - нажать
- release - отпустить
- rollOver – войти в область реагирования
- rollOut – выход из области реагирования
- dragOut - выход из области реагирования при нажатой кнопке мыши

Нажатие клавиш клавиатуры:

- keyPress “a”

Функциональные клавиши:

<Enter>, <Space>, <Right>, <Left>, <BackSpace>

События и обработчики событий клипов

- Обработчики событий клипов:
  on() – совпадает с обработчиком событий кнопок
  onClipEvent() - обрабатывает только одно событие

- События клипов:
  enterFrame – генерируется с частотой воспроизведения монтажной линейки
  mouseMove – генерируется при перемещении курсора с частотой, доступной процессору
  mouseDown – нажатие ЛКМ
  mouseUp – отпускание ЛКМ
  keyDown – нажатие клавиши
  keyUp – отпускание клавиши
  load – появление клипа в фильме
  data – загрузка в фильм внешних данных

Навигация внутри фильма

- Для управления воспроизведением монтажной линейки на ней необходимо разместить 2 слоя:

  * *первый* – для размещения анимации

  * *второй* – для размещения панели соответствующих кнопок с обработчиком

    on(press) { сценарий }

  и сценарием с соответствующей функцией:

    play(); - проиграть
    stop(); - остановить
    nextFrame(); - вперед
    prevFrame(); - назад
    gotoAndStop(1); - в начало

- Панель с кнопками должны быть видна на протяжении всего фильма

Управление окном фильма

- Для управления используется глобальная функция:
  FSCommand(«команда», «параметры»)
- Команды:
  fullscreen
  allowscale
  trapallkeys
  showmenu
  quit
  exec

Встроенные свойства клипов и кнопок

- Объекту нужно присвоить имя в Properties*: Instance Name*
- Встроенные свойства:

  _x – координаты объекта
  _y –
  _width – ширина
  _height – высота
  _xScale – масштаб сжатия в процентах
  _yScale –
  _rotation – угол поворота в градусах
  _alpha – прозрачность в процентах
  _visible – видимость
  _xmouse – текущие координаты курсора мыши
  _ymouse –

Адресация клипов и кнопок

- Если объект на монтажной линейке, то ссылка на него осуществляется по имени
- Если есть вложенные объекты, то нужно указывать путь

Адресация объектов:

абсолютная – путь указывается от основной монтажной линейки: _root

относительная – путь указывается относительно сценария, содержащего обращение: вниз – точка, вверх – свойство _parent

Основные конструкции ActionScript

- Переменные:
  - уровня монтажной линейки x=5;
  - локальные var x:Number=5;
  - глобальные _global x=5;
- УсловныйУсловный Условный оператор
  if(условие){утверждения 1;} else {утверждения 2;}
- Циклы
  while(условие){ утверждения; }
  for(нач.; усл.; прир.){ утверждения; }
- Функции
  function имя(арг1,арг2,…){ утверждения; }
- Методы

Назначение методов

- Используются для централизации программного кода в одном месте
- Вызывается объектом – *клипом MovieClip, кнопкой Button или объектами из других классов.*
- Вызов метода осуществляется только при наступлении соответствующего события
- Действия метода определяются функцией без имени, которая присваивается методу как его значение:

объект.событие=function(){ утверждения; }

Обработчики событий клипов и кнопок, используемых в качестве методов

onPress
onRelease
onRollOver
onRollOut
onEnterFrame
onMouseMove
onMouseDown
onMouseUp
onLoad
…

Использование класса Mouse

- Предоставляет возможность управлять стандартным курсором мыши
- Методы класса Mouse:

  hide()

  show()

  onMouseUp

  onMouseDown

  onMouseWheel

Управление звуком

- Объект Sound

  s1=new Sound();
- Методы:

  setVolume() – устанавливает громкость (0-100)

  setPan() – устанавливает громкость левой и правой колонок (-100 до 100)

  getVolume() – возвращает громкость

  getPan() – возвращает панорамность
- Свойства:

  _duration – продолжительность в милисекундах

Программное создание экземпляров клипов и кнопок

- Дублирование Дублирование имеющихся
  dublicateMovieClip(объект,дубликат,уровень)
  или
  объект.dublicateMovieClip(дубликат,уровень[,образ])
- Удаление (только для программно созданных экземпляров)
  removeMovieClip(объект)
  или
  объект.removeMovieClip()
- Размещение из библиотеки символов (*+Linkage*)
  объект. attachMovie(символ,имя_экз.,уровень[,образ])
- Создание пустого клипа
  объект.createEmptyMovieClip(имя,уровень)

Создание динамических масок

- Динамическая маска создается методом *setMask():*

- Процедура создания динамической маски:

  - создать на сцене клип *маска*
  - создать на сцене клип *маскируемый*
  - в первый кадр основной монтажной линейки поместить сценарий:

  маскируемый.setMask(маска)

  - добавить в сценарий свойство перетаскивания маски методом *startDrag():*

  маска.startDrag(true);