Introduction to Flash II

Nir Peer
CMSC434 TA
Fall 2008

ACTIONSCRIPT BASICS
Variables

```javascript
var i;       // declares an untyped variable
var i : *;   // declares an untyped variable
var i : int, j : int;
var i : int = 5, j : int = 10;
var pi : Number = 3.14;
var str : String = "Hello";
var u : uint = uint.MAX_VALUE;
var b : Boolean = true;
```

Arrays

```javascript
var ar : Array = new Array(1, 2, "c");
ar[0] = "A";     // -> ["A", 2, "c"]
ar.push("d");  // -> ["A", 2, "c", "d"]
var l : String;
l = ar.pop();   // -> ["A", 2, "c"]
delete ar[1];   // -> ["A", undefined, "c"]

var ar : Array = [1, 2, "c"];          // Array literal
```
Conditionals

```java
if (name1 == "jack" ||
    name1 == "john") {
    trace("Mr.");
} else if (name1 == "julie") {
    trace("Ms.");
} else {
    trace("M_.");
}

switch (name1) {
    case "jack":
    case "john":
        trace("Mr.");
        break;
    case "julie":
        trace("Ms.");
        break;
    default:
        trace("M_.");
}
```

Looping

```java
var i : int = 5;
while (i++ < 10) {
    trace(i);
}
```

```java
var i : int = 5;
do {
    trace(i);
} while (i++ < 10);
```

```java
for (var i : int = 5; i < 10; i++) {
    trace(i);
}
```
Functions

```javascript
function mult2(num : int) : int {
  return 2 * num;
}
```

```javascript
var mult2 = function(num : int) : int {
  return 2 * num;
}
```

```
trace(mult2(5));
```

EDITING ACTIONSCRIPT
Adding ActionScript

- ActionScript can be added to any Keyframe
- It is recommended to create a separate layer for scripts

The Actions window

- Actions Toolbox
- Script Navigator
- Actions Panel Context Menu
- Script Pane
Converting shapes to symbols

- After a shape is drawn on the stage, we can convert it to a symbol using the context menu
  - We can then assign a name
  - Set the symbol’s Registration (alignment anchor)
  - Choose its type
    - Movie clips have their own separate timeline

Referring to symbols from ActionScript

- Set the instance’s name

- Use this name in your ActionScript code
Event handling

```actionscript
function clickHandler (e : MouseEvent) : void {
  trace("Button was clicked!");
}

myButton.addEventListener(MouseEvent.CLICK, clickHandler);

myButton.removeEventListener(MouseEvent.CLICK, clickHandler);
```
Controlling playback

```javascript
this.play();
this.stop();
this.gotoAndPlay(15); // Jump to frame #15 and play
this.gotoAndStop(30); // Jump to frame #30 and stop
```

These functions can be invoked on any MovieClip object.

Scenes 1

- A Flash movie can be comprised of multiple scenes
- Each scene has its own timeline
- By default, Flash plays the scenes consecutively
  - This can be controlled using ActionScript
- To add a new scene, go to: Insert ⇒ Scene

- To switch between scenes, use the Edit Scene dropdown
Scenes 2

- You can rename, delete, and reorder scenes using the Scene Panel
  - Rename by double-clicking on a name
  - Reorder by dragging

Scenes 3

- Using ActionScript, you can play a different scene
- For example:
  `this.gotoAndPlay(1, "Scene 2");`
- In the above example, the jump is made to frame 1 of scene 2.
Hiding components

```javascript
// Hide components (events disabled)
cbl.visible = false; // see UIComponent

// Fade components by changing their transparency

cbl.alpha = 20; // see DisplayObject
```

Debugging

- Show the values of expressions using `trace(exp)`
  - Value is printed in the Output window

- Start the debugger from: Debug ➤ Debug Movie
  - Breakpoints are set in the debugger
Exercise 1

- Create a new Flash document
- Rename Layer 1 to content
- Add a new layer named scripts
- Draw a 75x75 pixels square in the content layer, at coordinates (0, 0)
  - Convert it to a MovieClip symbol named square
  - Set its instance name to box
- Right click on frame 1 of scripts and choose Actions
- Type the script:
  ```
  trace(box.x, box.y);
  box.y = 150;
  box.width = 400;
  ```
- Test your movie

Exercise 2

- Clear the ActionScript code from Exercise 1
- Insert a keyframe at frame 60 of content
  - In that frame, reposition the box on the right side of the stage
- Create a motion tween
- Increase the size of the box on frame 60
- Insert a keyframe at frame 60 of scripts
- Type the script:
  ```
  this.gotoAndPlay(30);
  ```
- Test your movie
Exercise 3

- Double click the box
- Insert a blank keyframe at frame 2
- Draw an 75x75 oval at coordinates (0, 0)
- Test your movie

- Edit frame 1 of the scripts layer, and add the code:
  ```javascript
  box.stop();
  ```
- Test your movie, and then add this code:
  ```javascript
  box.addEventListener(MouseEvent.CLICK, clickHandler);
  box.addEventListener(MouseEvent.MOUSE_OVER, overHandler);
  box.addEventListener(MouseEvent.MOUSE_OUT, outHandler);

  box.buttonMode = true;
  ```

Exercise 3 (cont'd)

- Add the code:
  ```javascript
  function clickHandler(evt : Object) : void {
      trace("Click!")
  }

  function overHandler(evt : Object) : void {
      box.gotoAndStop(2);
  }

  function outHandler(evt : Object) : void {
      box.gotoAndStop(1);
  }
  ```
Resources

- Adobe Flash CS3 Professional Help
  - Programming ActionScript 3.0
  - ActionScript 3.0 Language and Components Reference
- http://www.senocular.com/flash/tutorials/as3withflashcs3