Development of Games

Lecture 13

Introduction to Flash

Main concepts

- Flash technology is oriented on programming in WEB
- Flash player
- A Flash method is JavaScript function that is specific to Flash movies
- Flash movies are called by it in scripting environment

Flash

- Flash is an authoring tool that designers and developers use to create presentations, applications, and other content that enables user interaction
- Flash projects can include simple animations, video content, complex presentations, applications, and everything in between
- Flash is extremely well suited to creating content for delivery over the Internet because its files are very small

Flash (2)

- To build an application in Flash, you create graphics with the Flash drawing tools and import additional media elements into your Flash document.
- Next, you define how and when you want to use each of those elements to create the application you have in mind
- Flash file can be started and controlled from JavaScript by Flash methods

Flash (3)

- **The Stage** is where your graphics, videos, buttons, and so on appear during playback.
- **The Timeline** is where you tell Flash when you want the graphics and other elements of your project to appear. You also use the Timeline to specify the layering order of graphics on the Stage. Graphics in higher layers appear on top of graphics in lower layers.
- **The Library** panel is where Flash displays a list of the media elements in your Flash document.
- **ActionScript** code allows you to add interactivity to the media elements in your document. For example, you can add code that causes a button to display a new image when the user clicks it. You can also use ActionScript to add logic to your applications. Logic enables your application to behave in different ways depending on the user's actions or other conditions. Flash includes two versions of ActionScript, each suited to an author's specific needs

Standard Flash methods

GetVariable

Syntax GetVariable(varName)

Description Returns the value of the Flash variable specified by varName. Returns null if the variable does not exist. The argument type is string.

Example var firstName = movie.GetVariable("FirstName");

var radioButtonValue = movie.GetVariable("/Form/RadioButton:Value");

GotoFrame

Syntax GotoFrame(frameNumber)

Description Activates the frame number specified by frameNumber in the current movie.

If the data for a requested frame is not yet available, the player goes to the last frame available and stops, causing unexpected results during playback. Use the PercentLoaded() method to determine if enough of the movie is available to execute the GotoFrame() method. The argument frameNumber is zero-based; that is, frameNumber is 0 in the first frame of the movie, 1 for the second frame, and so on. This differs from the Goto action within Flash, which begins at 1. The argument type is integer.

- **Example** movie.GotoFrame(24);

IsPlaying

Syntax IsPlaying()

Description Returns true if the movie is currently playing.

Example if (movie.IsPlaying()) { alert("movie is playing"); }

LoadMovie

Syntax LoadMovie(layerNumber, url)

Description Loads the movie identified by url to the layer specified by layerNumber. The argument type is integer for layerNumber and string for url.

Example movie.LoadMovie(0, "mymovie.swf");

Pan

Syntax Pan (x, y, mode)

Description Pans a zoomed-in movie to the coordinates specified by x and y. Use mode to specify whether the values for x and y are pixels or a percent of the window. When mode is 0, the coordinates are pixels; when mode is 1, the coordinates are percent of the window. Pan does not pan beyond the boundaries of the zoomed-in movie. The argument type for all arguments is integer.

Example This example pans 50% right and 50% down:

movie.Pan(50, 50, 1)

This example pans -25 pixels left and -25 pixels up: movie.Pan(-25, -25, 0)

PercentLoaded

Syntax PercentLoaded()

Description Returns the percent of the Flash Player movie that has streamed into the browser so far; possible values are from 0 to 100.

Example if (movie.PercentLoaded() == 100) { loaded = true; }

Play

Syntax Play()

Description Starts playing the movie.

Example movie.Play();

Rewind

Syntax Rewind()

Description Goes to the first frame.

Example movie.Rewind();

SetVariable

Syntax SetVariable(variableName, value)

Description Sets the value of the Flash variable specified by variableName to the value specified by value. The argument type for both arguments is string.

Example movie.SetVariable("/Form:UserName", "John Smith");

SetZoomRect

Syntax SetZoomRect (left, top, right, bottom)

Description Zooms in on a rectangular area of the movie. The units of the coordinates are in twips (1440 units per inch). To calculate a rectangle in Flash, set the ruler units to Points and multiply the coordinates by 20 to get twips. (There are 72 points per inch.) The argument type for all arguments is integer.

Example This example zooms in on a 200 x 200 pixel rectangle in the upper left corner of the movie:

```
var pointsToTwips = 20;
movie.SetZoomRect(0, 0, 200 * pointsToTwips, 200 *
    pointsToTwips);
```

StopPlay

Syntax StopPlay()

Description Stops playing the movie.

Example

movie.StopPlay()

TotalFrames

Syntax TotalFrames()

Description Returns the total number of frames in the movie.

Example var totalFrames = movie.TotalFrames();

Zoom

Syntax Zoom(percent)

Description Zooms the view by a relative scale factor specified by percent. Zoom(50) doubles the size of the objects in the view. Zoom(200) reduces the size of objects in the view by one half. Zoom(0) resets the view to 100%.

You cannot specify a reduction in the size of objects in the view when the current view is already 100%. The argument type is integer.

Example movie.Zoom(50);

Tell target Flash methods

TCallFrame

Syntax TCallFrame(target, frameNumber)

Description In the timeline specified by target, executes the action in the frame specified by frameNumber.

Example This example runs the actions in the fifth frame of the main timeline:

```
movie.TCallFrame("/", 4);
```

TCallLabel

Syntax TCallLabel(target, label)

Description In the Timeline indicated by target, executes the action in the frame specified by the label frame label. The argument type for both arguments is string.

Example This example runs the actions in the frame labeled HandleScriptNotify in the main timeline:

```
movie.TCallLabel("/", "HandleScriptNotify");
```

TCurrentFrame

Syntax TCurrentFrame(target)

Description Returns the number of the current frame for the timeline specified by target. The frame number returned is zero-based, meaning frame 1 of the Flash movie would be 0, frame 2 would be 1, and so on. The argument type is string.

Example var currentFrame = movie.TCurrentFrame("/MovieClip");

TCurrentLabel

Syntax TCurrentLabel(target)]

Description Returns the label of the current frame of the timeline specified by target. If there is no current frame label, an empty string is returned. The argument type is string.

Example var currentLabel = movie.TCurrentLabel("/MovieClip");

TGetProperty

Syntax TGetProperty(target, property)

Description For the timeline indicated by target, returns a string indicating the value of the property specified by property. For property, enter the integer corresponding to the desired property.

Example var nameIndex = 13;

var name = movie.TGetProperty("/", nameIndex);

TGetPropertyAsNumber

Syntax TGetPropertyAsNumber (target, property)

Description For the timeline indicated by target, returns a number indicating the value of the property specified by property. For property, enter the integer corresponding to the desired property.

Example var framesLoadedIndex = 12;

var framesLoaded = movie.TGetProperty("/", framesLoadedIndex);

TGotoFrame

Syntax TGotoFrame(target, frameNumber)

Description For the timeline indicated by target, goes to the frame number specified by frameNumber. The argument type for target is string. The argument type for frameNumber is integer.

Example

movie.TGotoFrame("/MovieClip", 2);

TGotoLabel

Syntax TGotoLabel(target, label)

Description For the timeline indicated by target, goes to the frame label specified by label. The argument type for both arguments is string.

Example

```
movie.TGotoLabel("/MovieClip", "MyLabel");
```

TPlay

Syntax TPlay(target)

Description Plays the timeline specified by target.
The argument type is string.

Example

```
movie.TPlay("/MovieClip");
```

TSetProperty

Syntax TSetProperty(target, property, value)

Description For the timeline indicated by target, sets the value of the property specified by property to the value specified by value, which can be a string or a number. For property, enter the integer corresponding to the desired property.

Example

```
var visibilityIndex = 7;
var nameIndex = 13;
movie.TSetProperty("/MovieClip", visibilityIndex, 1);movie.TSetProperty("/MovieClip", nameIndex, "NewName");
```

TStopPlay

Syntax TStopPlay(target)

Description Stops the timeline specified by target. The argument type is string.

Example

```
movie.TStopPlay("/MovieClipToStop");
```

Standard events

OnProgress

Syntax OnProgress(percent)

Description

Generated as the Flash movie is downloading. The argument type is integer.

OnReadyStateChange

Syntax OnReadyStateChange(state)

Description

Generated when the ready state of the control changes.

The possible states are:

0=Loading, 1=Uninitialized, 2=Loaded, 3=Interactive, 4=Complete.

The argument type is integer.

FSCommand

Syntax FSCommand(command, args)

Description

Generated when an FSCommand action is performed in the movie with a URL and the URL starts with FSCommand :.

Use this to create a response to a frame or button action in the Flash movie. The argument type is string

Available properties

Property	Property number	Constant	Get	Set
X POSITION (_x)	0	X_POS	÷	÷
Y POSITION (_y)	1	Y_POS	÷	÷
X SCALE	2	X_SCALE	÷	÷
Y SCALE	3	Y_SCALE	÷	÷
CURRENTFRAME	4	CURRENT_FRAME	÷	
TOTALFRAMES	5	TOTAL_FRAMES	÷	
ALPHA	6	ALPHA	÷	÷
VISIBILITY	7	VISIBLE	÷	÷
WIDTH	8	WIDTH	÷	
HEIGHT	9	HEIGHT	÷	
ROTATION	10	ROTATE	÷	÷
TARGET	11	TARGET	÷	
FRAMESLOADED	12	FRAMES_LOADED	÷	
NAME	13	NAME	÷	÷
DROPTARGET	14	DROP_TARGET	÷	
URL(_url)	15	URL	÷	

Global properties

Global Property	Property number	Constant	Get	Set
HIGHQUALITY	16	HIGH_QUALITY	÷	÷
FOCUSRECT	17	FOCUS_RECT	÷	÷
SOUNDBUFTIME	18	SOUND_BUF_TIME	÷	÷