

Using Event Handlers (textbook page 32- 33) – you will need the files from Lesson 2 on the CD

For every event there is an action - Mouse Events, Frame Events, Movie Clip Events, Listeners

Objectives:

- 1) Learn how event handlers are used in scripts
- 2) Determine the best event handler for the job
- 3) Use mouse/button events to control interactivity
- 4) Add keyboard control to a project
- 5) Create a self running presentation
- 6) Use clip events
- 7) Orchestrate multiple events to accomplish a task
- 8) Define event handler methods
- 9) Learn about listeners and how to use them

Event Handlers

When this happens (eventHandler)

```
{  
do this;  
}
```

Mouse Events – in Flash 5 you could only attach them to buttons, in Flash MX you can also add mouse events to **buttons and movie clips. You will add mouse events to movie clips in this tutorial.**

on (press) {} - making contact, when you touch something

on (release) {} - letting go, when you let go

on (releaseOutside) {} - pulling snapping, push down and drag mouse then release

on (keyPress) {} - press on specific keyboard character

on (rollOver) {} - over but not touching or pressing down on to it

on (rollOut) {} - no longer on top of

on (dragOver) {} – bumping, press mouse over object and move back and forth

on (dragOut) {} – touch something then pull away “Oops”

.
.

MOUSE EVENTS

- 1) To cause a movie clip to be treated as a button attach a mouse event to it.
- 2) You can not attach both mouse events and clip events to a single instance (but you can use event handler methods to set a movie clip to react to both)
- 3) When you attach a mouse event to a movie clip it retains all of its movie clip properties
- 4) You can not attach clip events to button instances
- 5) Hit area of a movie clip will be the shape of the area of any graphical content it contains (however hitarea it can be changed see page 46)

There are 5 Flash exercises - one example of Listener

1) MouseEvents1.flas 2) ClipEvents1.flas 3) FrameEvents.flas 4) CarParts1.flas 5) OrchestratingEvents1.flas

Exercise #1 MOUSE EVENTS Time to complete approx 45 min to 1 hr.

Follow the instructions on starting on Page 37 (Playroom Scanner and cards)

Open Mouseevents1.flas - the movie has two scenes – Scan and Playroom (save as Mouseevents2.flas)

Basically you will add a variety of mouse events to various movie clips – hand, scanner.

Make a draggable “hand”

mouse.hide() – this hides the mouse when you drag the hand

startDrag(“hand”, true) - starts drag of the hand

message.text – outputs text messages into the dynamic text box called “message”

Note hand is a movie clip named – **hand**

Dynamic textfield below called **message**

Black box in the middle is a button named – **scanner** (not in my movie so give it the instance name scanner)

this keyword – refers to the current object, movie clip or text box

e.g. on(releaseOutside)

```
{
this._x = _root._xmouse
this._y= _root._ymouse
}
```

this = the movieclip instance name you moving e.g. “hand” and it can be replaced with the instance name it is simply an abbreviation for the object

You will add various mouse events - note you can add more than one event on (press), on(release), on (rollOver) to the same button or movie clip.

Events

startDrag ("hand", true)
on (rollOver)
on (rollout)
on (press)
on (dragOut)
on (release)
on (releaseOutside)

Exercise #2 FRAME ACTIONS - time to complete approx 45 min to 1 hr.

Follow the instructions starting on Page 47 (My European Vacation – Slide Show + related text)

Open FrameEvents.fla - save finished product as FrameEvents2.fla

You will make a slide show – movie within a movie and control the movie and text that appears in dynamic text fields. You will then add play, stop and rewind actions to buttons.

The way this slide show works is you have your main movie 10 seconds long (200 frames), it loops and advances your picture movie clip one frame which is made up of 5 images. You could easily add more images. Also the movie displays text when a specific frame is displayed. You then add buttons to rewind, stop and play the main movie timeline.

This exercise will demonstrate how to use frame events to control movie clips.

- 1) on (release) { stop}
- 2) on(rollover) {balloon.text = "stop"}
- 3) on(rollout) {balloon.text = " ";}

Exercise #3 CLIP EVENTS - Burglar movie clip appearance or disappearance triggers changes in the main movie. Clip Events are associated with movie clips and can be used to trigger changes when they appear (load) or disappear (unload).

Although you can add mouse events to movie clips, you can not add clip events to buttons.

Follow instructions starting on page 57

Refers to movie clips entering a scene with attached scripts

- 1) **onClipEvent(load)** - triggers a script when a movie clip enters a scene
- 2) **onClipEvent(unLoad)** – triggers a script when a movie clip exits a scene
- 3) **onClipEvent(mouseMove)** – motion detector within your movie detects mouse movement
- 4) **onClipEvent(mouseDown), onClipEvent(mouseUp), onClipEvent(keyDown), onClipEvent(keyUp)**

*execute scripts when mouse buttons or keys are pressed or released – these events can be triggered anywhere on the stage, unlike buttons where you must click on the button.

5) **OnClipEvent (data)** – flash lets you load various types of data (variables, external swfs, jpgs) from external sources, and once the data is loaded it triggers a script.

6) **onclipEvent(enterFrame)** – triggers a script continually at the same rate as your movie. If your movie plays at 20 frames per second the event handler triggers the script at 20 frames per second.

OnClipEvent(enterFrame)

```
{
_root.time++;
_root.timeAmount.text = _root.time;
root.timer._rotation = root.timer._rotation + 1;
}
```

This makes a movie clip called timer rotate at 24 fps $360/24 = 15$ seconds to completely rotate

EVENT HANDLER METHODS (CarParts fla - changing text field triggers movieclips)

So far you have seen how mouse interaction on (press), timeline events (gotoAndPlay()) and onClipEvents(load) can trigger scripts in your movie. However you can also trigger scripts when

- a) sounds finnish playing
- b) the stage is resized
- c) text is entered in a textfield

```

OnClipEvent(mouseDown)
{
    _rotation= 45;
}

```

this script when attached to a movie clip instance causes the movie clip to rotate when the mouse is pressed down and the script is attached to the movie clip instance

Using an Event Handler Method you place the script in a frame (root of the movie)

```

MyMovieclip.onMouseDown = function ()
{
    myMovieClip._rotation = 45; // path is required to identify movieClip
}

```

Defining an event handler method is the only way to execute a script when an event occurs in relation to a Sound, Loading a Variable, or modifying a text field

Another advantage of using Event Handlers is that events associated with single buttons can change in different frames.

| Mouse Event | Event Handler Method |
|--------------------|---|
| on (press) | buttonName.onPress or movieClipName.onPress |
| on (release) | buttonName.onRelease or movieClipName.onRelease |
| on (enterFrame) | button.onEnterFrame or movieClipName.onEnterFrame |

SOUND

SoundObject.onLoad
SoundObject.onSoundComplete

TEXT FIELDS

nameoftextfield.onChangeed (change the text in the text box)
nameoftextfield.onKillFocus (move away from the text box)
nameoftextfield.onScroller
nameoftextfield.onSetFocus (placed curson in the text box)

LOADING A VARIABLE

NameofLoadVarsObject.onLoad

Exercise #4 Orchestrating multiple events – you will apply on(press), on(dragOut) and on(releaseOutside) – in order to track the x coordinate of the mouse – which will determine how hard you hit a pool ball and hence how far it goes.

```
hitAmount = mouseEnd-mouseStart
```

```
TrackMouse = true; // turns on mouse tracking  
TrackMouse=false; // turns it off
```

Cool Script forms a loop that controls how far the ball moves

```
OnClipEvent(enterFrame)  
{  
  if (_root.hitAmount > 5)  
  {  
    this._x=this._x -10;  
    _root.hitAmount = _root.hitAmount - 2;  
  }  
}
```

LISTENERS (note typo error on page 82 remove quotes around Mouse.addListener(myTextField))

A Listener method enables an object (e.g. text box) to react to changes in the object it is listening to (e.g. Mouse). You can have multiple objects listening for particular events

The Mouse has several listener methods: onMouseUp, onMouseDown, onMouseMovie

Simple example – create a new movie, place a dynamic text box on the screen and call it myTextField. Select the first frame and add the following script

```
myTextField.onMouseMove = function()  
{  
  myTextField.text = x_mouse + “,” + _ymouse; // could use this in place of myTextField  
}  
Mouse.addListener(myTextField)
```

When you move the mouse it will output the x, y coordinates of the mouse into the text box called myTextField

You can add another text field box and then add a listener to to it and output a different message to it.

To remove a listener Mouse.removeListener(object)

ASSIGNMENTS for 5 marks 15 marks each

- 1) Create a movie clip (box or ball), make the movie clip draggable. The position of the movie clip should be output to 2 dynamic text fields – xposition, yposition. As the movie clip is dragged – the coordinates of the box should be displayed. Note that moving the object downward results in positive y values.

Answer

1. Make the background of the flash movie a different color, create two text fields – dynamic text boxes, name one xposition, the other yposition.
2. Create a new layer and draw a box or circle on the main movie stage– convert the box to a movie clip and give it an instance name (not necessary to name movie but its a good idea).
3. Select the movie clip and add the following script to the movie clip

```
onClipEvent(load)
```

```
{  
    startDrag(this, true); // this is current object, true attaches mouse to movie clip  
}
```

```
onClipEvent (mouseMove) // this outputs the x,y coordinates of movie clip to text boxes
```

```
{  
    _root.xposition.text = root._xmouse ;  
    _root.yposition.text = root._ymouse;  
}
```

Previously we had to make a movie clip, then convert it to a button and add the event handlers

```
on (press)
```

```
{  
    startDrag("movieclipname");  
}
```

```
on (release)
```

```
{  
    stopDrag(); // no quotes  
}
```

```
onClipEvent (mouseMove) // this outputs the x,y coordinates of movie clip to text boxes
```

```
{  
    _root.xposition.text = root._xmouse ;  
    _root.yposition.text = root._ymouse;  
}
```

this script follows the mouse and outputs its coordinates not the movie clip unless the movie clip is being dragged by the mouse

Assignment #2. 10 marks (Tip see page 152-153 functions)

- a) Make a circle or use the wheel movie clip from Chapter 2 in a new movie write a script so that when the movie starts the WheelClip rotates continuously.

Answer

- 1) create a circular object or use the tire movie clip name it WheelClip
- 2) create a new layer, call it actions and add the script below to the frame

```
WheelClip.onEnterFrame = function()
{
WheelClip._rotation += 30; // actual value not important must be += not =+ !!!!
}
```

Control Test movie – the object should rotate continuously

Another way you could do it – add the script below to the movie clip instance called timer

```
onClipEvent(enterFrame)
{
    _root.time++;
    _root.timer._rotation++;
    _root.timeAmount.text=_root.time; // just outputs time to text box
}
```

create the wheel movie clip and call it timer, object will rotate at the current frame rate

- b) Modify the script so that the wheelclip movie turns 30 degrees only when the mouse is pressed down

```
WheelClip.onMouseDown = function() // attach to frame not movie clip, .onpress OK
{
WheelClip._rotation +=30;
}
```

- c) Modify the movie clip so that the WheelClip movie onEnterFrame rotates and moves down and to the right

```
WheelClip.onEnterFrame = function()
{
WheelClip._rotation++; // ++ rotates clockwise, -- rotate counter clockwise
WheelClip._x++; // move to the right you can vary the amount WheelClip._x+=5 or more
WheelClip._y++; // move down
}
```

Here is another way to do it. Action is added to movie clip not the frame.

```
onClipEvent(enterFrame)
{
    _root.time++; // time is just a variable to record the time or frame number
    _root.timer._rotation++;
    _root.timer._x++; // moves object to the right
    _root.timer._y++; // moves object to left
    _root.timeAmount.text=_root.time; // outputs time – not required for animation
}
```

- d) To make the WheelClip movie move down a plane – you need to know the slope of the plane. Slope is determined by $y_2 - y_1 / x_2 - x_1$. You can calculate the slope of a line in Flash by creating two points on the line and reading the x,y coordinates in the property box. If the slope is $142/530 = 0.26$ you modify the script above as follows

```
WheelClip.onEnterFrame = function() // must be attached to frame not the movie clip
{
    WheelClip._rotation++; // ++ rotates clockwise, -- rotate counter clockwise
    WheelClip._x+=1.42
    WheelClip._y+=5.30
}
```

End Notes

```
onClipEvent(mouseDown)
{
    _root.WheelClip._rotation = 30;
}
```

using the script attached to the movie clip results in the movie clip only rotating once root is required. In this instance the main movie timeline rotates. This.WheelClip._rotation does the same thing.

Another way you could make a self rotating movie is to:

- 1) make a box, convert to symbol movie clip
- 2) edit the movie clip, right click>create 5 keyframes add an action to first 4 keyframes

```
on (press)
{
    this.nextFrame()
}
In the last frame add
```

```
GotoAndStop(1)
```

Assignment #3

Take the Slide Show on Page 48 – Change the background, add your own pictures and text