# THE FLASH CLASSROOM ACTIONSCRIPT REFERENCE GUIDE

Version 1 – Released October 2006 – Developed by Kristine Kopelke

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Make the Flash movie Full Screen when showed in projector format. | fscommand("fullscreen","true"); | Add this script to the first keyframe of your flash file. |
| Exit the current flash file. | on (release) {<br>fscommand("quit");<br>} | Add this script to an exit button you create. This button would normally be located in the top right hand corner of your flash file. |
| Stop the Flash movie at the current keyframe. | stop(); | Apply this script to the keyframe on the timeline where you'd like your movie to stop. |
| Go to another frame within the timeline in the current scene. | gotoAndStop(10);<br>// remember to edit the 10 with the frame number you want. | Apply this script to a keyframe when you want Flash to move to another frame when this keyframe is hit. |
| Go to another frame within the timeline in the current scene when the user presses the button. | on(press){<br>gotoAndStop(10);<br>}<br>// remember to edit the 10 with the frame number you want. | Apply this script to a button. |
| Go to another section of the Flash file when the user releases a button.<br><br>Note that the event release can be replaced by a range of other options e.g. press, release outside, rollover or even keyPress "<Left>" . In this final case, the term Left refers to the Left arrow key. You can replace this with any key on the keyboard e.g. "<Enter>". | on (release) {<br>gotoAndPlay("scenename", 1);<br>} | Apply this script to a button. |

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Launch a website or an external file when the user clicks this button.<br><br>Note that if it is a website, the url needs to begin with http:// . This lets Flash know it is a website. If it is an external file, ensure that the file is located within the same folder as the flash movie and just enter document name e.g. overview.pdf. | on (release) {<br>getURL("enterurlhere ", "_blank");<br>} | Apply this script to a button. |
| Stop all sounds currently playing. | stopAllSounds(); | Apply this script to a keyframe or add to a button after the standard on (press) or (release) command. |
| Makes an object into an object that the user can drag and drop. | on(press) {<br>startDrag(this);<br>}<br>on(release) {<br>stopDrag();<br>} | Apply this script to a movie clip symbol. |
| Make an object into a draggable object that the user must drag to a specific target.<br><br>In this case, the target will be a movie clip that you have given an instance name. In the script to the right, the movie clip that is the target has been given the name target1. The script works like this. The first lines enable the user to drag and drop the movieclip that the script is applied to, The next lines state that if the movieclip is dropped on the target, it will remain on the target. Finally, the last lines say that if it isn't dropped on the target movieclip, that the movieclip will return to the original coordinates. | on(press) {<br>startDrag(this);<br>}<br>on(release) {<br>stopDrag();<br><br>if (this._droptarget == "/target1") {<br>this._x = _root. target1._x;<br>this._y = _root.target1._y<br>}<br>else{<br>this._x = enterxcoordinate;<br>this._y = enterycoordinate;<br>}<br>} | Apply this script to a movie clip symbol.<br><br>Ensure that you have another movie clip symbol which has been given an instance name e.g. target1. This symbol which becomes the target should be located on a layer below the layer where the first movieclip appears. |

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Make an object into a draggable object that will generate a response if dropped in the correct location.<br><br>This requires that you have created two movie clip symbols in the library. The first symbol is the one that you want to be dragged. You can name this whatever you want. The second movieclip will be the one that provides the response. This movieclip should have two keyframes both with a stop(); action applied to it. The first keyframe could be empty or could contain the image that the response will go in e.g. an unchecked box. The second keyframe should contain the response you want. The script states in the if else statement that if the movieclip is dragged into the correct coordinates – play keyframe 2 of the movie, If not, remain on keyframe 1. | on (press) {<br>startDrag(this);<br>}<br>on (release) {<br>stopDrag();<br><br>if ((this._x>=84.5) & (this._x<=200.3) & (this._y>=122.9) & (this._y<=197.1)) {<br><br>_root.movieclipinstancename.gotoAndStop(2);<br>} else {<br>_root.movieclipinstancename.gotoAndStop(1);<br>}<br>} | Apply this script to a movieclip symbol.<br><br>Ensure that the movieclip symbol containing the response is given the instance name you use in your script. |

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Makes a movie clip visible if it is currently not. If it is visible, it makes it invisible. | on (release) {<br>if (_root. instancename._visible == 0) {<br>_root.instancename._visible = 1;<br>} else {<br>_root.instancename._visible = 0;<br>} | This script is added to a button that the user will press. When they press this button, the movieclip that has the instance name that is referred to in the script will become visible or invisible. This script can also attached to a button off the stage and release is replaced by the keypress option. |
| Changes the alpha of a movie clip to 50% if it is at 100%. If it is already at 50%, it converts it back to 100%. | on (release) {<br>if (_root. instancename._alpha == 100) {<br>_root.instancename._alpha = 50;<br>} else {<br>_root.instancename._alpha = 100;<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |
| Scales the movieclip using the x dimensions of the movie clip or returns them to original dimensions.<br><br>** Exchange xscale with yscale in this script to change the y dimensions. | on (release) {<br>if (_root. instancename._xscale == 100) {<br>_root.instancename._xscale = 50;<br>} else {<br>_root.instancename._xscale = 100;<br>}<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Changes the movie clips height or returns it back to the original height. | on (release) {<br>if (_root. instancename._height ==200) {<br>_root. instancename._height = 50;<br>} else {<br>_root. instancename._height = 200;<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |
| Changes the movie clips width or returns it back to the original height. | on (release) {<br>if (_root. instancename._width == 100) {<br>_root. instancename._width = 50;<br>} else {<br>_root. instancename._width = 100;<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |
| Moves the movie clip to a new x coordinate location.<br><br>** Replace x with y to change the y coordinate location. | on (release) {<br>if (_root. instancename._x == 200) {<br>_root.instancename._x = 100;<br>} else {<br>_root. instancename._x = 100;<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Rotates the object by 90 degrees or if already rotated returns it to 0 degrees. | on (release) {<br>if (_root. instancename._rotation<br>== 0) {<br><br>_root.instancename._rotation = 90;<br>} else {<br>_root.<br>instancename._rotation = 0;<br>}<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |
| Rotates the object by a number of degrees each time it is pressed (in this case the degrees is set at 10). This script does not return the object to 0 degrees as per example above. | on (release) {<br>(_root. instancename._rotation ==<br>0) {<br>_root. instancename._rotation<br>= 10;<br>} else {<br>_root. instancename _rotation<br>= _root. instancename._rotation<br>+10;<br>}<br>} | This script is added to a button that the user will press. It can also be put on an invisible button off the stage when release is replaced by the keypress option. |
| Create a Custom Mouse Cursor<br><br>To achieve this effect, there are two separate pieces of Actionscript you need to apply to a movieclip and the timeline. | onClipEvent (enterFrame){ ←<br>startDrag(this,true);<br>}<br><br>Mouse.hide(); ← | To change the cursor, draw the object you want the cursor to look like and convert it to a movieclip symbol. Attach this script to the movieclip symbol.<br><br>Add this script to the first keyframe in which you want the mouse cursor to be hidden. |

| Task to Perform | Actionscript Required | Location for Script |
|---|---|---|
| Inserts the current date into a dynamic text field. | thisDate = new Date();<br>allDays = new Array("Sunday", "Monday",<br>"Tuesday", "Wednesday", "Thursday",<br>"Friday", "Saturday");<br>allMonths = new Array("January", "Feburary",<br>"March", "April", "May", "June", "July",<br>"August", "September", "October",<br>"November", "December");<br>thisDay = thisDate.getDate();<br>if (thisDay.length()<=1) {<br>}<br>thisYear = thisDate.getFullYear();<br>thisDateString = " " add<br>allDays[thisDate.getDay()] add ", "<br>add<br>allMonths[thisDate.getMonth()] add " " add<br>thisDay add ", " add thisYear;<br>_root.DateTimeField = thisDateString; | This script is added to the keyframe on the timeline where the dynamic text box appears.<br><br>Ensure that you have given the dynamic text box the variable name **DateTimeField**. |

Note: Great care has been taken to ensure the code within this guide is correct. If you do find an error, please email it through to me at kristine@eq.edu.au.