# **Coding Buttons**

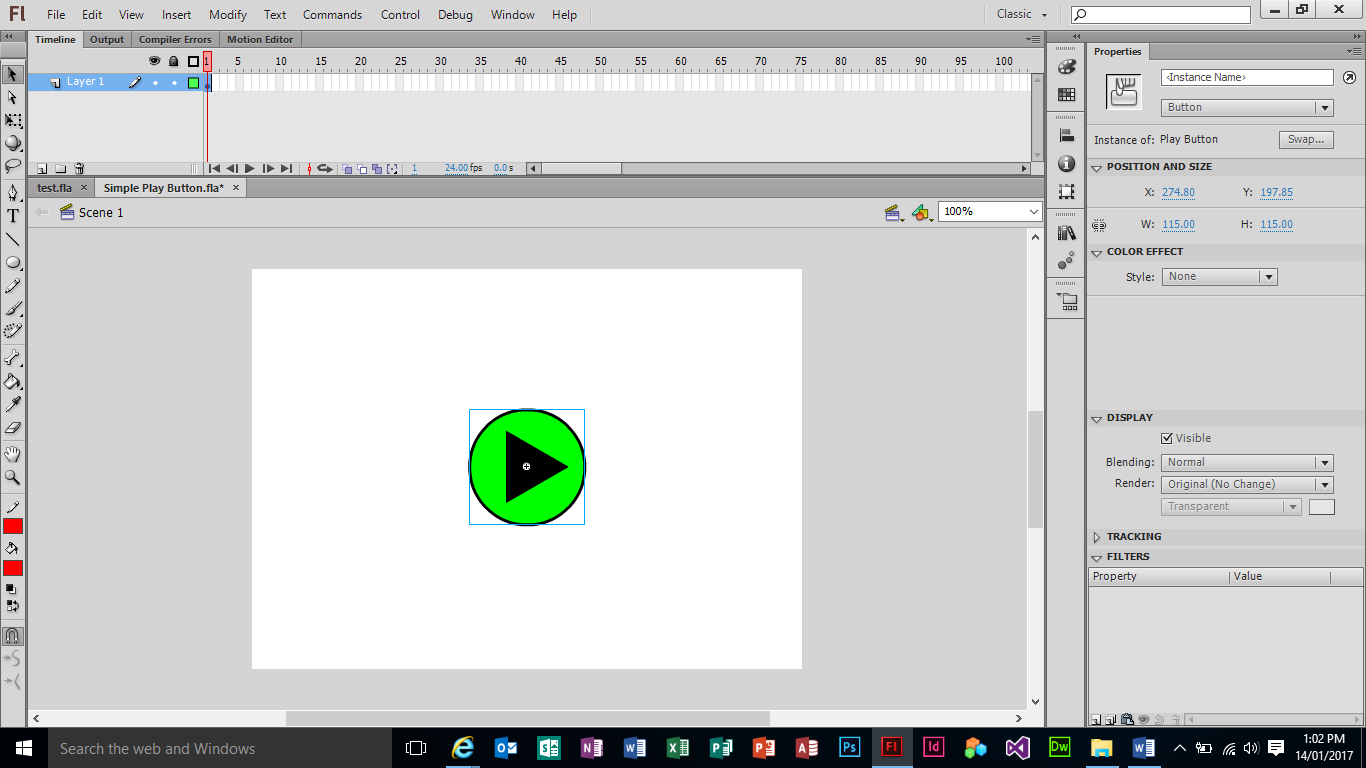
## **A Simple Play Button**

Most games will require a button to start the game by moving to a particular frame. The following instructions will show you how to create a play button, a new scene, and create the Action Script to make the button work.

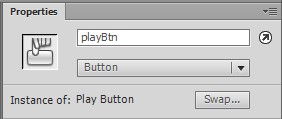
**Create a Play Button**

Open a new Flash ActionScript 3 file and save it as “Simple Play Button.fla”.

Create a new button symbol called “Play Button” and create a play button that looks something like the following. Drag a copy of the symbol onto Frame 1 as shown below.

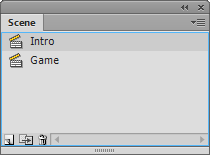


Click on the button symbol on the Stage and then in the Properties panel give the symbol and instance name of “playBtn”.



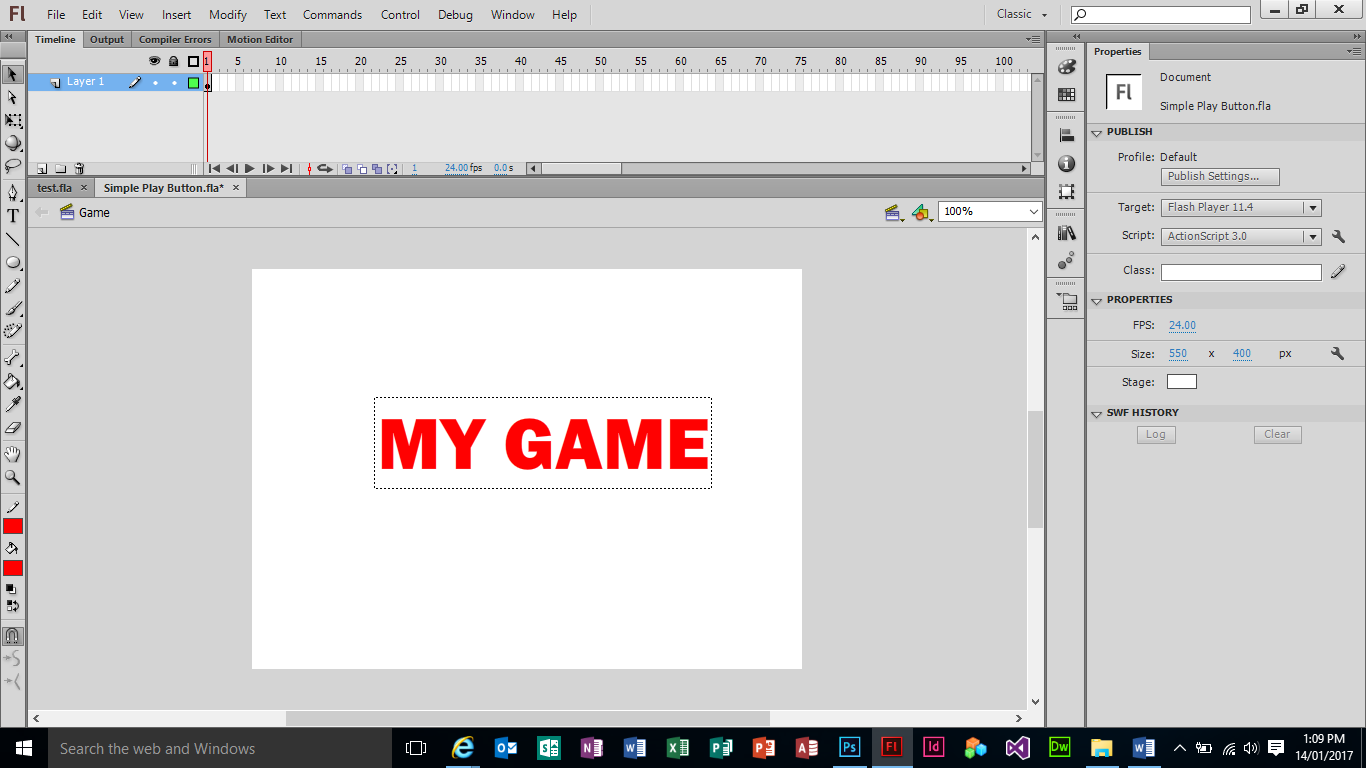
**Create a New Scene**

To create a new scene, press SHIFT + F2 to bring up the Scenes window. Click on the new icon and create a new scene called “Game”. Rename Scene 1 to “Intro”.



Close the Scene window.

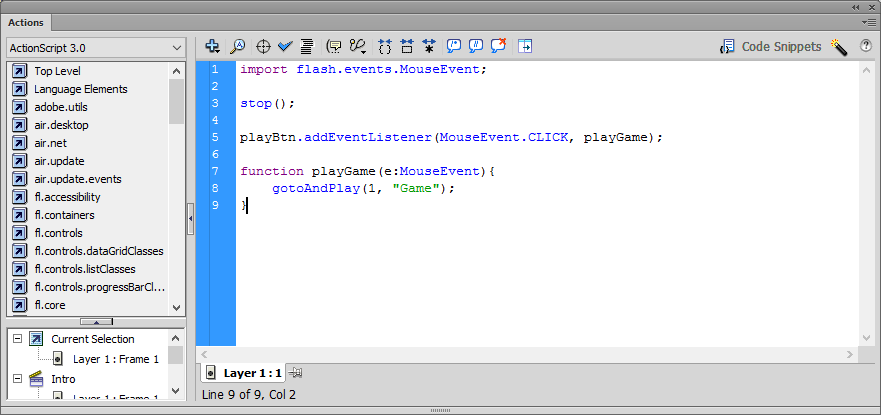
Go to the Game scene and put some text on the Stage that says “My Game”.



**Setting up the Action Script**

Go back to the first scene, Intro. Click on Frame 1 and open up the Actions Window (F9).

Type the following code (you don’t need to type the import statement on line 1 as Flash will do this for you automatically when you type in line 5).

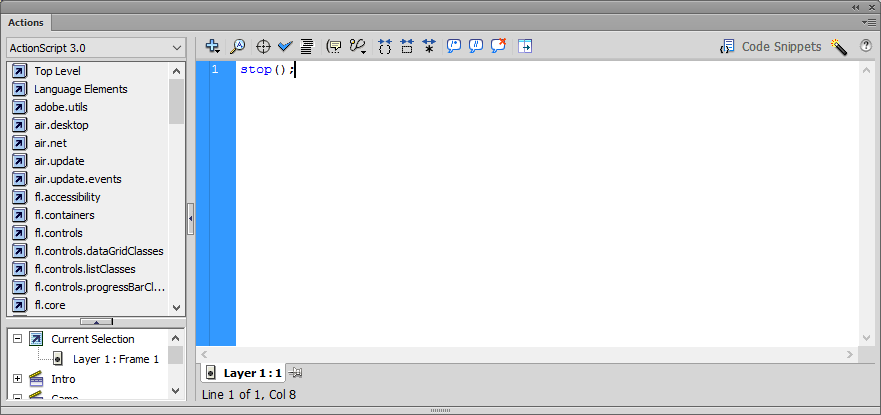


Line 3: the stop() function will make sure the play head stops on frame 1 and waits until the play button is clicked.

Line 5: this code adds an eventListener to the playButton instance. It will listen for a mouse click and then will call the playGame function.

Line 7: this code creates a playGame function which will move the play head to frame 1 of the Game scene and start playing the movie from there.

Close the Actions Window and go to the “Game” scene. Click on frame 1 of this scene and open the Actions Window. Type the following code.



This code makes sure the playhead remains on this frame.

Save and test your game.

## **Using Buttons to Manipulation Movie Clip Objects**

Movie Clips have built-in properties – variables that you can modify to change the appearance of the Movie Clip. These variables include x and y coordinates and the scale or size.

Open up a new Flash Action Script 3 file and save it as “More Buttons.fla”.

Create three layers and name them Labels, Buttons and Background.

On the buttons layer we need to create 4 new buttons. You can use existing buttons in Flash. Click on Window – Common Libraries – Buttons. I have chosen buttons under Class buttons – Aracade buttons. Drag out four buttons and place them along the bottom of the stage.

On the Labels layer, use the Text tool to label the buttons “Left”, “Right”, “+”, and “-“. Make sure the text boxes are Static Text.

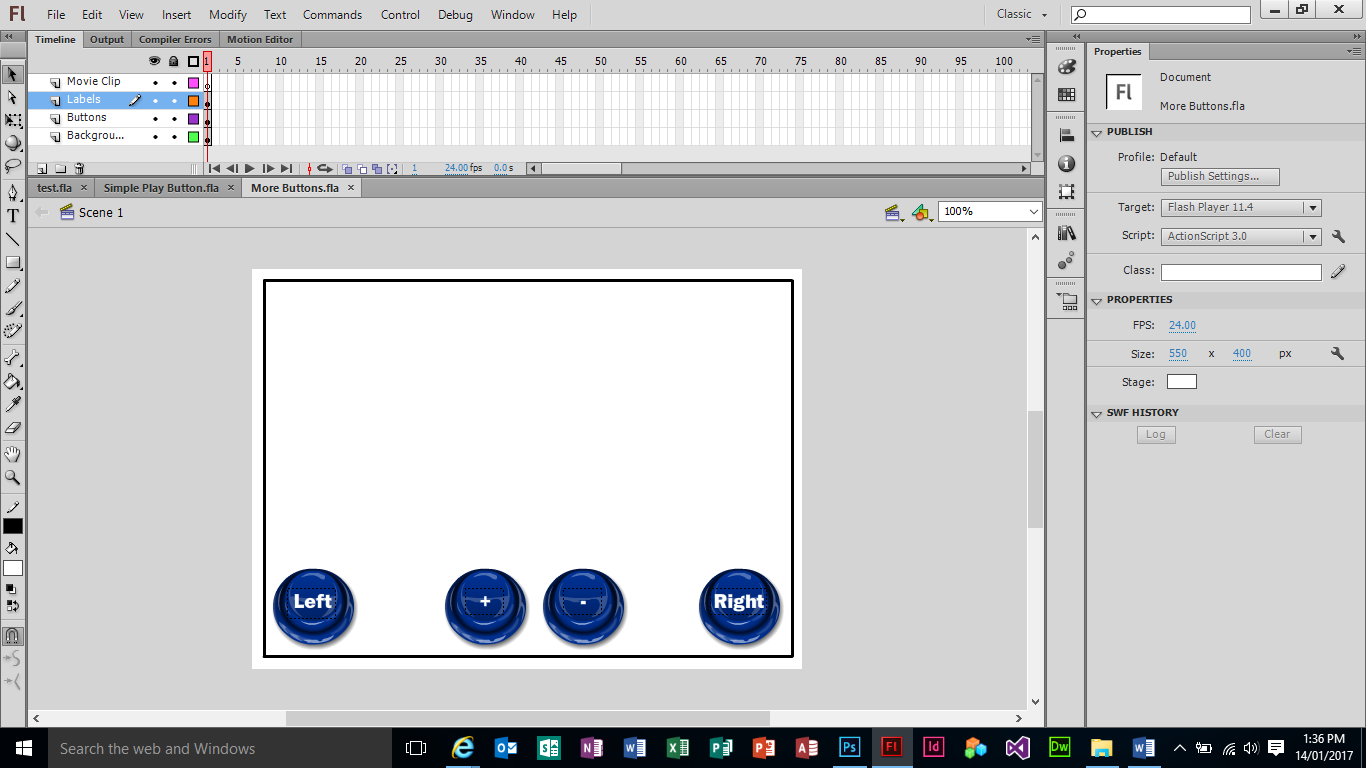
On the Background layer draw a black rectangle (with no fill) as a border just inside the edges of the stage.

Give each of the four buttons an instance name as described below:

* Left Button – leftBtn
* Right Button – rightBtn
* + Button – increaseBtn
* - Button - decreaseBtn

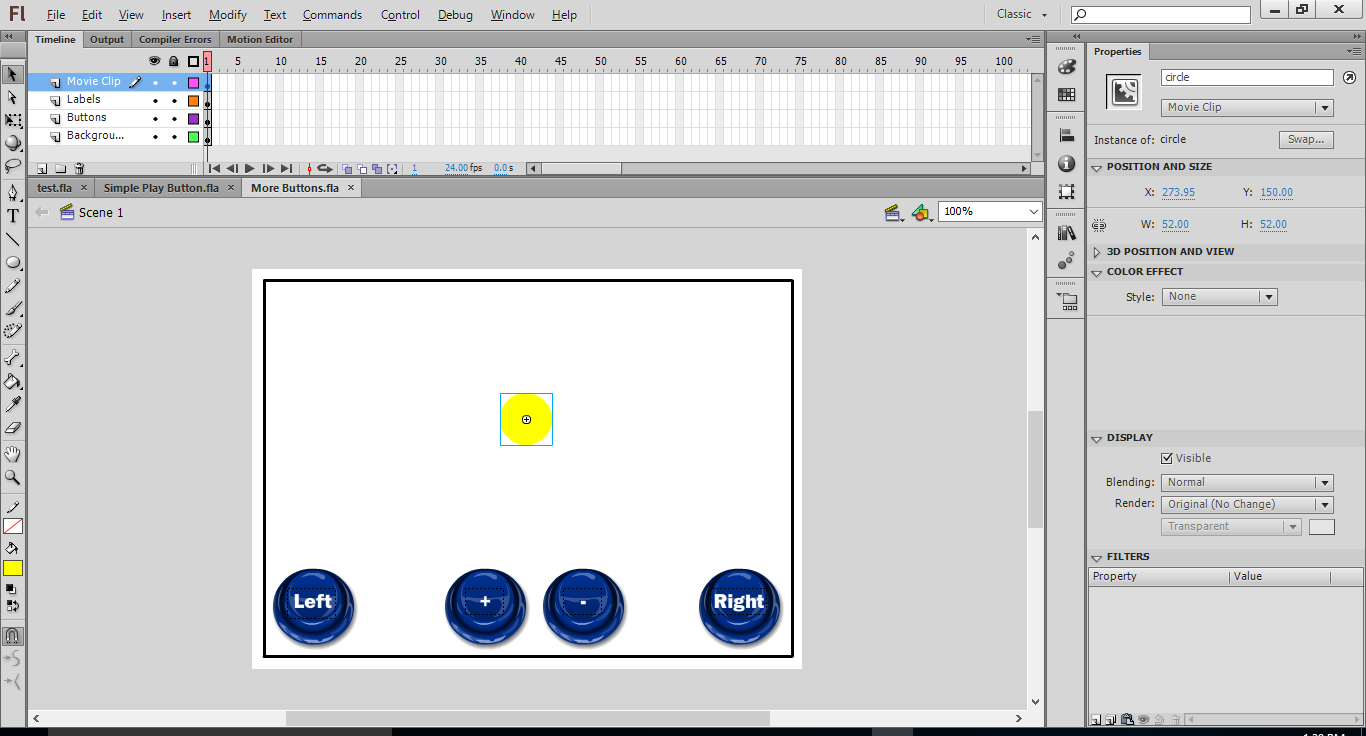
Create a new layer called “Movie Clip”.

Save your file. Your stage should look something like the following:



On the Movie Clip layer create a small circle. Convert this to a Movie Clip symbol called “Circle”.

Give it an instance name of “circle”.

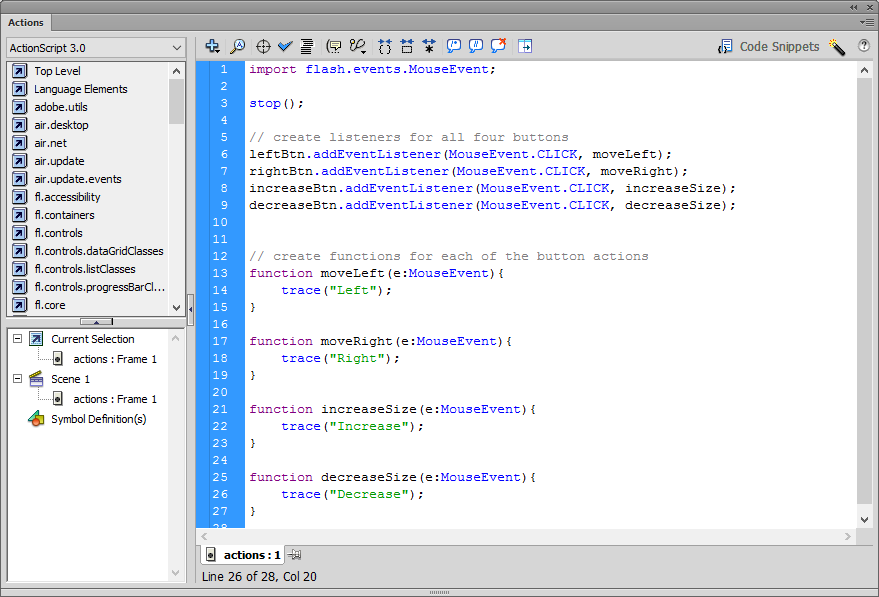


Create a new layer called “Actions” and then open the Actions window.

First, we need to add a stop function to make sure the play head remains on frame 1.

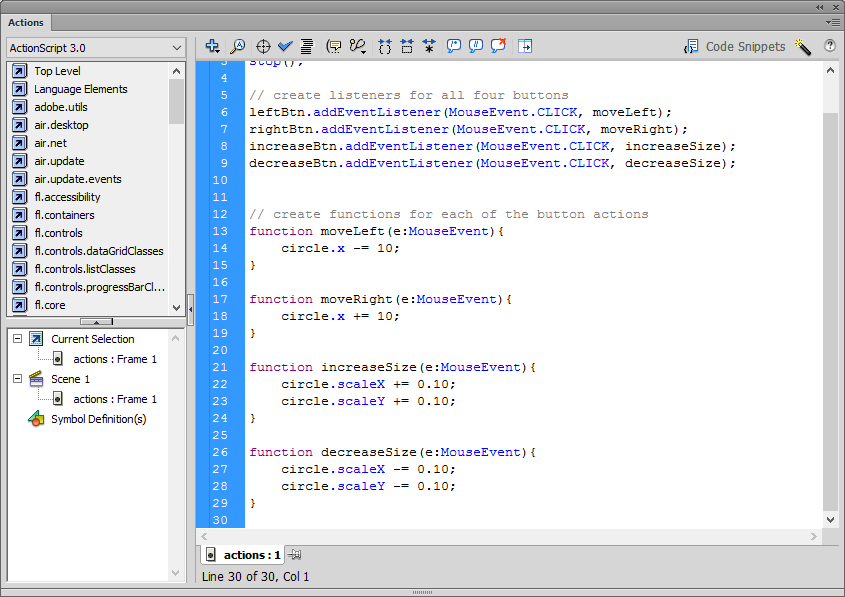
Then we need to add our event listeners for each of our four buttons and then create empty functions which are called by mouse clicks on our buttons.

|  |
| --- |
| Event Listeners are like ears that we can add to objects such as Movie Clips, Buttons and even the Stage. The Listeners (ears) will listen for a particular event such as a Mouse Event (click, mouse down, mouse up). Once the event has been detected it will run the relevant function. |



Save and test your move to make sure the buttons are actually working.

Now we need to add the code to make the functions work.



We can move the circle Movie Clip to the left and right by adding 10 pixels to the circles x and y coordinates.

To increase and decrease the size of the circle we can use the scaleX and scaleY property which increases or decreases the size depending on a percentage you give it (ie 10% would be 0.10).

Save and text your movie. You can now play around with the values to move the circle faster and slower and increase and decrease it at different rates.

### Activity 1:

Add two more buttons to the stage which all the circle to be moved up and down. Add the code to make this happen. Try out the other properties below.

**Other Movie Clip Properties**

|  |  |  |
| --- | --- | --- |
| **Property** | **Values** | **Description** |
| rotation | any value | changes the rotation |
| alpha | 0 to 1 (0 = transparent, 1 = opaque) | changes the transparency |
| visible | true (visible) or false (invisible) | changes the visibility |