

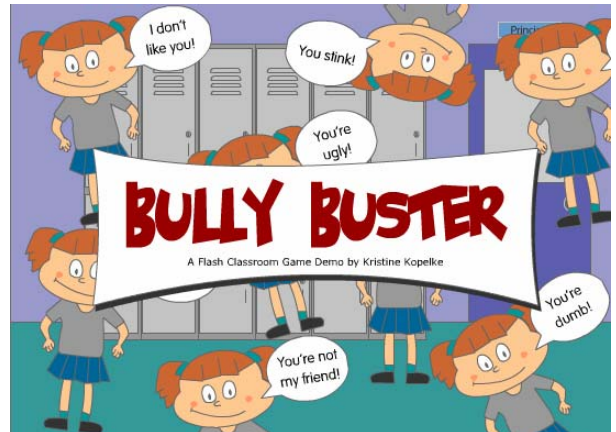


MAKE A SHOOTING GAME WITH RANDOM OBJECTS

In this tutorial, you will learn how to make a shooting game where the targets appear in random places in the game environment.

An example of this type of game is the 'Bully Buster' game that is located in the Game Design section of the Flash Classroom gallery.

In this type of game, the user has to try to hit the moving target to earn points. In the context of the Bully Buster game, the players' goal is to click on the bully in order to 'send them to the Principal'. The player earns points for each bully they 'report' or hit.



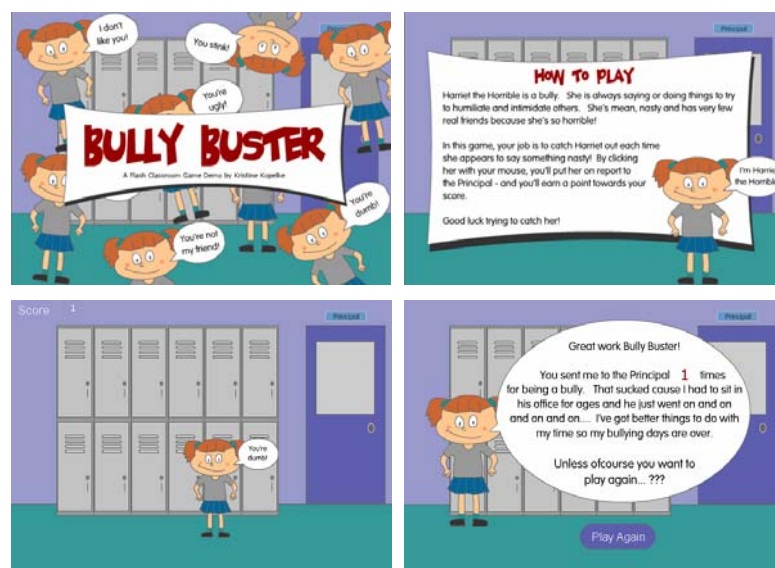
In essence, 'Bully Buster' is a shooting game, however, it was created in order to challenge the mindset that shooting games have to contain blood and guts. Whilst the bullies in the game are mean and nasty, the outcome of hitting them is that they are reported rather than responding to their bad behaviour with violence.

Pre-Game Design Task

1. Take a few minutes now to play the Bully Buster Game in the Flash Classroom gallery.
2. Now that you understand the game play involved in Bully Buster, take a few moments to consider what other games could be created using this same approach. Brainstorm a scenario for atleast one other game that does not contain guns, blood or violence—yet still requires the user to 'hit' a moving object in order to earn points.

Features of the Bully Buster Game

The game you will learn to make in this tutorial will have the same features as the 'Bully Buster' game. This includes a title scene, an instructions scene, a scene containing the actual game and a final score scene (see diagrams below).





PLANNING YOUR GAME

It's time for you to start planning the game that you will build by following this tutorial. As this tutorial only covers a certain amount of content, you will need to ensure that the game you are thinking of building initially only has the features listed below. Although, the sky's the limit with what's possible in game development in Flash, this tutorial will only introduce you to game design—so make sure you remember that your game will only have the features below—otherwise you'll be disappointed.

FEATURES OF YOUR GAME

1. A **Title Scene** to introduce the game. The user will click a button on this scene to enter the Instructions Scene. This scene will feature the name of the game, a picture / animation of some sort to engage the user and the name of the author/s.
2. An **Instructions Scene** where you outline how to play the game. Once the user has read the instructions they click on a button to enter the game.
3. A **Game Play Scene** which contains a background, the object that must be hit to earn points and a dynamic text box which contains the changing score. The movieclip object that is the target will include an invisible button. This object will contain script which makes the object move randomly.
4. A **Final Score Scene** which contains a copy of the dynamic text box, a message for players and a replay button that takes the player back to the Title Scene.

Note that once you have completed the game, you may want to enhance it by doing things such as changing the mouse cursor to an object that suits the context of your game and adding sound to your game.

These enhancements aren't covered in this tutorial. To learn how to do these things, complete the Working with Sound in Flash tutorial and the Changing the Mouse Cursor tutorial. Both are available on the Flash Classroom site.

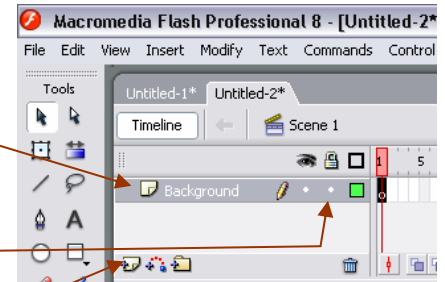
Now that you're aware of the features your game can have, start planning by drawing some storyboards / layouts of each scene on paper. Think of a catchy name for your game and write the instructions for your game. You may even want to do some rough sketches of the characters / objects in your game.

Once you've got all this worked out—you're ready to start creating your game. The following pages will show you step-by-step how to do this.



SETTING UP YOUR TITLE SCENE

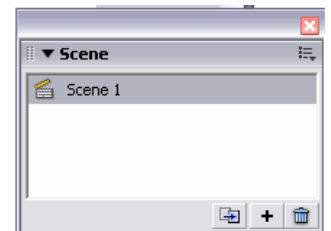
1. Open a new Flash file by selecting **File > New**.
2. Double-click on the text Layer 1 on the first layer of your timeline and type in **Background**.
3. On this layer, **draw** your background picture. Once you are happy with your background, lock this layer by clicking on the dot underneath the lock icon.
4. Make a new layer by clicking on the **Add Layer** button.
5. Rename this layer **Text** by double clicking on the words Layer 2. Add your title and any other text on this layer. **Lock** this layer once you are happy with it.
6. You may want to have a picture of one or more of the games characters / objects on the title page. In the case of my game, 'Bully Bust' I have a picture of one of the bullies. If you do, make a new layer by clicking on the Add Layer button and rename this layer **Characters**. Draw or copy your character onto this layer.



That's nearly everything for the title scene—however, at the moment there isn't any way for the user to get to the next scene—or for Flash to know to stay on the title scene until the user clicks a button. We are going to fix this in the next sections by setting up multiple scenes for the game and adding buttons.

SETTING UP MULTIPLE SCENES

7. To begin with, we're going to open up our Scene panel by selecting **Window > Design Panels > Scene**. This will open the panel shown to the right.
8. The first thing we are going to do is change the name of the scene from Scene 1 to **Title Scene**. To do this, double click on the words Scene 1 and type in **Title Scene**.
9. Now let's set up the other scenes for our game. You have a choice at this point based on the design of your game. In my game, 'Bully Bust' I decided to use the same background on each scene. This meant that when I started setting up my scenes, it made sense to duplicate my scenes so I had the background automatically in every scene. In the other scenes, I simply removed, elements from the Title Scene that I didn't need.

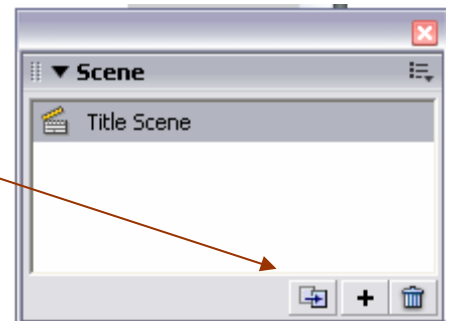


However, you may not want to have the same background on every scene and therefore duplicating the first scene doesn't make sense. Take a moment to think which of these options suit you.



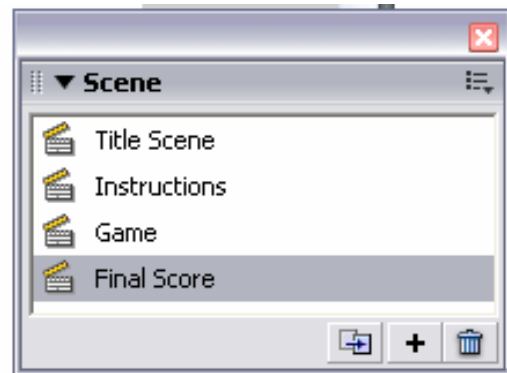
9a. Option 1—Duplicate the Title Scene

If you've chosen to duplicate the title scene, you can do this by clicking on the title scene in the Scene panel and selecting the **Duplicate Scene** button shown here.



If all of your scenes are to have the same background, repeat this process 3 times. If not, repeat it until you have the number of scenes you need with that background.

At this point, your Scene Panel should look something like this. Note that I have renamed all the duplicated scenes by double clicking on them and typing in the new names.



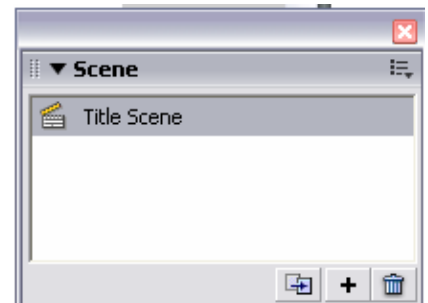
I encourage you to use the same names as I have so that you find it easier to follow the actionscript we write later.

If you have set up all your scenes—go to Step 10.

9b. Option 2— Add New Scenes

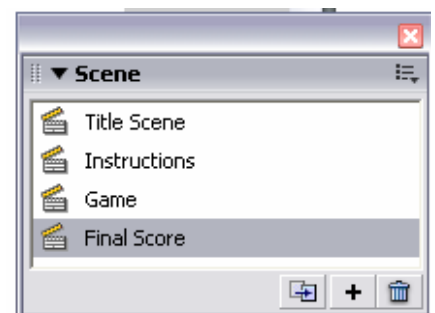
You may want each scene in your game to have a different look. That's fine too and if you choose to do this, you will now simply add blank scenes that you can design new backgrounds etc for later on.

To **add a new scene**, click on the + button at the bottom of the panel.



Rename the scenes you add by double clicking on their names and typing in your own names. I encourage you to use the same names as I have so that you find it easier to follow the actionscript we write later.

Your scene panel should look like this once you have added all four scenes.





ADDING BUTTONS TO MOVE BETWEEN EACH SCENE

In the game we are creating, we need to add some buttons to enable the player to move between scenes.

In the 'Bully Buster' example, there are 3 buttons which make this possible. These are:

- 2 x Invisible full screen buttons that are located on the top layer named buttons in the **Title Scene** and the **Instructions** scene.
- 1 x Replay button which is located on a layer named buttons in the **Final Score** scene.

This page will walk you through how to create those buttons. The design of the buttons and type of buttons you use is up to you—you may even wish to use a button from the Common Library that Flash provides. To view the buttons in this collection, select **Window > Common Libraries > Buttons**.

The instructions below show you how to make your own buttons.

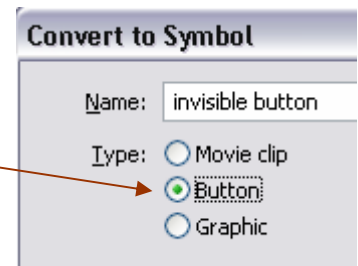
Making Invisible Buttons

10. Make a new layer in the Title Scene and name it Buttons. To do this, click on the **Add New Layer** button at the bottom of the timeline. Double click on the name of the layer and enter **Buttons**.



11. On this new layer which should be at the top of the layers, use the **rectangle drawing tool** to draw a large rectangle over the entire stage. (Note that if your buttons layer isn't at the top, you can click on it and drag it above the other layers).

12. Convert the rectangle to a button by selecting **Modify > Convert to Symbol**. Name your button **Invisible Button** and select the button behaviour.



13. To make the button invisible, click on the button and then right click to bring up the context menu. In this menu, select **Edit in Place**.

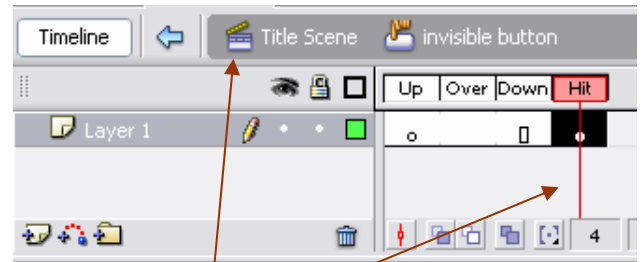
14. You are now editing the button symbol. You will see that the timeline for this button symbol only has four frames—one for each of the button states.

The states are **UP** - when the mouse isn't over the button
OVER - when your mouse is over the button
DOWN - when you click on the mouse
HIT - the active area that is the button.



15. Your button isn't going to have an up, over or down state. It will just have a hit state that contains the rectangle. This will enable the user to simply click their mouse in any location to go to the next scene.

By default, the rectangle will be sitting in the Up state. To move it to the **Hit** state, simply **click** on the keyframe in the Up state and then click again and drag the keyframe into the Hit state. Alternatively, click on the rectangle and cut and paste it into the **Hit** state frame. The timeline within your invisible button symbol should look like the one above.



16. Once you have done this, return to edit the **Title Scene** by selecting the **Title Scene** link just above the timeline.
17. You should now find that your invisible button symbol has turned a transparent aqua colour. This is normal and represents that this button is simply a hotspot where the user will click—it won't actually be visible in the final movie. If you want to edit layers underneath the buttons layer, simply lock this layer and make it invisible by clicking on the dots underneath the lock and eye symbols in the timeline.

You have now made an invisible button. You may wish to use this as I did to link from the Title Scene to the Instructions Scene and from the Instructions scene to the Game Scene. We will look at programming the buttons on the next page. Before we do that, I'll show you how to make a Play Again button similar to the one in the 'Bully Buster' example.

Making Your Own Button

18. On the grey area to the side of the stage, draw what you want your button to look like when the player does not have the mouse over it. This is the **Up** state.
19. Select your button and select **Modify > Convert to Symbol**. Name your button and select the **Button** behaviour.
20. Click on your button and then right click to bring up the context menu. From this menu select **Edit in Place**.



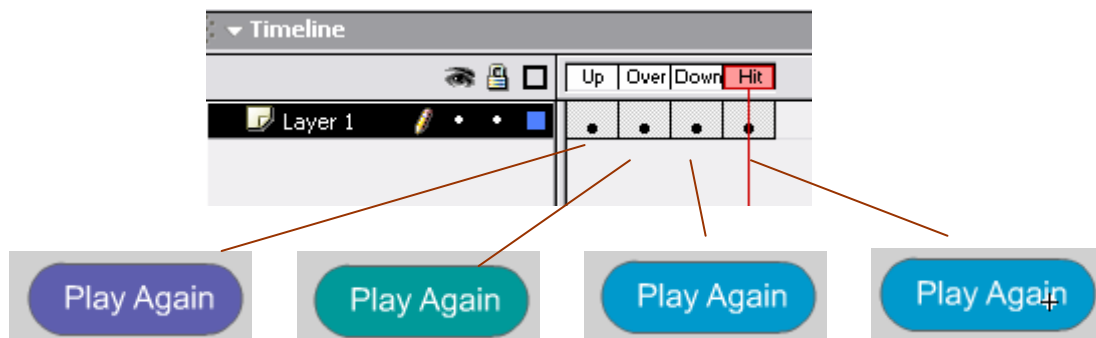


21. You should notice that when editing your button, the timeline will only have a keyframe in the **Up** state. The button will still work when this is the case, however, you may want to change what the button looks like in the other states. For example, it is common for a button to move slightly or change colour when rolled over.

Add **Keyframes** to each of the other button states (**Over, Down** and **Hit**) by clicking on each frame and selecting **Insert > Timeline > Keyframe**.

In each keyframe, change what you want the button to look like. The hit state keyframe usually doesn't need to be edited. This state is usually only needed if you want to create an invisible button or say you have an image or text in the other states that is intricate. When this is the case, you may wish to draw a bigger shape over the text or image in this state so that the button is easy for the user to hit. For example, if you didn't do this for a button containing only text, the user would actually have to put their mouse over the lines of the letters to make the button work. Drawing a shape in the hit state makes the button easier to hit.

Below is a diagram that shows the four hit states of the Play Again button I created for the 'Bully Buster' game.



22. Once you are happy with your button states, click on the Scene name link at the bottom of the timeline to stop editing the button and return to the current scene.

Programming Your Buttons to Work

It's now time to position the buttons on the button layers in the appropriate scenes and to add the actionscript that will make them go to another scene when clicked. Follow the steps below for each of the buttons to do this.

23. Position the button you want to program in the location you want it on the stage. To make multiple versions of the same button, simply copy and paste or open the Library using **Window > Library** and drag a copy onto the stage.
24. Click on the button you want to add the script to and select **Window > Actions** or press **F9**. This will open the **Actions Panel** where you will type in the script on the following page.



25. Type the following script into the Actions Panel.
Note that the scene name has been put in **blue bold** to remind you that you will need to edit this name in the next step.

```
on (release) {  
    gotoAndPlay("Title Scene",1);  
}
```

This script tells Flash that when the player clicks on the button and then releases it, that Flash needs to go to and play another scene. In this case, the Title Scene. The number 1 indicates that it should go to frame 1 of that scene.

26. Edit the script by changing the name of the scene to the name of the scene your button is going to take the player to.

Repeat this process for each of your buttons.

Adding Stop Actions to Scenes

If you were to test your movie now you would find that your movie automatically jumps from one scene to another. This is because Flash naturally plays through scene 1 and then moves to scene 2 and so on, unless it is instructed to stop.

We are going to add **stop actions** to the final keyframe in our **Title Scene**, **Instructions** scene and **Final Score** scene. This will mean that the Flash player will stop at these scenes until the user clicks on the buttons you have added in the previous step. Note that we don't want to add a stop action to the **Game** scene as we want this scene to end by simply jumping to the **Final Score** scene.

27. To add a stop action to the final keyframe of each scene, click on the keyframe and press **F9** to open the **Actions Panel**.
28. Type in the stop action shown below:

```
stop ();
```

This will trap the user in the current scene until they click on the button to go to another scene.

Repeat this process for each scene the user should be stopped at.

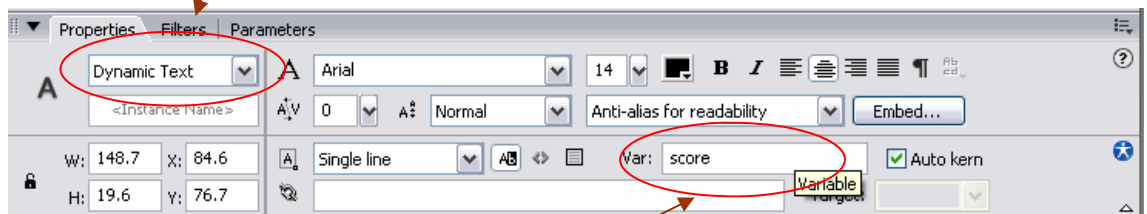


Building Your Game Scene

You have now completed most of the steps involved in setting up the different scenes and the navigation between them. In all of the scenes except for the **Game** scene, you really just need to change the content on the layers to include the text and objects you want. This section of the tutorial will show you how to create the dynamic text box which will show the score and how to create the movieclip targets that when hit will change the value of the score.

Part A— Setting up the Score

29. Add a new layer in the **Game** scene by selecting the **Add New Layer** button at the bottom of the timeline.
30. Rename this layer **Score** by double-clicking on the Layer name and typing in score.
31. **Draw a text box** in the location on the screen where you want your score to appear. In the properties panel, change the font size and colour of the text to make it look how you want it to. Type a zero into this text box.
32. Change the drop down menu shown at the top left corner of the properties panel to the **Dynamic Text** setting (see diagram below).



33. Type the word **score** in the Var (variable) cell located in the bottom right hand corner of the properties panel.
34. Add the script **var score:Number = 0;** to the first keyframe in the current scene. This will initialize the score variable.

Explanation

What you have just done in the previous two steps is set up a variable for the score that we have called **score**. If you are new to variables, a variable is like a container that contains different values. Our variable or score will change because we will add script to a button in our target movieclip (bully) that instructs the Flash player to add a certain amount to the score when clicked.

35. The final scene of the game that you named **Final Score** will also contain the score. At this point, copy this text box and paste it into that scene so it is there for later.
36. You may wish to add a text box next to the score variable that contains the word score as I have done. Ensure for this text box you change Dynamic Text back to **Static Text** as this word won't change.



Part B — Designing the character to be hit.

It's now time to create the target character or object that the player must hit. In the 'Bully Buster' game, the target is 'Harriet the Horrible' (pictured here).



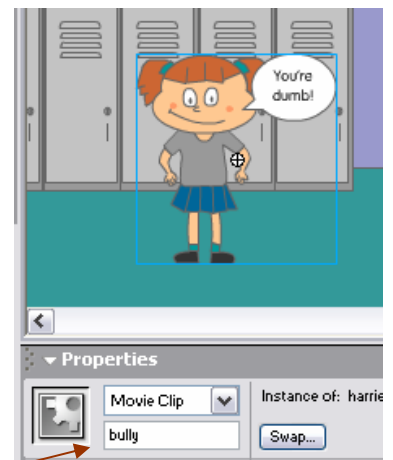
In this part of the tutorial, we are going to draw our target character / object and then convert in to a movieclip symbol. We will then edit this symbol so that we can add a button that will be the hit area for our object. This button will contain the script that tells the Flash Player to increase the score variable by one each time the object is hit.... Confused?? Don't worry—just follow the steps below..

How to create your object and give it an instance name.

37. In the grey area beside the stage, **draw the object or character** that the user will have to hit to earn points.

38. Once you are happy with the object or character you have drawn, convert it to a movieclip symbol by selecting it and choosing **Modify > Convert to Symbol**. Make sure you have the **Movieclip** behaviour checked and give it a name. (In my game, I named the object 'Harriet').

39. Now that you have converted your object or character to a movieclip symbol, you can drag it onto the stage area. Make sure your object or character is selected and in the **Properties** panel enter an instance name in the cell shown here. I have given my movieclip symbol the instance name **bully**.



How to make your object a moving target that will earn the player points.

40. Select your object / character and right click on it. Select **Edit in Place** from the menu.

41. You are now editing the movieclip. Select your character / object and select **Modify > Convert to Symbol**. Name your symbol **target** and check the **Button** behaviour.





42. Make sure the **target** button you've just made is selected. Select **Window > Development Panels > Actions** or press **F9**. This will open the **Actions Panel** where you will type the script that will add points each time the object is hit and also make the characters
43. Type the following script into the Actions Panel.
Note that the object name (bully) has been put in **blue bold** to remind you that you will need to edit this name in the next step.

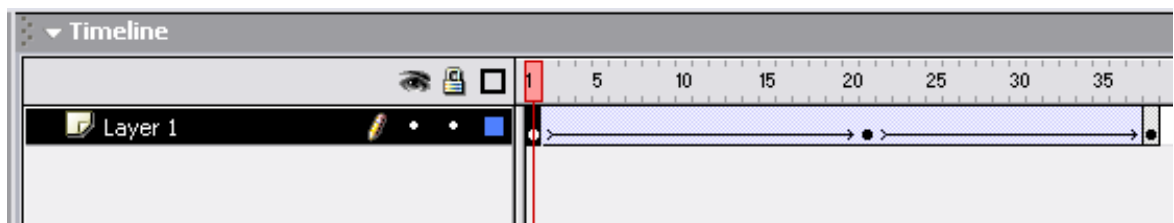
```
on (release) {  
    _root.score = number(_root.score + 1);  
    _root.bully._x = number(random(550) + 20);  
}
```

This script tells Flash that when the player clicks on the button and then releases it, that 1 point is added to the existing score. It also tells Flash to make the bully appear in a new x coordinate location - this new location is determined randomly.

44. Edit the script by changing the object name to the instance name you gave your movieclip symbol.

Make your target harder to hit by using a motion tween to animate the button.

45. Though not essential, you can make your object harder to hit by creating a motion tween. In my 'Bully Buster' game, I used a motion tween to make the target button move up and down. If you have not created a motion tween before, you may want to take a ten minute break and work through the **Create A Basic Motion Tween Tutorial** on the Flash Classroom site.





Adding some actionscript to track the number of moves the object / character has made.

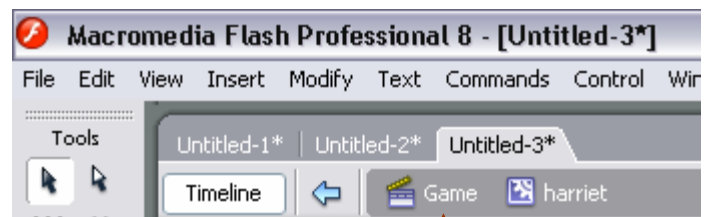
46. Within the character / object movieclip symbol you have created, you need to click on the **first keyframe** and add the following actionscript to the **Actions Panel**. If your actions panel is closed, select **F9** to reopen it.

```
__root.moves = number(__root.moves + 1);
```

This script adds one to a variable called moves each time the movieclip symbol is moved to a new x coordinate location.

Return to the Main Stage

47. You have now completed all the 'inside' work on your object / character movieclip.

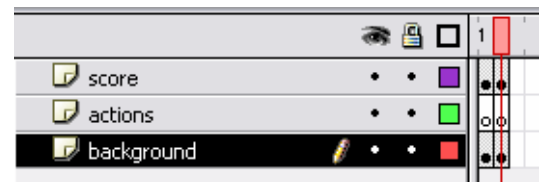


To return to the main stage, select the **Game** link on the bar directly above the stage.

Adding a second set of keyframes to the timeline

On the timeline in the Game Scene, we are going to insert new keyframes into the second frame of the timeline. Within one of these new keyframes we will place an if statement in order to make the game end after the character has moved 15 times.

48. On each layer of the timeline, add a new keyframe to the second frame. To do this, click on each layers second frame and select **Insert > Timeline > Keyframe**. Your timeline should look similar to this.



49. Now we are going to add the actionscript to the second keyframe on the Actions layer. Click on this keyframe and open the **Actions Panel** by selecting **F9**. Type in the following script:

```
if (moves eq 15) {  
    gotoAndStop ("Final Score",1);  
} else {  
    gotoAndPlay("Game",1);  
}
```

This is an **if else statement**. This statement tells the Flash Player to go to the Final Score scene if the character has moved 15 times. If the character hasn't moved 15 times, the script tells the player to go to and play the first frame of the Game scene.



Adding the final piece of actionscript to reset the score if the user chooses to play again.

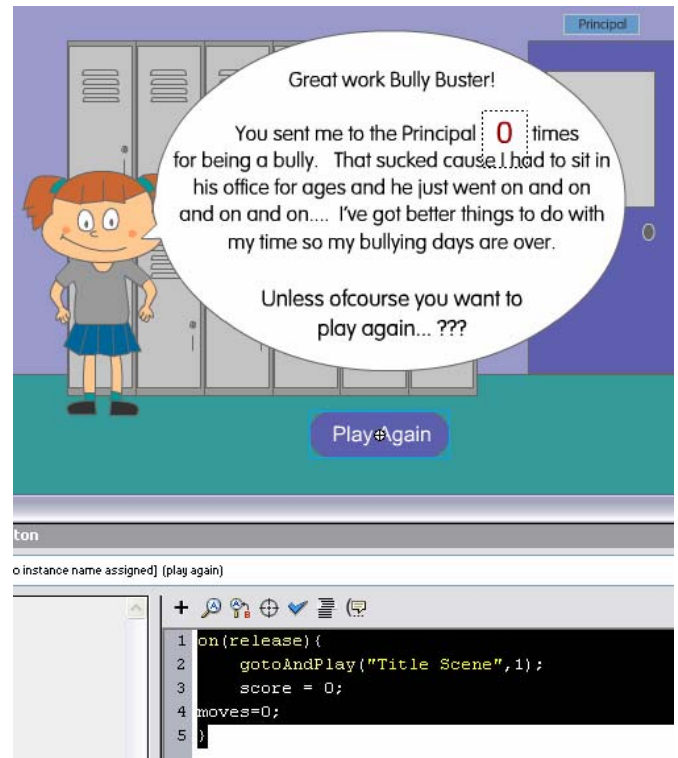
50. The final part of the actionscript we need is added to the script that is attached to the Play Again or Replay button you have created in the Final Score scene.

Select this button and click **F9** to open the **Actions Panel**. Type the following script in.

```
on(release){
gotoAndPlay("Title Scene",1);
score = 0;
moves=0;
}
```

This script tells the Flash Player that when the player clicks this button and releases it, it should return to the Title Scene.

The lines of script with `score = 0;` and `moves = 0;` tell the Flash Player to reset these two variables to 0. If you didn't add these two lines of script, your score would just keep adding to the existing score. It would also mean that the game wouldn't end as the script that makes the Game Scene move on to the Final Score Scene is set to work when the moves variable equals 15. This wouldn't happen the second time you played as the variable would be past 15. Therefore, these two small lines of script are an essential part of your game.



The Final Touches

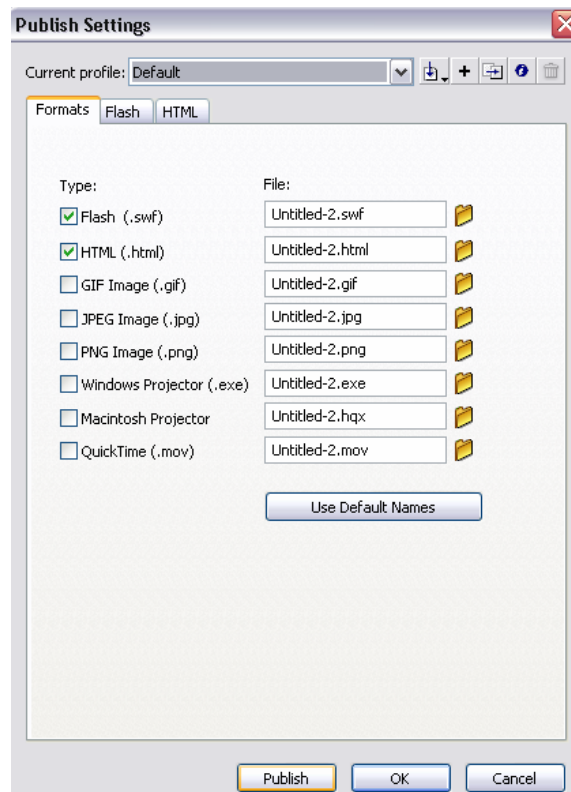
If you have followed these instructions correctly, you should now have a game with four scenes. You should take the time now to make sure you're happy with the design and layout of your game. For example, in the Final Score scene, you may wish to put the dynamic text box score variable within a sentence as I have done. Alternatively, you may wish to just write the words Total Score next to it. The quality of your game will depend on how much effort you put into your concept and graphics so don't be slack at this point. You've done all the hard work and coding—so now take some time to do the finishing touches.

Once you're happy with your game, select **Control > Test Movie**. This will allow you to play the game before exporting it by following the instructions on the next page.



PUBLISH & SHARE YOUR WORK

26. Save your work by selecting **File > Save**.
27. Turn your flash file into a game that can be played on any computer by publishing it in different file formats. To do this select **File > Publish Settings**. The box shown below will appear.



28. Tick the file formats you want and click on the **Publish** button. These files will be saved in the same location you saved your original file. If you want your game to be a standalone file that can be played on Windows or Macintosh machines—ensure you check the Windows Projector (.exe) and Macintosh Projector (.exe) format options.

ENHANCE YOUR GAME

If you've got to this point, why not learn more about game design and how to enhance your game with a customized mouse or sound by completing the following Flash Classroom Tutorials

Working with Sound Changing the Mouse Cursor tutorial