

Name: Jason Gunn

Number: 06039966

**Course: Games
Computing**

**Unit: Creative
Technologies**

**Unit Lecturer: John
Hamlin**

Unit Code: 1014

Blog 1 – Introduction

A Note:

For this assignment I was lucky enough (Or not) to be able to use Adobe Flash CS3, however I have found that the .fla files will sometimes not work on older versions of flash, I have saved my .fla in both CS3, and Flash 8 formats, however some of the text is not available in earlier versions, so is not in line with how it should look. However my movie file shows correctly, with the correct fonts, sizes and positioning.

Introduction:

This assignment was an interesting one for me, I have always wanted to learn to use flash, but never had the incentive to do so, I have wished to make small animations, website introductions etc. As I had not used flash before, this was quite a steep learning curve, however thanks to its similarity to previous Adobe products, such as Premier and Photoshop, I found it relatively easy to adapt to its features, and soon realised that the ideas I had were actually possible, without having to simplify them to meet the course deadline. I will also mention, that all my assets were imported into my library, before being used, this is an easy process, you can simply drag and drop assets into the library.

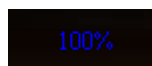
Blog 2 – Planning

My Idea:

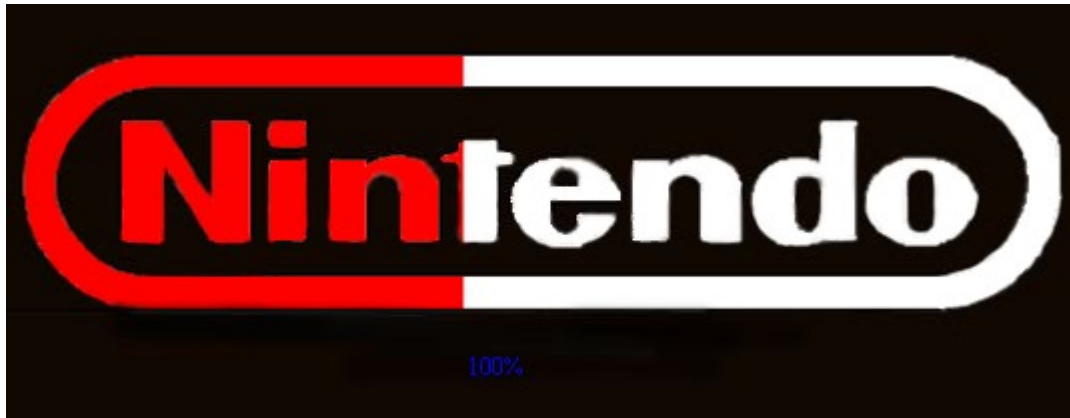
For my assignment I went through several ideas, for several different games, however I decided on the recent (and brilliant) Zelda - Twilight Princess from Nintendo. I had the idea in my head, I wanted a preloader that filled up the Nintendo logo as it loaded, an animation of Zelda turning to the camera, but I was not certain on the design of my interface.

Blog 3 – The Preloader

My original plan for the preloader, was to have it filling up gradually, to show the progress of the loading, however I decided to include a percentage bar:



Which fills up as the file loads, I also set it so that each update to the percentage bar would restart the filling up animation of the Nintendo logo: :



I tried to create several different versions of the preloader, however found this to be the best method. :

To create it I looked at several guides, most notably the following: :

<http://www.kirupa.com/developer/mx/percentagepreloader.htm>

I then followed the guides, and made several different preloaders, until I was confident that I could create the custom one I wanted! :

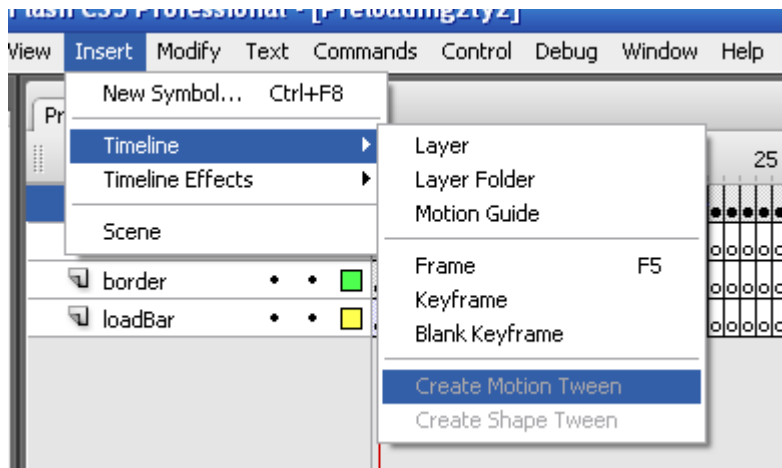
I used similar code to that given in the guide, however modified it slightly to work as I wanted. :

My first step was creating a key frame, and then importing the Nintendo logo into it (I first edited it in Photoshop, and gave it transparent areas). :

Next I created a loading bar, and set it to the size I wanted: :



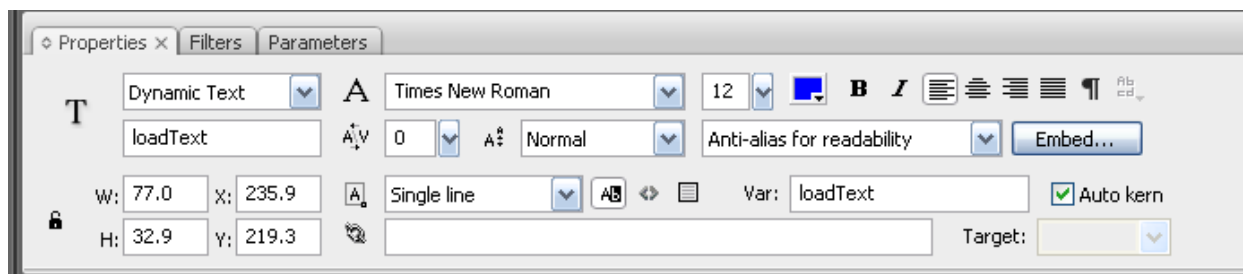
After this I created a new key frame at frame 20, and this allowed me to click on frame 19, and select Insert>Timeline>Motion Tween. :



After this I moved the loading bar to the position I wanted, and tested that it worked, it did! :



I then created a text area, giving it the name loadText, the variable loadText, and setting it as a dynamic text :



After this I created another key frame after my loading bar, just to contain the final preloader frame, and end code for my preloader, in frame 1 I inserted the code :

```
stopAllSounds();
```

```
bytes_loaded = Math.round(this.getBytesLoaded());
```

```
bytes_total = Math.round(this.getBytesTotal());
```

```
getPercent = bytes_loaded/bytes_total;
```

```
this.loadText = Math.round(getPercent*100)+"%";
```

```
if (bytes_loaded == bytes_total) {
```

```
    this.gotoAndPlay(start);
```

```
}
```

This code will stop all sounds, if for some reason the user manages to loop back round to the start of the preloader, and check how many bytes are loaded, displaying it in percent form in my text box, if it is loaded it will start my animation, else it will loop again.

In frame 20:

```
}
```

```
if (bytes_loaded == bytes_total) {
```

```
}
```

```
    this.gotoAndPlay(start);
```

```
}
```

```
}
```

```
}
```

```
else {
```

```
}
```

```
gotoAndPlay(1);  
  
}  
  
}  
  
}
```

This is again another check to make sure the movie has loaded 100%, if not it will loop back to the start of the preloader, and display the current percent status, of loading.

I then tested to see if this all worked, and found that it did indeed load until the entire file had been loaded.

Reflection:

I am quite pleased with the way my preloader turned out, however feel it could have been better, had I made for example the loading bar, fill up as the movie loads, rather than in stages. However overall it works, and does what I had planned.

```
this.gotoAndPlay(start); br />  
  
}  
  
else {  
  
gotoAndPlay(1);  
  
}
```

This is again another check to make sure the movie has loaded 100%, if not it will loop back to the start of the preloader, and display the current percent status, of loading.

I then tested to see if this all worked, and found that it did indeed load until the entire file had been loaded.

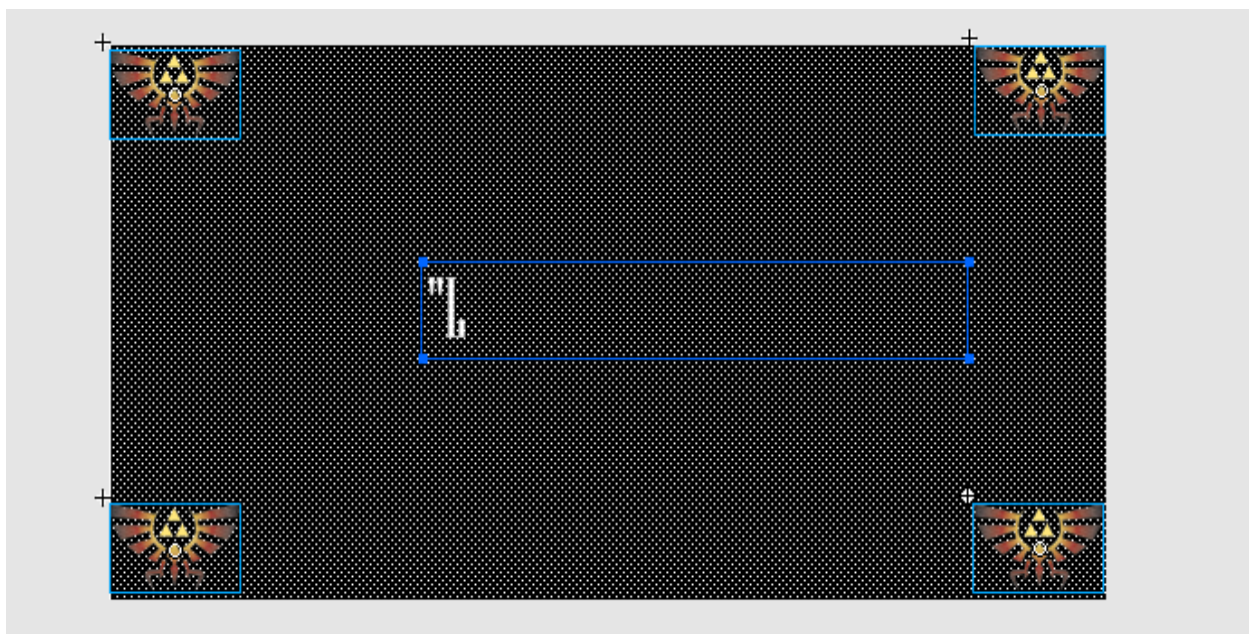
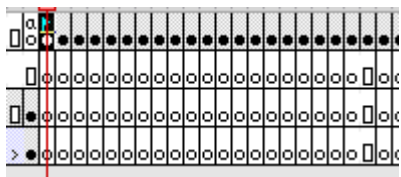
Blog 4 – The Animation

For my animation, I went through several different stages, I tried first to import my movie as a sequence of images, however found this to make the overall file size to large, so exported it from premiere as a movie file, and inserted it as an embedded object into my stage.

I then tried putting it in a frame, however this looked very bad, so I created a backdrop, and made the rest of my animation match it:

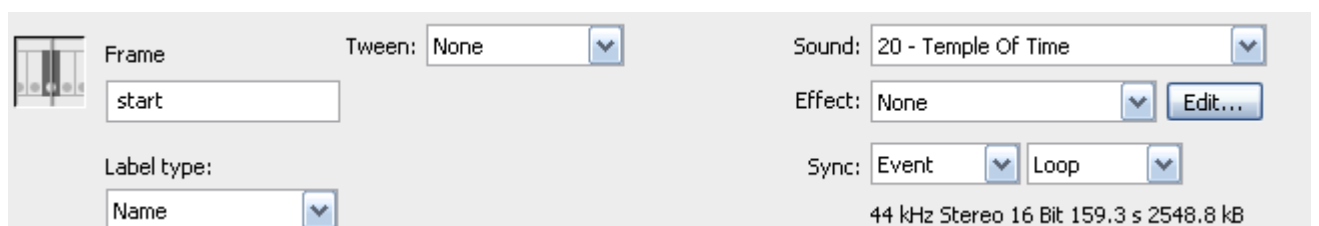


I then decided to add some animated text, to do this I would create a frame, write a letter, then create the next frame, adding the next letter, etc until it was complete:

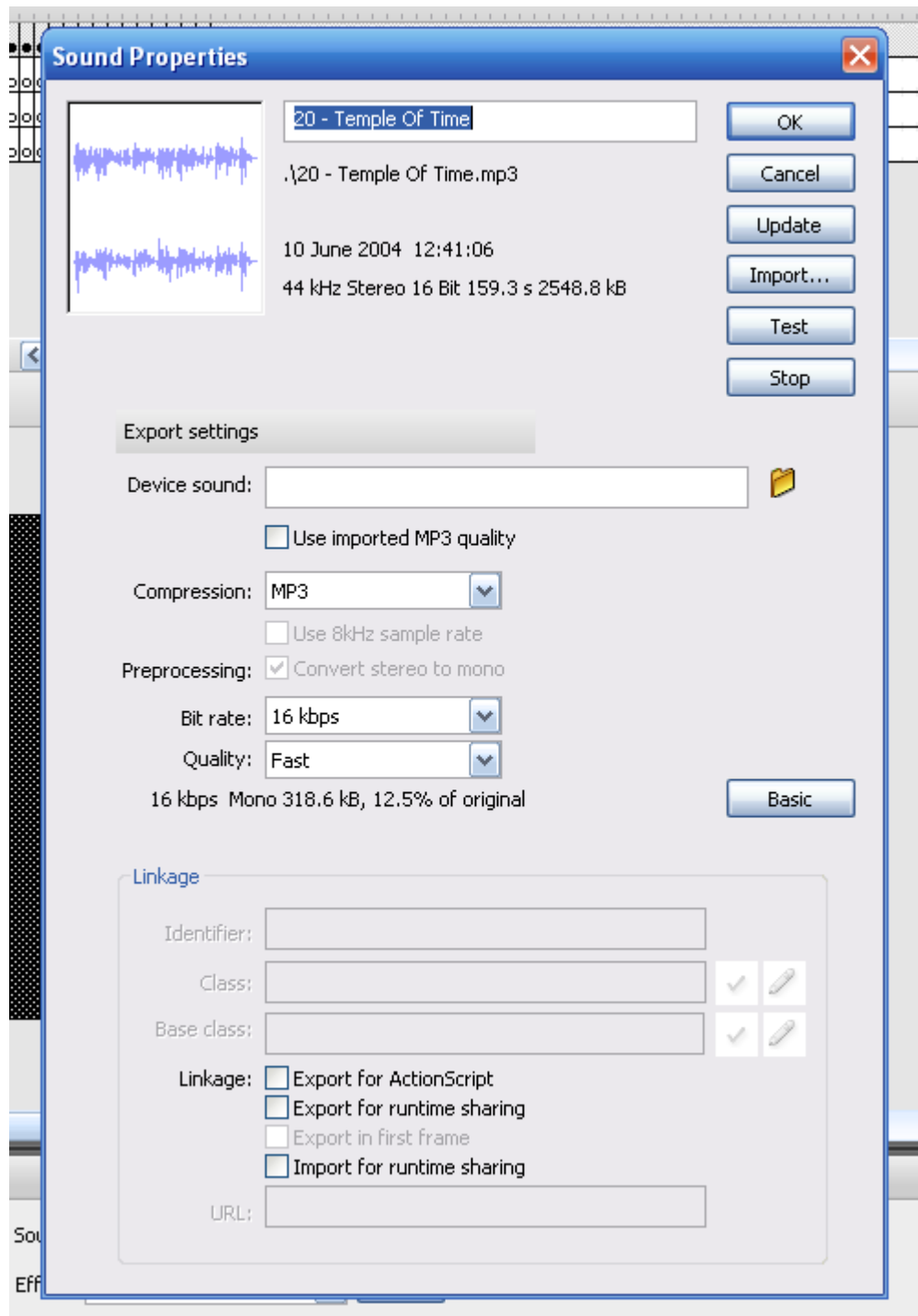




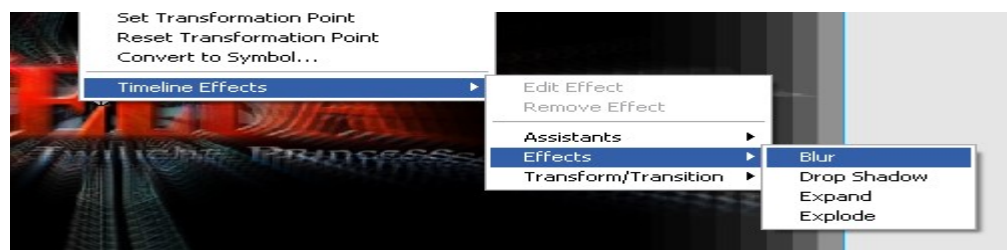
After I had got the section working, with animated text, and embedded movie file, I then inserted a sound into my work, so that it would started playing just after the preloader had finished loading. I set this to loop so that there would be a continues background music throughout the users experience.

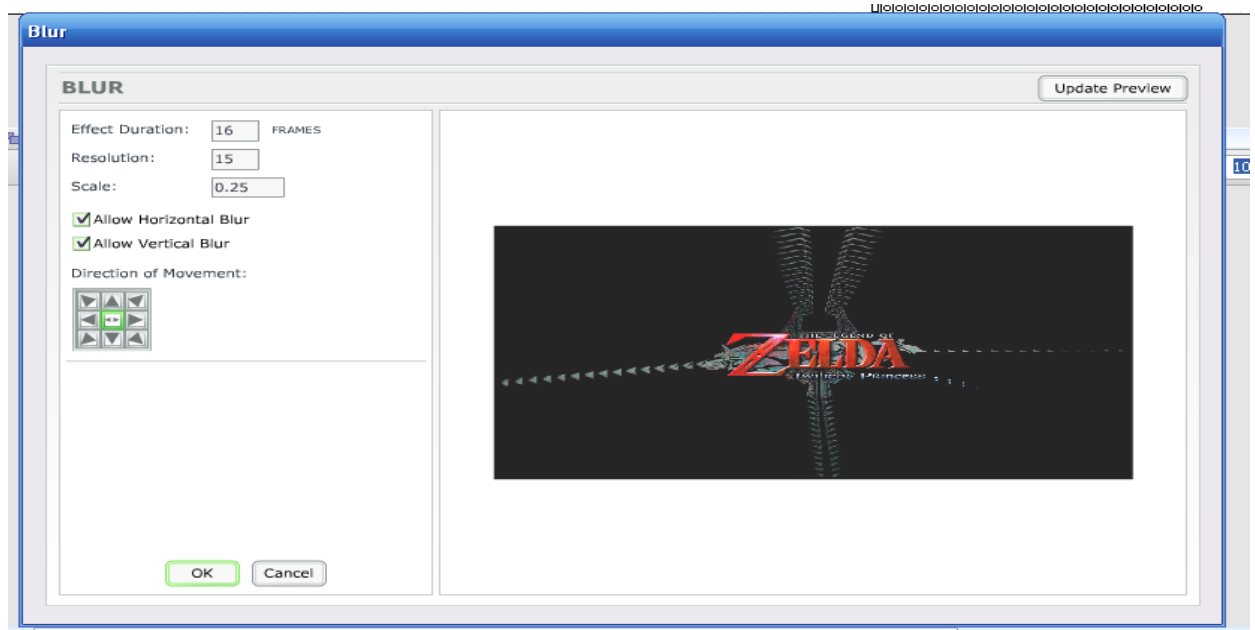


I also compressed the music to make the file smaller.



Finally, I found, and resized a game logo, and added the blur effect, I found this to be quite effective, and a good way to merge into the actually interface:





Reflection:

I am quite pleased with the way my animation is shown, it meets all the criteria I had set myself, with the use of tweening, and effects, it is simple to look at, as well as quite short, however I would have liked to improve it by having a higher quality animation, in place of the low quality movie clip

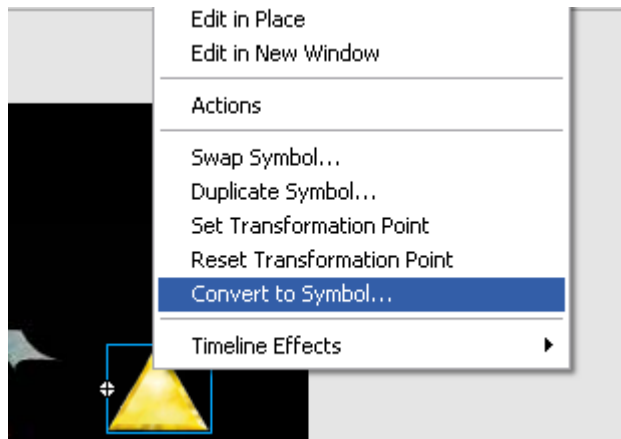
Blog 5 – The Interface

After much planning, I decided to go with a simple interface, that matched the theme of the flash project, a black background with the Zelda logo, and Nintendo logos on. I then decided for my buttons I would use shards of Triforce (Item from the games).

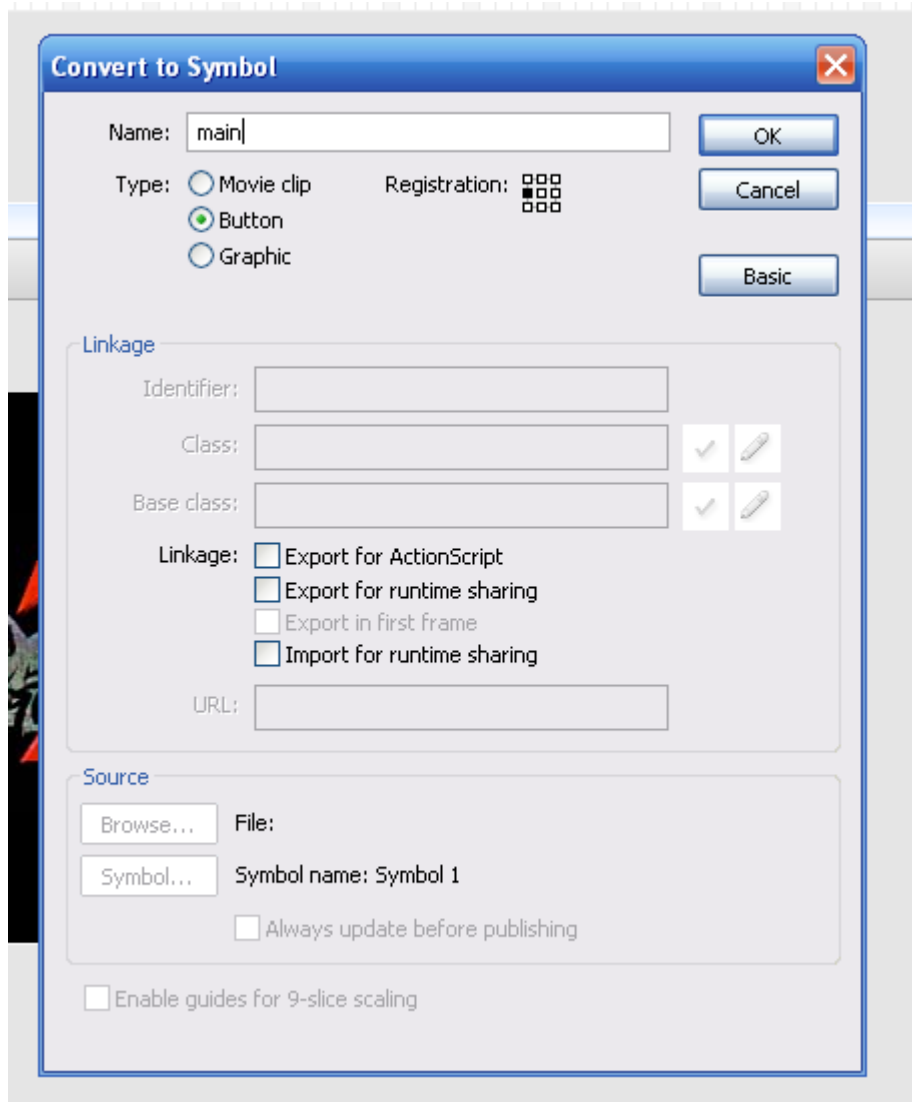
I created my buttons in Photoshop, and then for button down shaded it a darker gold.



Next I created button objects in Flash, To do this I converted the image to a Symbol:

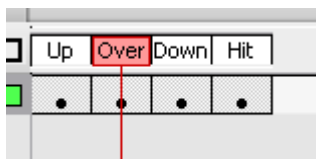


I then gave it a name, selected button as its type and clicked ok:



After this I double clicked the button, and added an action for each section:

Up, Over, Down, Hit



For Up I used the default button, I had created.

For Over, I used the down button I had created, and added an information box:



For Down I used the same images from the frame before, as I did not want this to change.

For Hit, I selected just the button as the animation, as this defines the button click area, and thus I did not want the button activating if the user clicked else where.

I repeated this same process for each button, until it was finished.

I then added the following action scripting to each button:

Button - Main

```
on (release) {
  getURL("http://www.zelda.com/tp/", "_blank");
}
```

Button - Review

```
on (release) {
  getURL("http://uk.wii.ign.com/articles/746/746691p1.html", "_blank");
}
```

Button - Buy

```
on (release) { getURL("http://www.play.com/Games/Wii/4-/1027911/The-Legend-Of-Zelda-Twilight-Princess/Product.html", "_blank");
}
```

This gets the URL I wish the button to link to, and then opens it in a browser window for the user.

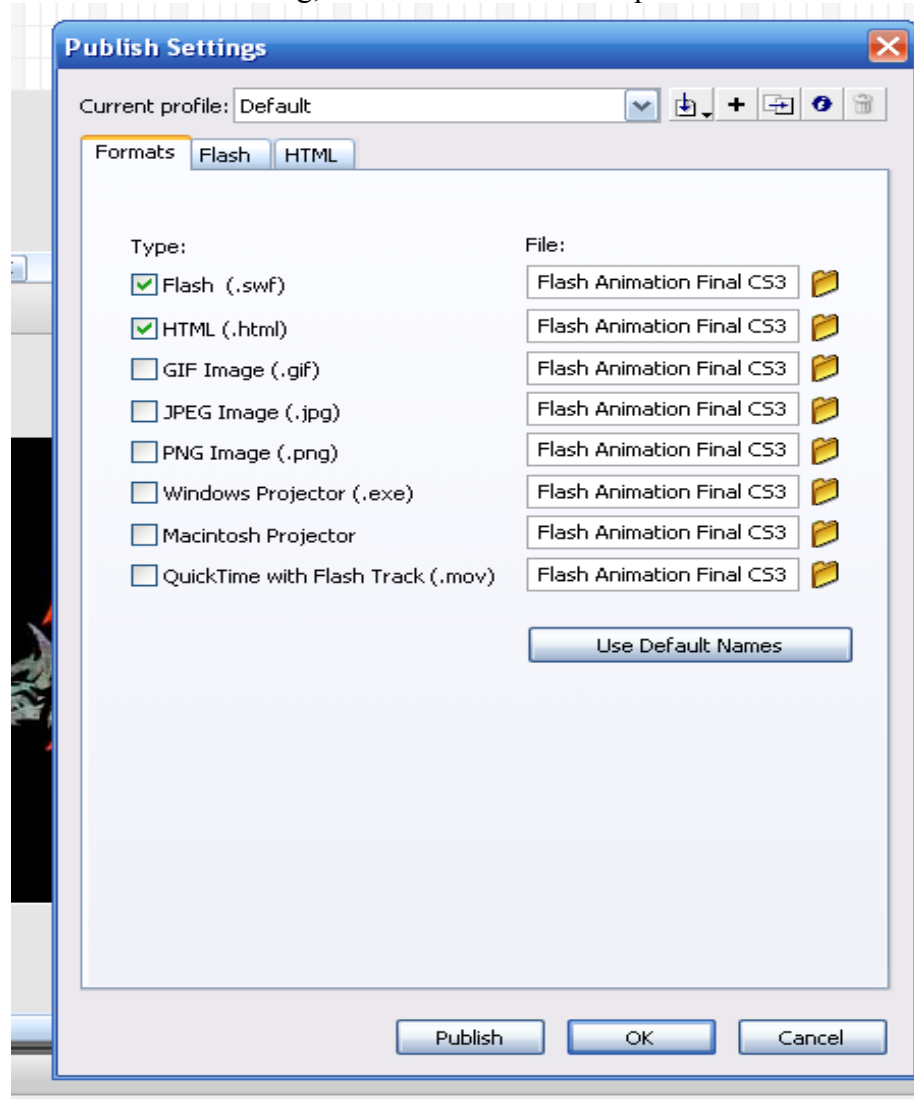
I also added the script "stop();" To the final frame, in order to ensure that the flash animation would not continue to loop around, but stay on the interface for the user to navigate.

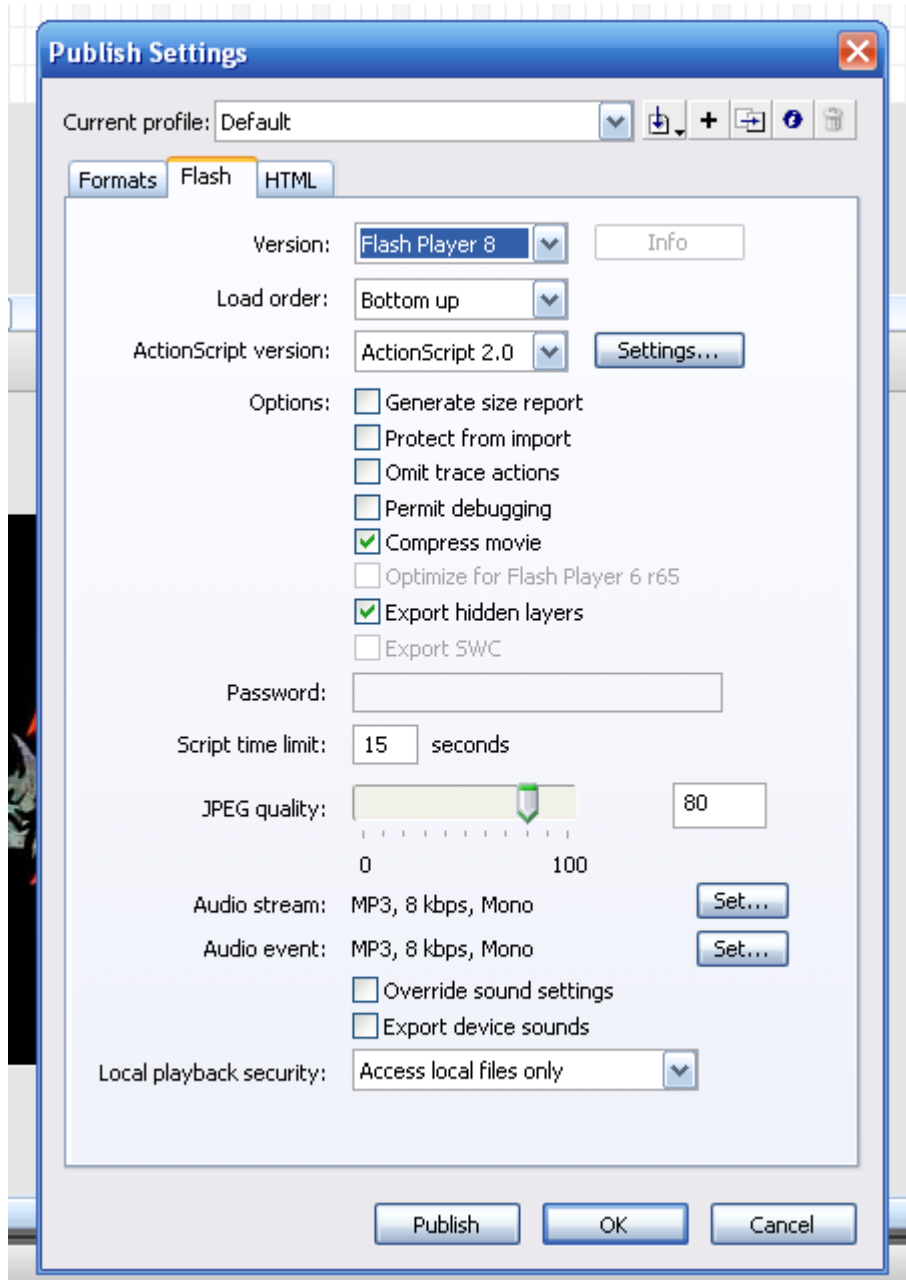
Reflection:

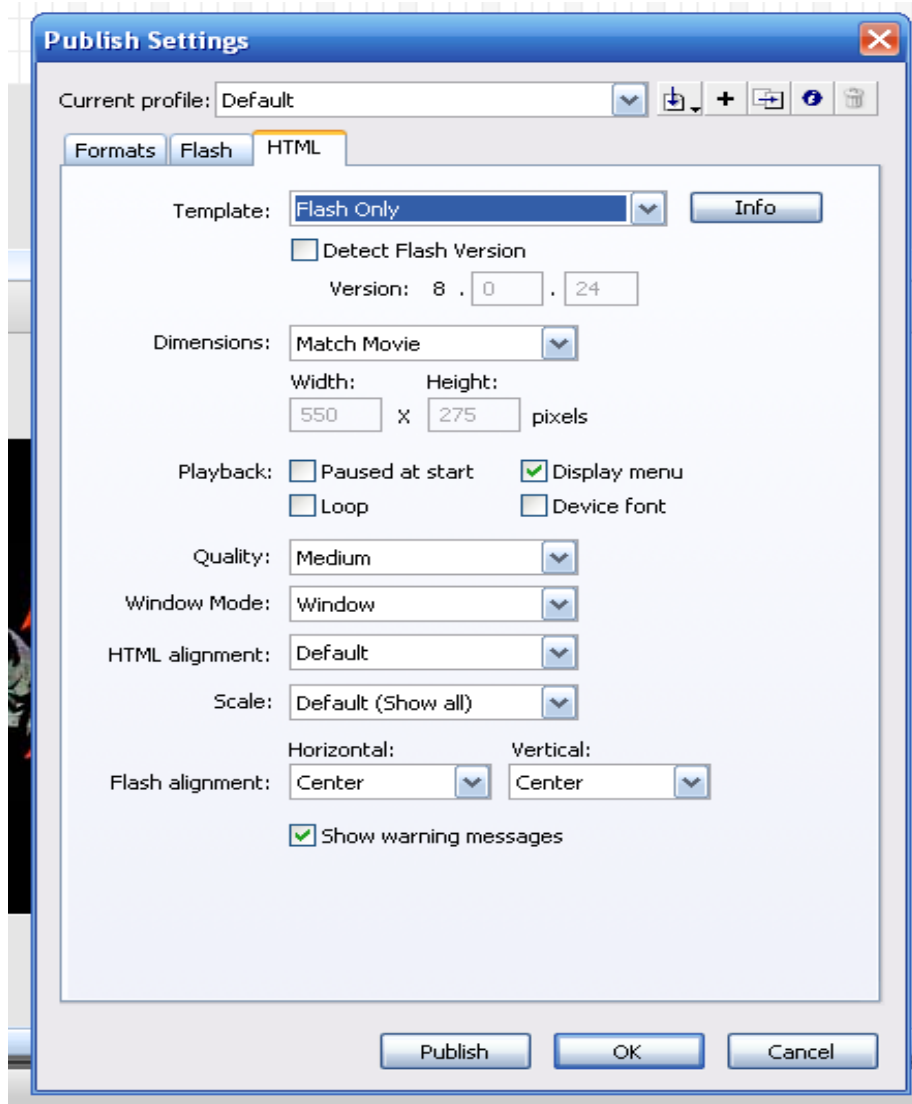
I am pleased with my interface, it has a good quality background, with all the correct logos displayed in appropriate areas, I have also used placement of the buttons (Triforce) reminiscent to the Zelda series, and as they are not part of the logo itself it should be quite intuitive to the user,

Blog 6 – Publishing

For publishing, I selected several options to try and make it compatible with earlier versions of Flash, I set the version to 8, and the actionscript to 2.0, I also stopped the animation from loading, and set the audio to 8kbps to reduce file size.







My final movie can be found at:

<http://jcb33.co.uk/host/Uni/Flash%20Animation%20Final%20CS3.html>

For easy viewing.

Blog 7 – Conclusion And Bibliography

Conclusion

I quite enjoyed doing this assignment, however would have liked to be able to use more of my own assets, sadly due to lacking an artistic flare I had to use many readily available assets such as the Zelda logo.

I did enjoy getting my ideas working, such as the preloader, but would have liked to do certain things better, such as embedding a better quality animation, as apposed to a movie file.

I am relatively pleased with the work I have created, however feel that it is not as good as many animations out there today.

Bibliography

Websites:

<http://www.kirupa.com/developer/mx/percentagepreloader.htm>

<http://www.tizag.com/flashTutorial/flashbutton.php>

Other Sources:

Lecture Notes

Virtual Campus Resources