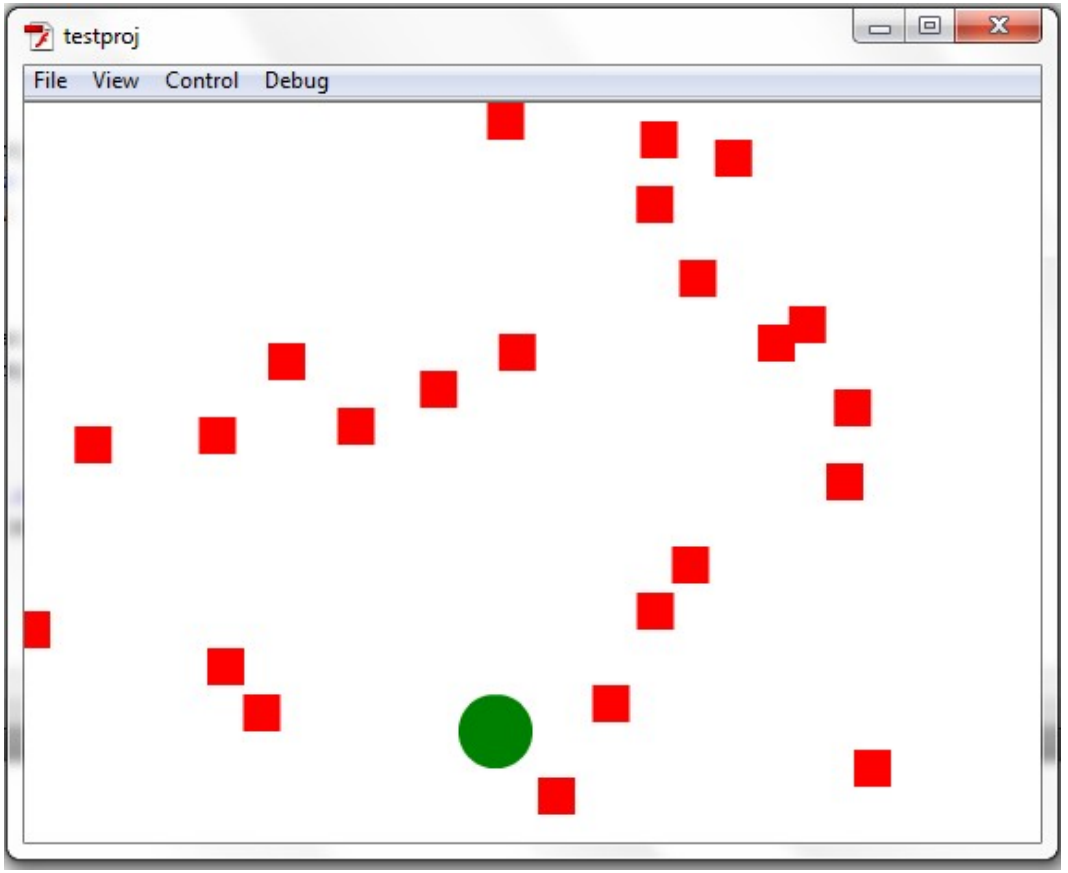
Flash {AS}3

Session 4: Importing Assets

Shiny Yang (mootothemax@gmail.com), special thanks to Alex Miller

ActionScript Recap

Last time...



We made BlocDodger the game.

Draw sprites on screen.
(Add to display tree)

Animate the blocks.
(Set a timer interval)

Have the player move.
(Listen to keyboard events)

End the game when player is hit
(Listen to keyboard events)

What if we want to add in some special effects?

ActionScript Goal

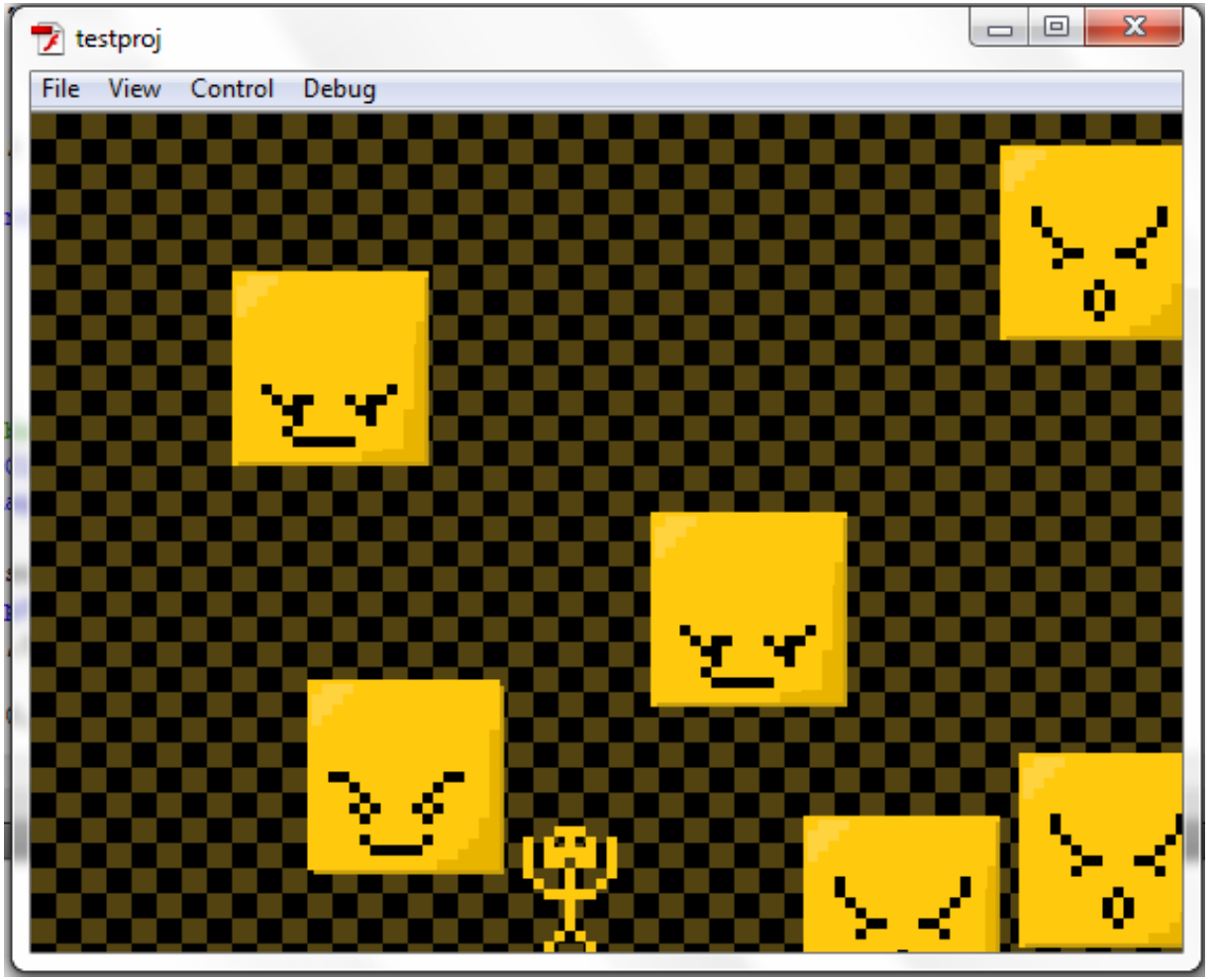
Let's jazz it up a bit!

Add image sprites for enemy block and player.

Add background texture.

Play background music.

Play sound effects when game start and game over.



ActionScript Syntax

Embed Images

```
import flash.display.Bitmap;

[Embed(source='source.png')]
private var imagedata:Class;
private var myImage:Bitmap = new imagedata();
```

Change source.png to the image you want to embed, place outside any method (but still inside package and class)

myImage is now a new instance of the class Bitmap (which is a child of DisplayObject)

ActionScript Notes

[Embed]

The embed tag is one of several “Meta” tags that can be included in ActionScript code that will have a final effect on the output .swf (can you remember another?)

You can only “embed” objects into type String or Class (as3's version of Object)

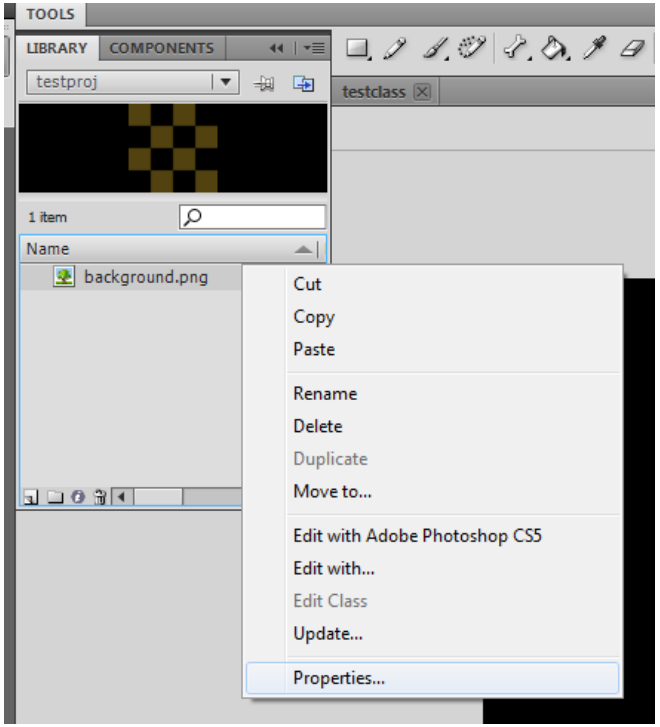
ActionScript has classes that are smart enough to know how to automatically convert this byte data into a useable format (see the Bitmap class type conversion)

You can use embed to have access to many other filetypes as well! (including music, video, text, xml, etc)

Embedded files are automatically included “inside” the swf file when compiled (see file size difference)

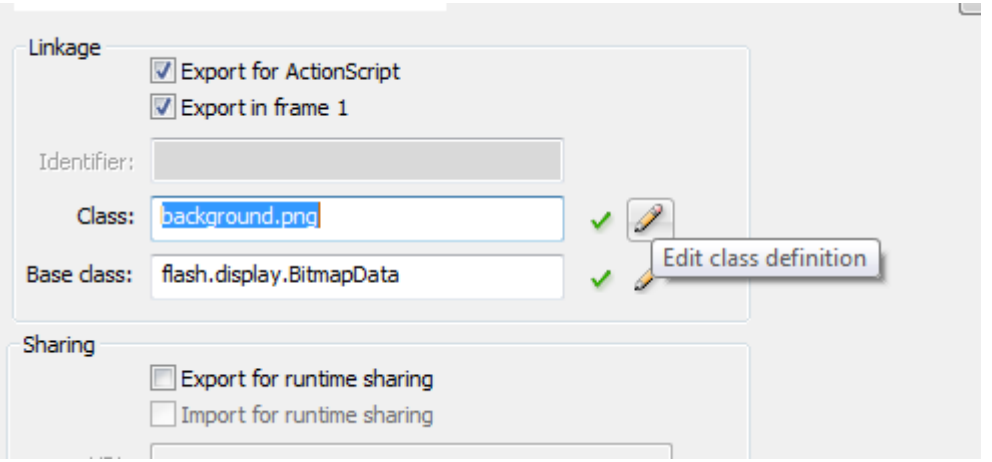
ActionScript HowTo

If you don't like embed(and are using CS)...



Drag and drop into CS to “import” to library, then open library window and go to properties.

Enable “Export for ActionScript”, then give a name to the class.
(You can now access this asset from your actionscript)



ActionScript Syntax

Textured Fill

```
graphics.beginBitmapFill(yourgrafix.bitmapData);
```

Note: you need to pass it the bitmapData field of your Bitmap displayobject.

After that, you can just start drawRect()-ing or drawCirc()-ing like however.

Less important note: BitmapData is the actual pixel data of the Bitmap resource you are using. You'll be working with this if you want to edit some pixels.

ActionScript Syntax

Embed Sounds

```
import flash.media.Sound;

[Embed(source='music.mp3')]
private var sounddata:Class;
private var mySound:Sound = new sounddata();
```

Big surprise! The syntax is exactly the same.

Play a Sound with the .play(int,int) function
(first param is initial start time, second param is loop count)
(just go .play(0,9999) to loop indefinately)

There's no pause button though...

ActionScript Syntax

SoundChannel

```
import flash.media.SoundChannel;

var sc:SoundChannel;

sc = mySound.play(0,9999);
sc.stop();
```

This has a stop() button!

Acts as a wrapped for your sound that is currently playing, keep it somewhere accessable to be able to manage the sounds that you are playing.

(It can also listen for the Event.SOUND_COMPLETE event...)

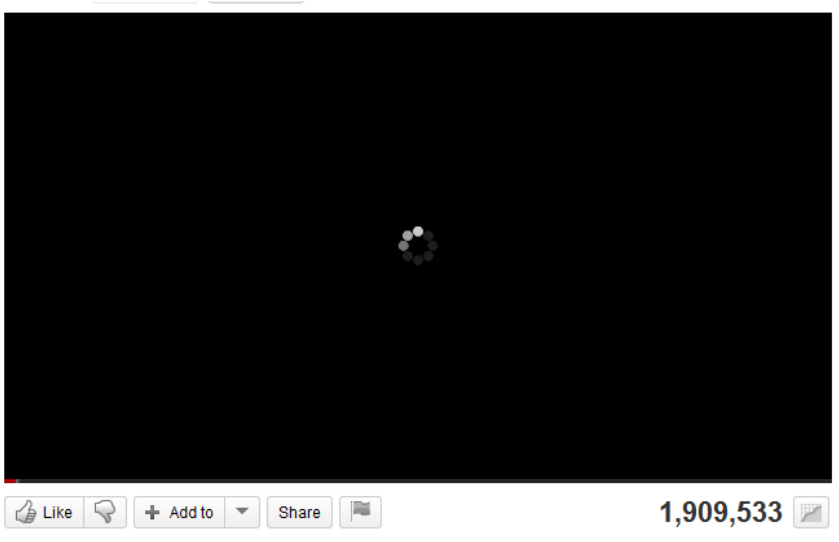
ActionScript Concept

Preloaders

Embedding all this content is great!
But now the filesize of our SWF is growing.

On a slow internet connection your game can take up to several minutes to load (and you can simulate this through the flash player)

The solution? A preloader - to keep the user informed on the progress and speed of loading.



ActionScript Concept

Preloaders

There are two ways to do this - internal and external.
(External is the easier of the two because it encapsulates all the target assets in another .swf file)

To make an external preloader:

Create a seperate flash “project”.

This will compile to an entirely seperate .swf that “loads” the other .swf.

We then add this loaded .swf to our stage.

All of this is accomplished through a URLRequest (very much like an Ajax request) that sends its “response” to a Loader object
(which, being a child of DisplayObject, we can add to the display tree)

How does this look in code?

ActionScript Syntax

Loader

```
var swfLoader:Loader = new Loader();
swfLoader.load(new URLRequest("testproj.swf"));
stage.addChild(swfLoader);
```

And you can go like:

```
swfLoader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS, loadPro);
swfLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, loadCom);
```

You would add them before you called load(), and then do this:

```
function loadPro(evt:ProgressEvent) {
   event.bytesLoaded; //total loaded sofar
   event.bytesTotal; //total
}
```

So overall, not too bad.

ActionScript Note

Loader Notes

One very important thing:

```
swfLoader.load(URLRequest);
```

Runs the constructor code (of your loaded .swf) immediately after the whole .swf package is finished loading. **However,** it is not added as a child to the Loader object until a unspecified amount of time later. (And thus will have null as the stage field)

What does this mean?

```
stage.addEventListener(Event.....
```

Any of your stage calls in your loaded swf constructor will break.

How do we fix it?

ActionScript Note

Loader Notes

In your constructor, replace all stage calls with this:

```
if (stage != null) {
    initKeys(null);
} else {
    addEventListener(Event.ADDED_TO_STAGE, initKeys);
}

...


public function initKeys(evt:Event) {
    stage.addEventListener(KeyboardEvent.KEY_DOWN,playermove);
}
```

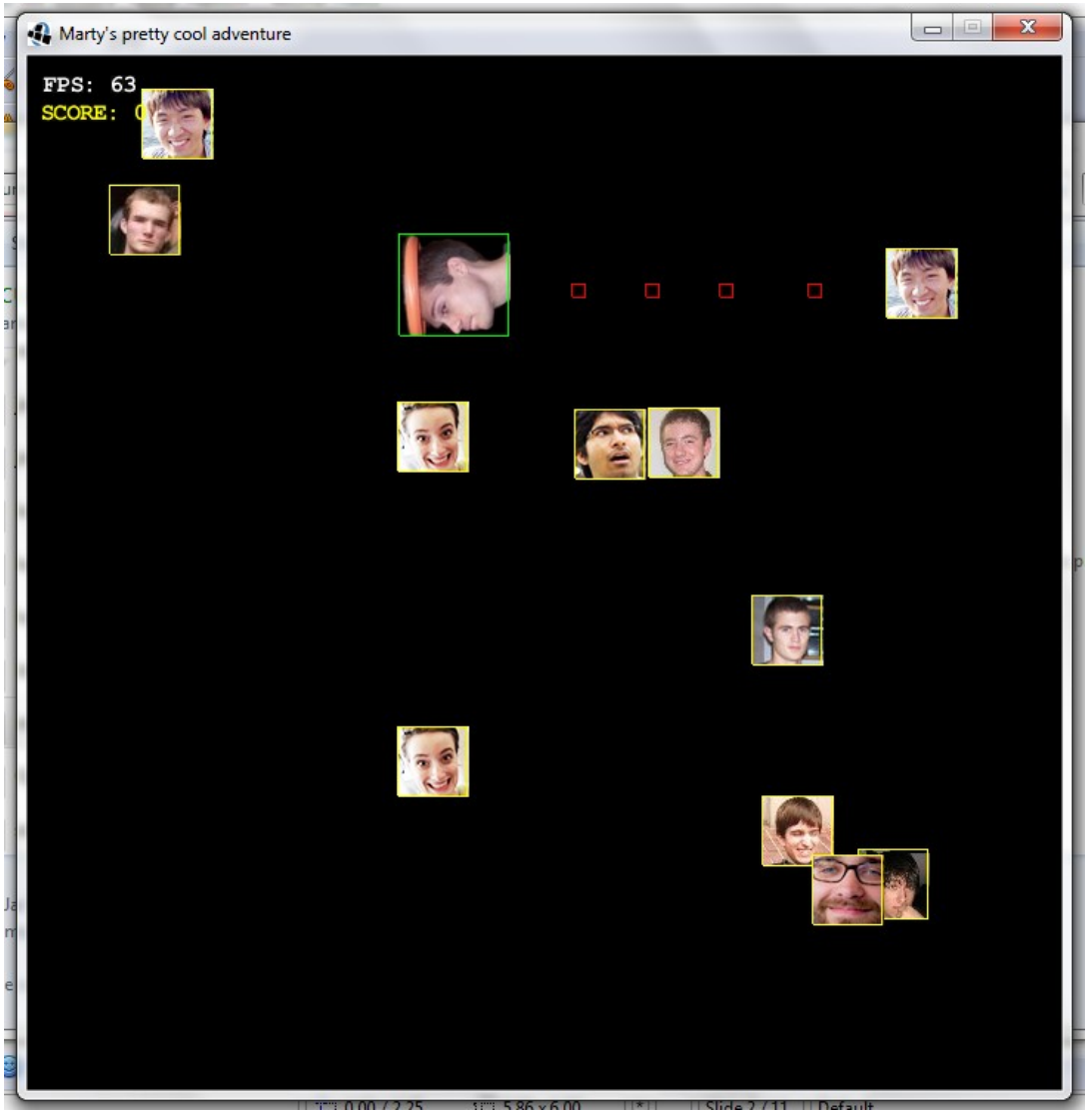
You will (strangely enough) have full access to the stage after your constructor is finished.

Preloaders = No Problem

By the way, what is the URLLoader? Find out next time...

ActionScript Homework

Marty's Pretty Cool Shooter Redux



Add graphics, sound and a preloader to your marty shooter game.

-Both marty, the enemies and the background should have some sort of image.

-Play a sound when the game starts, when an enemy is hit and when the game is over.

-Play background music that is stopped when the game is over.

-Add an external preloader that does more than just display the loaded % in a text field.

On the website are some of my sound/graphics assets that you can use (if you're not creative)