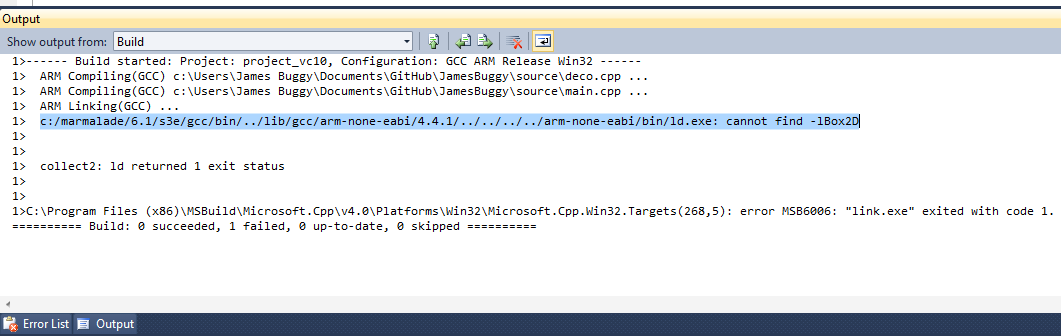
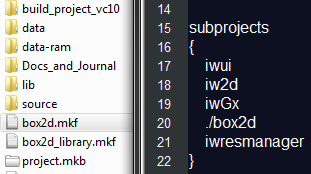
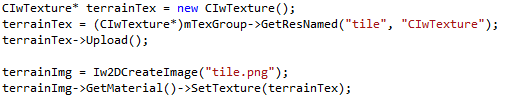
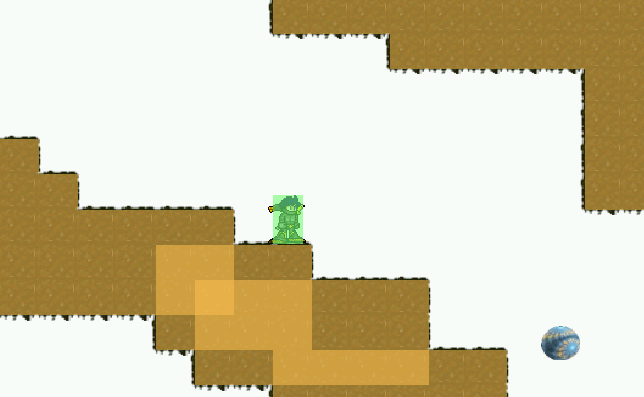
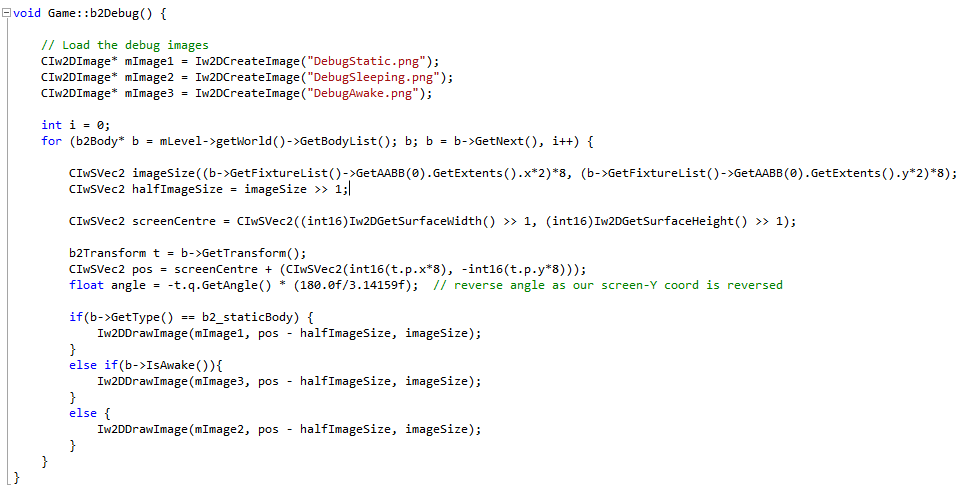
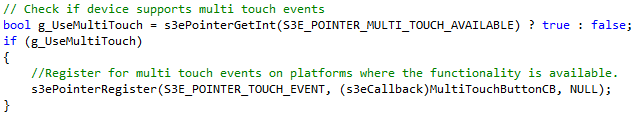
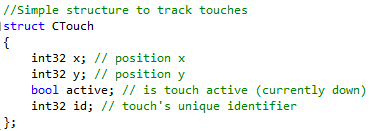
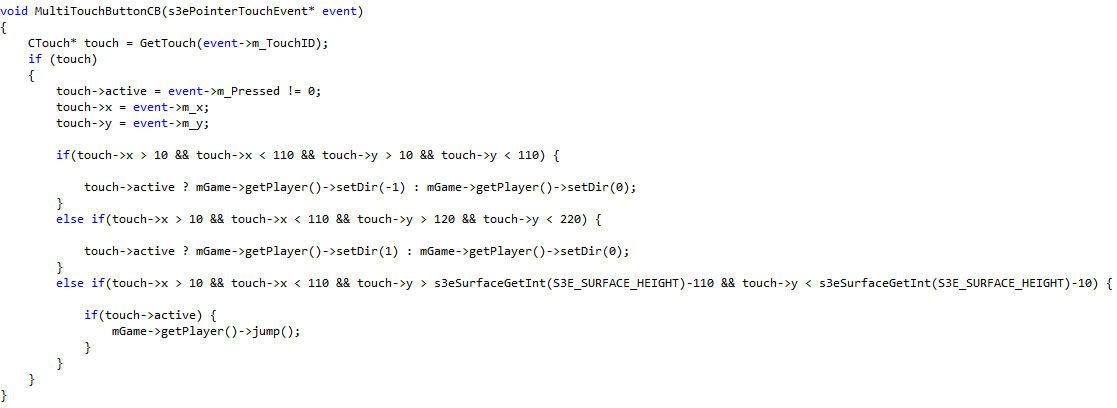
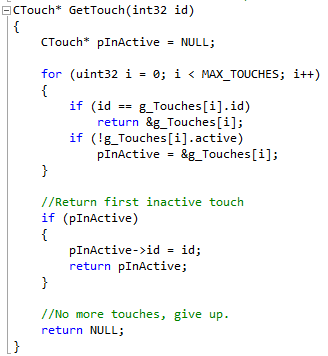
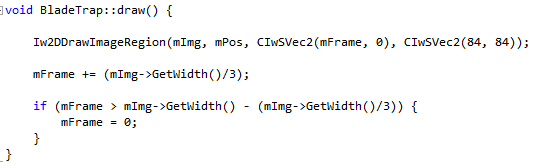
Final Year Project Journal

**LO:** **Incorporate Box2D to Marmalade Project (3)**  
  
**The goal:**   
 To incorporate Box2D into my final year project.  
  
**Problem:**  
 When building the project for an ARM release I received the following error.  
  
**Solution:**  
**[From** [**https://www.madewithmarmalade.com/devnet/forum/box2d-linking-error**](https://www.madewithmarmalade.com/devnet/forum/box2d-linking-error) **]**

This was caused by using a box2d\_library.mfk file which did not include the arm library. This was quickly fixed by including the correct box2d.mkf file as a subproject in my projects .mkb file.

**LO:** **Load images to represent the player and other game objects (2)**  
  
**The Goal:**  
To load images and render them to represent the various game objects.  
  
**The Problem:**  
Marmalade loaded and rendered all images correctly in the Marmalade simulator. However when I finally installed my game on an actual iPad none of the images were rendered. This seems to be an issue with my code as the iPad renders images in all Marmalades example code. I was loading images using:  
https://lh3.googleusercontent.com/if30CqP8LWb4TsPAJ1zTeiTKLVwjNnqh_QyDcAwPUxbl5Dfm6JQxS6zG-LChFmAFxd-RdfbHU0ABQtRBSxJ2Lm1fhhIyRXkly0JXp61W9WLATknXPTZ6  
  
**The Solution:**  
I have yet to discover a real solution to this problem however I have discovered a workaround using resource groups. A resource group is basically a text file that points to all resources to be used in an app.  
  
  
This group is then loaded using:https://lh5.googleusercontent.com/Jz8-gdt3emsyKNzL6AxwD8YAI47SZOBpgCzkr3qP4qj_9Ou6genoB8L2JMRCiOkGFSd-0NZEKeJDqPjOGDlV0nMcUHD74-yY_oVabbkjgV5al8BEb8Gl  
  
Using the group I then set the texture of my images like so:  
  
Using this method Images are render on the device.   
  
  
  
**LO: Add box2D physics objects to resolve ground and wall collisions. (3)**  
  
**The Goal:**  
To add box2D physics objects to my level to resolve player collisions with the ground and walls.  
  
**The Problem:**

As box2d is agnostic about rendering I had to create my own simple debugging function to render all box2d objects. Otherwise I could not position them correctly as I could not see them.

**The Solution:**  
To fix this I wrote a simple function to loop through all box2d objects in the world and draw a semi transparent colour over them.  
All static object are rendered with an orange colour, dynamic objects(the player) are rendered green while awake and grey while sleeping.  
The function:  
  
  
  
**LO:**  **Add multi touch input to the game  (3)**  
  
**The Goal:**  
To add multi touch input to the game as the player can both perform multiple actions at once.  
  
  
**The Problem:**  
The initial method I was using to handle input from the user was more suited for use with a menu system as it only handled a single touch event at any time. As the player can both move and jump/attack, I needed multi touch compatibility in my game.  
  
**The Solution:**  
 **[From Marmalade s3e multi touch example code]**  
  
I first check if the device supports multi touch events and register the “MultiTouchButtonCB” function as the handler for this event.  
  
I created a simple struct to track the information of each touch.  
  
I then created an array to contain these touches.https://lh5.googleusercontent.com/Lwnj-blQJzmmiNMTlZpORgAWQbcvLOrhgC7sBIsOL3019gIwGJzNIdafAomY4hRzizLZBgdoVLGcsr-3zU17d9OSj72xpiwH4cO5AGHZcpotCuiZc8Mf  
  
The MultiTouchButtonCB function checks if one of three buttons are touched and performs to appropriate action.  
The get touch function will return the information of the touch with the id passed to the function. If there is currently no touch with this id it will allocate a free one from the list. Up to a maximum of ten (though I will lower this as only 5-6 touches will be needed).  
  
  
I have left the first input method in my code as I intend on using it for the menu system of my game.  
  
  
  
**LO: Animate game objects (2)**  
  
**The Goal:**  
To animate the player and other game objects.  
  
**The Problem:**  
 There is another inconsistency between the emulator and an actual device when it comes to animating the sprites in a game.  
In the emulator, animating sprites using the following work perfectly.  
  
However on the actual device, the image is drawn to the correct scale, but is extremely distorted. The image is rendered dozens of times at a small scale within the area that the correctly rendered image should occupy.  
  
**The Solution:**  
  
 I have not yet found a solution to this problem but am certain that it is not a very large issue.