Graphics Assignment 1

Ben Townshend – 13480634 – CGP2012M

# Features

* Window
  + Window title contains student name, number and module title.
  + Window is moveable and resizable.
  + Window can be made full screen.
  + Aspect ratio is maintained at 4:3, and black bars are rendered.
  + Window starts non-full screen and half the desktop window size.
* Geometry
  + Visible player, barricades, enemies and bullets.
  + World boundaries represented with geometry.
  + Enemies have a “spin off” animation when they are killed.
* Gameplay
  + Both player and enemies move, are stopped by world geometry and can shoot.
  + Player has multiple lives represented visually.
  + Game ends, and restarts when the player’s life counter reaches 0, or the enemies reach the bottom of the screen.
  + Enemies speed increases over time.
* Textures
  + All geometry is textured.
  + Barricades degrade over time, and show different textures based of its state.
  + Enemies animate over time.
  + Double layer background consisting of two textures with transparency.
* Extras
  + The camera tracks the player.
  + Player score indicated in the top right corner.
  + HUD elements are independent from camera movement.
  + Background elements scroll and parallax.
  + Endgame scene when player kills all the enemies.

# Reflection

The design process overall went well. Due to previous work on Games Programming last semester, getting an initial build system running was very simple as the basic solution is very similar. This time however, more modules had to be included and initialised.

Once OpenGL was setup within the project, I decided to separate out the rendering code immediately, and have it running within a class that represented its own sprite, this allowed rendering code to be written once and work across all sprites. It also made creating multiple objects very easy.

I created a class that represents the game state. This class stored lists of all the elements in the scene, as well as all the texture information. This meant that the update and render function in the main program could access all this data and make any modifications necessary. Storing the texture information outside of the sprite objects, and passing it as a reference during the render sequence made changing the textures on each sprite very easy.

I stored the code in a git repository. This meant developing across multiple machines much easier, as well as ensuring that if any changes broke the project, I could easily reset it to a previous state.

# Video URL

<https://www.youtube.com/watch?v=bTHc8x8VVq4>

# Evidence

Window title displays all crucial information:

https://i.gyazo.com/050e84950c5be0351eb583564c025ca9.png

Window renders black bars. Player, barricades, enemies and bullets are all visible. World boundaries are represented with geometry. All geometry is textures, barricades degrade visually. Parallax background with asteroids above coloured background.



Player lives represented in the bottom left, score represented in the top left:



Spin off animation: Sprites:

C:\Users\Ben\AppData\Local\Microsoft\Windows\INetCache\Content.Word\bullet.pngC:\Users\Ben\AppData\Local\Microsoft\Windows\INetCache\Content.Word\barricade-broken.pngC:\Users\Ben\AppData\Local\Microsoft\Windows\INetCache\Content.Word\barricade.png