**Asteroids**

Requirements/Rules:  
- Player controls rotation and forward direction, player starts with 3 lives   
- All objects wrap around the edge of the screen  
- A few large asteroids spawn at the beginning of each level, more asteroids each time the screen is clear  
- Larger asteroids break into smaller asteroids that move more quickly   
- Player lives decrease to 0 game over  
- Large asteroids less points, small asteroids more points

**Classes:**  
Game Object:: Collision Object  
Position, parenting information and functions, Draw functions

Sprite Object:: Game Object  
Position, loads an image and stores as texture data

Collision Object  
Sphere, Plane, collision maths, sphere centers and radius checks

Player:: Game Object  
Lives, Speed, Rotation, Fire Bullet, collision with other objects.  
The player can only move in the forward direction and rotate accordingly, and accelerates and decelerates, checks for collision with asteroids every frame, if it does lose a life, get some invincibility time so player does not lose straight away if player collides with the bounds of the screen check which bound it collided with and teleport them accordingly

Bullet:: Game Object  
Collision detection against asteroids and remove them from the game, and from the list of total asteroids. For each object in the scene the bullet will check if an asteroid is hit with sphere collisions, if it is then the asteroid is split if it has splits left, if not it will just destroy the object. If a bullet hits the edges of the screen it gets destroyed.

Asteroid:: Game Object  
Destroy large asteroid, replace with two smaller asteroids with similar trajectory upon destruction add to game score. Asteroids will be stored inside of a vector upon and then removed from that vector when destroyed. if Asteroid collides with the bounds of the screen check which bound it collided with and teleport them accordingly

All objects that are made must be deleted when created to avoid memory leakage

**Initialise:**

Creates planes to create bounds of the screen for objects to interact with, loads a texture to use for the background, creates an instance of the player to control and check for lives.

**Update:**

When the game is updated it first checks whether the vector storing the asteroids is empty, if it is then it spawns more, the number of asteroids spawned at a time is incremental, acting like levels of difficulty. Delta Time is calculated ad update is called on each individual object inside of the scene, each object checks its collisions and returns the result of those collisions. If a player’s lives is 0 then the players then the player will no longer be able to be controlled.

Repository Link (Proof of Version Control)  
<https://github.com/DanielBainbridge/AIEProg_Asses4>