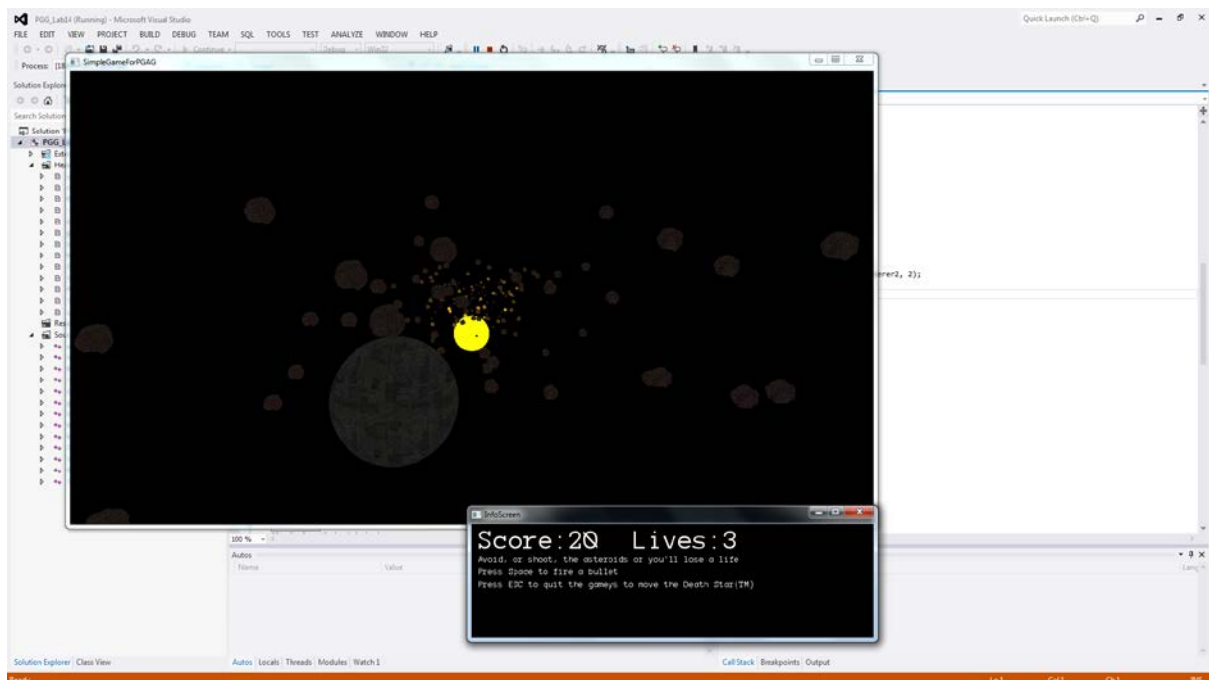


## Assignment 2 Report

### Introduction

For this assignment I was tasked with demonstrating my understanding of 3D graphics and programming principles by doing one of two options given to me in the brief. The first option is to develop a simple 3D game of my choice that makes use of 3D graphics, has gameplay, and appropriate in-game information. Some of the examples given to me for the first option were scrolling shooters, 3D platform games, a racing game or an FPS (First-person shooter). The second option is to create a 3D technical demo by creating a small prototype program that shows off a specific advanced aspect in a game-like environment, for example advanced lighting or rendering, or animation blending or skinning, or a physics demonstrator with my own physics library, such as pool table simulator or fluid simulation.



(A screenshot of game in action)

For my assignment I decided to go with the first option of creating a 3D simple game, as I felt this would be more enjoyable to create, and would allow me greater freedom when it comes to attempting to implement 3D techniques into my program. For my 3D simple game I created a simple scrolling shooter where the user controls a space ship (in this case a Death Star looking ship) and can move vertically and horizontally while asteroids fly forward at random speeds and in random positions, giving the illusion of the space ship constantly moving forward. The aim of the game being to get as high a score as possible, by shooting an asteroid to gain 10 points, before you completely die when the user runs out of 3 initial lives that they are given.

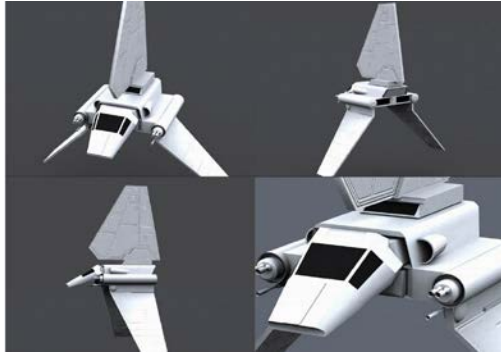
## Research

Before actually creating my game, I first undertook research into a few things; firstly researching a few OpenGL tutorials, secondly I researched some similar games to my idea in order to gain inspiration for features I should include and things I should avoid, next I research game models and textures that I could use in my game, and lastly I undertook research, as I went along, into some techniques that I would need for my game (3D collision, lighting, shaders, etc.).

To start I looked into some OpenGL tutorials on [opengl-tutorial.org](http://opengl-tutorial.org) in order to get re-equated with OpenGL after not using it regularly for over a year. I then looked into similar games to my idea and got inspiration from many never ending scrolling shooters and found that the simplicity was a big reason I wanted to keep playing. I therefore decided to keep mine simple for the user, by just having them move vertically and horizontally and pressing space to fire, while the scene moves towards them to make it seem like they are moving forward. The classic game Asteroids was also played while researched and gave me the fairly simple idea of having a high degree of randomness, for the size, speed, and position of the asteroid as this helps to keep the user moderately alert and keeps the game from falling into a pattern and getting boring.



For game I wanted the models and textures to look good but while also keeping the game simple and quick to load. I therefore went for mostly minimal models, which was very easy as I only needed a spaceship of sorts, asteroids and bullets, and I also added in a sun in the background which acts as the light source for the game. The model mesh for the sun was just a sphere, and the texture for it was simple sun looking texture that I was able to find online. As the bullet is only small and is never seen properly by the user, I used a small cube that was easily created in Maya, for the bullet, and added a bright gold coloured texture to distinguish it from the rest of the game, which has a fairly dark, slightly bland looking colour palette. When researching possible meshes for the asteroid I considered just using a deformed sphere and adding a rock texture, but managed to find a good mesh for free on 'turbosquid', and then added a rock looking texture.



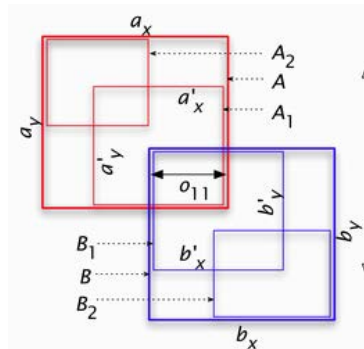
(Lambda Shuttle)



(Imperial Star Destroyer)

The bulk of the research into the game models came when researching a good model mesh for the spaceship, as I wanted a spaceship that would be easy for aabb box collision, that looked cool, and also for it to be a good size, memory and polygon wise, in order to keep load time down, and lastly for the model mesh to look cool and the spaceship is the most important mesh as it is the mesh the user will see most and it is the model that allows for greatest creative freedom, due to the many different shapes and types of spaceships. I firstly looked on Turbosquid for some Star Wars type ships and found a Lambda Shuttle from Star Wars for free, but thought this would be too difficult to do accurate collision with. I next looked at the Imperial Star Destroyer as this was a simpler shape that had not too many concave type parts, which would make collision moderately easier. However, when loading the model into Maya it wouldn't group properly. Keeping with the Star Wars theme of ships I went for a Death Star looking ship, which was basically just a sphere with a space ship texture on it.

(Death Star)



(AABB Collision diagram)

In my game the only mathematical problem I encountered and needed research into was the collision detection between two objects in my game. For this I looked into AABB collision for 3D objects, which required me getting the absolute min and max of each mesh and then later using this every frame, along with the position of the object, to test whether an object has or is about to go inside the object by testing whether the object's maximum point is higher than the other object's minimum position and if object's minimum position is less than the max position of the other position. I called a function that did this every frame for when collision testing was needed, which was for the array of asteroids with the array of bullets to testing bullet hitting, and the player with the array of asteroids to check for asteroid hitting so the player can die.

### Program Description

A high-level description of your program showing how it works, with program logic flow diagrams and system-level block diagrams as appropriate

### Project History

For my Project I have Git repository with many commits acting as a sort of diary of creating and fixing my program as I went along. To find this you will need to go to just go into the PGAGAss2 folder and right-click, go down to Tortoise Git and then go to show log to view my commits.

### User Testing

Some basic user testing of your program, with feedback comments on usability and design

### Analysis

An analysis of your program, identifying its strengths and weaknesses

### Conclusion

Conclusions with proposals for future improvements

## References

<http://www.turbosquid.com/FullPreview/Index.cfm/ID/451171>

[http://fc02.deviantart.net/fs71/i/2012/162/e/1/rock\\_texture\\_by\\_roskvape-d531x8m.jpg](http://fc02.deviantart.net/fs71/i/2012/162/e/1/rock_texture_by_roskvape-d531x8m.jpg)

<http://svenniemannie.deviantart.com/art/Spaceship-Texture-91959740>

[http://all-free-download.com/free-photos/download/gold\\_textured\\_background\\_hd\\_picture\\_4\\_169653.html](http://all-free-download.com/free-photos/download/gold_textured_background_hd_picture_4_169653.html)

[http://static.brusheezy.com/system/resources/previews/000/049/324/original/Surface\\_of\\_the\\_sun\\_prv.jpg](http://static.brusheezy.com/system/resources/previews/000/049/324/original/Surface_of_the_sun_prv.jpg)

<http://www.opengl-tutorial.org/>

<http://www.turbosquid.com/3d-models/lambda-shuttle-max-free/290438>

<http://www.turbosquid.com/3d-models/3d-star-destroyer-imperial-model/423013>

[http://cgvr.cs.uni-bremen.de/papers/egve06/constellation\\_og.png](http://cgvr.cs.uni-bremen.de/papers/egve06/constellation_og.png)