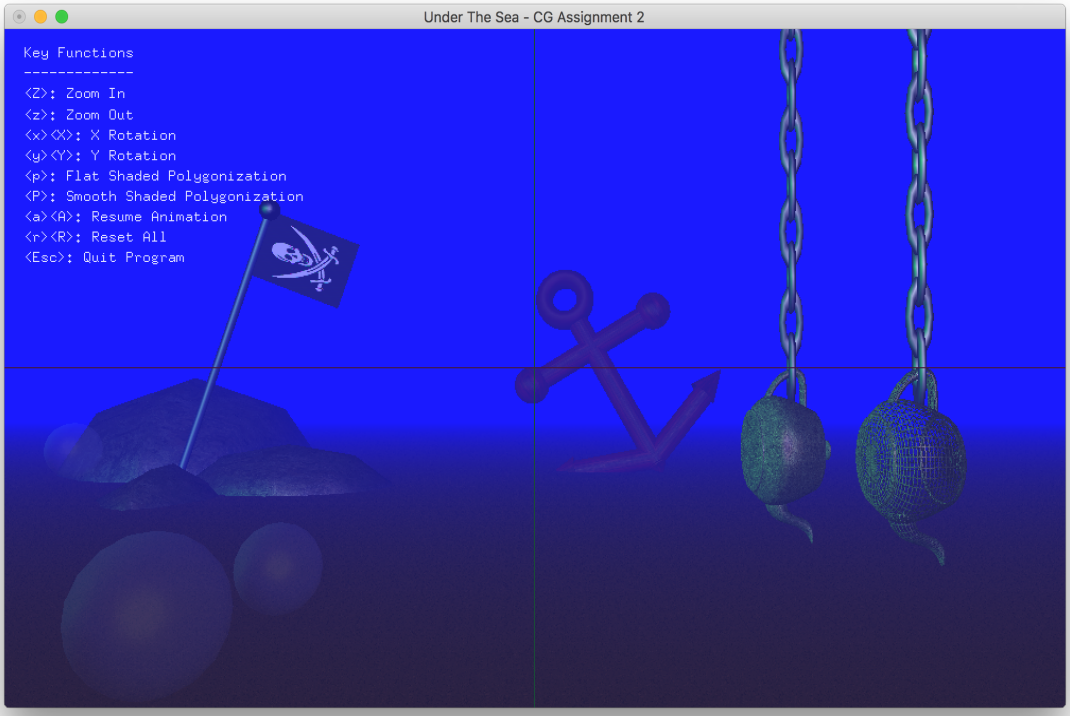
CG200 Report
OpenGL Assignment

Connor Beardsmore - 15504319



Curtin University
Science and Engineering
Perth, Australia
October 2016

OpenGL "Under The Sea" Assignment

Features Implemented

All features specified by the marking guide have been implemented and the scene consists of the following features. Key press functions have been utilized to allow the user to control the scene with the keyboard. The keys and instructions to run are clearly illustrated in the scene. A basic animation has been developed, with full details of this described later in the report. Each shape becomes clearly the closer you zoom, with level-of-details clarity performed. To increase realism, several objects have a combination of both texture maps and surface finishes. Two coloured light sources are located in the front and rear of the scene, with the specular surface finishes showing realistic highlights on the object surfaces. The use of additional effects has also been applied, with the bubbles illustrating transparency and the entire scene being enveloped in light blue fog.

Main Algorithms

- how was chain created
- how was anchor created
- how does fog work
- level of details
- all ojects take in x,y,z for easy movement
- push and pops

Objects Modelled and Surface Finishes

The scene consists of multiple shapes, both simple and composite. The floor was modelled as a simple square plane, with a basic "dirt" bitmap for it's texture mapping. Three rock objects were created via calls to gluSphere(). These rocks also utilized a texture map to provide additional realism. Further spheres were employed for use in the three transparent bubbles on the left of the scene. These spheres vary in size with transparency to correlate with real bubbles. While appearing complex, the teapots are actually created through the GLUT library, with calls to glutSolidTeapot(). The teapot is the only object in the GLUT library that allows for texture mapping and thus, a "brass" texture map was applied to the teapots. The use of material surface finishes creates the appearance that the teapot is old, mouldy and rusted.

The composite objects are the teapot chain, the pirate flagpole and the wooden anchor. These three composite objects use combinations of various simple objects such as spheres, torus, cylinders and cones to a varying degree of complexity. The teapot chain is simply a torus, scale and stretched. The torus is repeatedly created in a for loop, with each alternative object being rotated at a 90 degree angle to the previous. The flagpole consists of a sphere top, cylinder pole and flat plane for the flag.
// TALK ABOUT ANCHOR HERE

External Tools

The imageloader.cpp file was gathered externally from Jacobs 2016. This resource is a bitmap loader, allowing for the use of texture mapping on objects within the scene. The resource was easily applied from my code and was a simple method to call to read in the .bmp image files. All composite objects were modelled within the code with no external assistance. More complex composite objects could have been converted from .obj files, but it was a stronger learning experience to create these objects myself through trial and error techniques.

Animation

The scene includes a basic animation illustrating the capabilities of OpenGL. The pirate flag sinks into the ground slowly over the course of the animation, while the transparent bubbles on the left of the scene float into the background. These bubbles are simply translated simultaneously along all three coordinate axis. The two GLUT teapots are pulled up from the bottom, with the chain rotating round as they move. This simulates the spinning of the chain about the Y axis. To add a further layer of realism, the fog moves forwards to fully cover the scene as time progresses.

The user can utilize a variety of key presses to control the animation as they wish. The A key starts the animation, while the T and C keys pause and resume the movement respectively. S and F work to slow down or speed up the scene. The key presses work my modifying a global *speed* variables. This variable controls the translations and rotations of the objects, with a larger value resulting in a faster animation.

References

College, Ostfold University, ed. 2009. "Computer Graphics: Some Materials". http://www.it.hiof.no/~borres/j3d/explain/light/p-materials.

Jacobs, Bill. 2016. "OpenGL Tutorial". http://www.videotutorialsrock.com.

Li, Ling. August 2016. *Introduction to OpenGL*. Lecture Notes in Computer Graphics 200. Perth, Australia: Curtin University.

Lighthouse3d.com. 2015. "GLUT Tutorial". http://www.lighthouse3d.com/tutorials/glut-tutorial/.

"OpenGL API Documentation Overview". 2016. Khronos Group. https://www.opengl.org/documentation.

Productions, NeHe. 2012. GameDev.net. http://nehe.gamedev.net/.