Comp Graphic and Visualization Project Reflection

Shawn Whittaker

Southern New Hampshire University

OpenGL is the computer industry’s standard application program interface for developing 2D and 3D graphics. Using a set of commands and functions, OpenGL creates a hardware cognizant system to write graphics that can be used across operating system platforms. Each command directs the graphics card in a computer to draw a specific effect. This makes creating graphics far more efficient as you can use these commands for repetitive effects and providing each operating system use in that system’s windowing setting. By using OpenGL, a developer can use the same code to render the same graphics for Mac, PC, and mobile devices.

I set out to render a 3D scene is of my dog’s food and water bowl with the holder. The holder is made of three rectangular cubes, which comprises the two sides and the top. The underside of the wooden holder resides a rectangular dowel that connects the two sides. The two bowls should have been two cylinders placed inside the top of the holder. The holder sits on a mat, which would be a plane.

As I recreated the scene, I struggled to implement more than one shape at a time. The tutorials we used in class would set up the side assignments one way, but I found it quite challenging to use those set ups in a way that would work with this project. In the end, my current build was set up as one large shape, utilizing only triangles, although used in a way to form what would be different shapes.

I used a series of triangles to build my first rectangular cube which made the top of the holder. I was able to use this shape for my lights, so instead of perfect cubes, my two light sources were also this long rectangular cube. As I added on the sides, I adjusted the number of triangles my URender would draw but left the triangles the same number for my lights, meaning I could add onto my food bowl holder while the lights stayed rectangles.

I used two triangles as the plane, and my secondary texture to be the only texture for the plane, which enables us to have a seamless plane while not matching the same wooden texture the rest of the holder has.

The left food bowl was meant to be a cylinder, but since I could not get more than one shape to render in my current set up, I made the cylinder a hexagonal cube. I had the intentions of adding more and more sides, but it became very taxing, so a hexagon it became. The right food bowl because a pyramid instead of a sphere, but this was a much simpler reason. I needed a fourth shape that wasn’t a plane, cube, or what would have been a cylinder, so pyramid it became.

The camera in OpenGL moves by one of two ways, we can either set it to move around the object automatically as an animation graphic, or we can control it through user input. To navigate my scene, the user can use their keyboard and mouse as their user input. Just as most computer video games, we use the W key to move forward, S to move backward, A to move left, and D to move right. The mouse changes the camera view. I also have a spotlight set so the camera and spotlight both follow the movement of the user’s mouse.

About halfway through this course, I realized I could use my project as a template for multiple assignments I was undertaking. I was able to continue to grow my project each week and I would import my project template to do that week’s assignment, and then update my template with the working prototype of that assignment. For example, week 2 was creating a 3D pyramid. I used that template to create my 3D rectangular cube. We then added texture to that pyramid, which I then used to texture my rectangular cube. Then we added light to the textured pyramid, which then went to light my project which was not no longer just a rectangular cube. Each week the code continues to modularize, keeping each part able to be added one with another – the texture, the lighting, the 3D shapes, the plane, all coming together for one scene.

Sources:

Contributor, T. (2011, March 24). OpenGL (Open Graphics Library). WhatIs.Com. https://whatis.techtarget.com/definition/OpenGL-Open-Graphics-Library