Final Project

Eugene Panchenko

CS-330-T2655 Comp Graphic and Visualization 21EW2

Southern New Hampshire University

12/12/21

This class started off with a simple assignment, take a picture of four items and theorize how you would replicate them in OpenGL. I must admit I was bit apprehensive at first. Having done virtually zero with C++ in almost a year, I couldn’t even imagine designing 3D elements. I set my goals realistically and in line with where I deemed my abilities to be. The elements I chose were a cylindrical apparel holder, a soap box, a comb, and the attachment to my Thera gun. Looking back now I wish I chose items that had more dimension and color as the final scene looks very stale and monotone. However, being my first step into 3D graphics design I believe it was an adequate learning experience.

To break my first scene down, I received source files from our professor that had classes which you could call to make a sphere and plane. Since the milestone assignment was to make a complex object, I chose to start with the Thera gun attachment. Having used those classes I was able to construct a sphere then added a cylindrical part to it. To say that was difficult would be an understatement. I spent more time researching. working on and trying to understand the material than I did any other class in the past two years studying at SNHU. The next part was the plane which stood as the base for the rest of my scene. Having used the shape generator class, I was able to work through this part quite quickly as it was a just a matter of moving and sizing the element so it would fit my vision of the scene. Following that there was the nightmare of adding textures and shaders. Not having the time to dive deep into the advanced OpenGL I wasn’t able to figure out fragment shaders in time for the final submission deadline. My scene isn’t as complete as I would like it to be considering once the lighting shader was applied it broke my cylinder textures into a gradient. However, all the assignment bullet points have been met so I hope this doesn’t hurt my final grade.

The last two components were the soap box and the comb. The soap box was quite easy once I figured out how to work with the code. I set up the vertices to form the point coordinates, normal coordinates, and the texture coordinates. This was then compiled, added to a vertex array, and bound to a buffer to be used later. Once it was added to the scene it was just a matter of adding the texture, applying the shader and the transformations to get it into the right position. The pyramid shapes of the comb worked much the same way although I had to add a “for loop” to replicate the final pieces twenty-two times to match the teeth of the comb. After that I added a final cube for the base, then came the lights. I chose to do a directional light which would illuminate my scene and a spotlight which would be attached to the camera to give the user the ability to apply their own “light” to each object.

One of our assignments revolved around navigating our scenes. Since the scene is 3D, you have the power to move the camera and view your elements from each side. We had to add a 2D Orthographic projection view as well, but I didn’t really find it very useful aside from building the initial cube at the beginning of the class. A user moves through my scene by pressing “W” to move forward, “S” to move back, “A”/”D” to move left and right respectively, and “Q”/”E” to move up and down respectively. To switch between the view options “P” could be pressed although it was a bit finicky as it felt OpenGL would move between the two options several dozen times before I was able to select the view I wanted. Having done that I believed the requirements of the assignment had been met, but I noticed there was a bullet point that said at least one of the lights of the scene had to be colored so I figured I would have some fun with this. I didn’t want to dedicate a specific color for the directional light, so I left that up to the user. By pressing “1” the lights will turn-on to their default setting, “2” the lights will turn off and the user will have to navigate the scene using only the spotlight that’s attached to the camera. I also added a modifiable element for color. Since there is a gradient for each major color (R,G,B) code was added to allow the user to increment each color independently. “R”/”T” raises and lowers the red gradient, “G”/”H” raises and lowers the green gradient, and finally “B”/”N” raises and lowers the blue gradient. All these things combined with the mouse movement and the scroll wheel which allows the user to choose how fast they move through the scene gives a lot of power over a seemingly simple 3D scene.

In a perfect world I would have made my code much more modular and broken out most of the functions into their own sperate classes/source files but in the interest of time I chose to make a very well documented code with some modularity. Everything is organizing just like reading a book, most of the functions either have an explanation or relation clearly titled to allow the user to modify whichever element they chose. Modularity would be the next thing on the list, It shouldn’t be too hard of task with how things are documented. Knowing this I would say my code is reusable, that is if you know what you’re looking for and how the elements and transformations work!

References

*Welcome to OpenGL*. Learn OpenGL, extensive tutorial resource for learning Modern OpenGL. (n.d.). Retrieved December 13, 2021, from https://learnopengl.com/.