Southern New Hampshire University

CS-330 T5454 Comp Graphic

Final Reflection

Ethan Cook

In my endeavors throughout this term, I have created multiple images. My final project was far from perfect, but to be honest creating a functional program was a step in the right direction in developing my programming skills. I chose very simple objects, not trying to make it easy, I just have little kids so I thought using the blocks they play with would be a good way to hit all the requirements of the scene. I chose depicting the red rectangular prism as that was a very straight forward item that incorporates multiple components to generate a quality output. Writing the program was very challenging, but very rewarding using the OpenGL framework. This was my first term incorporating frameworks into programming. It was quite interesting to look at the code behind the code, so to speak. I also got good experience making sure you include the correct packages and declaring the correct path for the header file. One specific issue I had was making sure to implement the stb\_image.h file. But once I ironed that out, I was able to utilize my own .png file of the scary cat to be the texture on my cube. I also played around with the vertex coordinates of the texture to manipulate how the image looked. It was a great experience. I also added a plane to my scene, but it somehow ended up on the left side of the cube instead of the top. Trying to think in 3 dimensions was a challenge for me despite living in a 3d world. After generating that image, we added a camera, and then eventually lighting.

Adding controls to the camera made it very easy for the user to traverse my scene. We wrote several if statements that included the glfwgetkey and GLFW\_PRESS to navigate the camera around the scene. All the iterations of my final project include the camera functionality. I was happy to be able to include that and see it work without crashing the program. We included the camera.h file and recorded inputs such as key presses and mouse scrolling. From that point we added the lighting, which I was never able to get quite right with my final project but understood how it was used in the correct way. I was really surprised by the amount of Math was used in generating projections and views. I generally consider myself good at math but do get humbled at times. The glm math inclusion had us use matrix transformations to generate the model view and projection matrices. The final project brought all these new skills together to create a functioning project.

Creating the final project was a cumulation of numerous hours put into each module developing a modular and organized code. The markdowns in github and the tutorials in visual studio were the biggest help in generating quality code. Each markdown was very descriptive and helpful. Learnopengl was structured a bit differently, but that helped as well. I had to meticulously go through each tutorial to even produce anything worth submitting. With each module tutorial and markdown I made sure to comment as much as possible. Sometimes I would get moving too fast and forget to make a comment. The code developed created a scene and let the user navigate the scene with the camera, per the prompt. It is very reusable in many ways. First, the code is structured well enough that it can easily be improved by someone who has a lot more programming skills in opengl than I do. Someone that was very skilled at it could add many more sources of light, a lot more content, extra camera features, and the 2d to 3d button. So where my code falls short on some of the prompts, it is at least still a working program. That is much better than I have submitted as projects in the past. Normally I tend to break programs, but with this project I can slowly see myself working through problems and figuring out solutions to keep applications working. My programming skills have increased tremendously.