Final Project Reflection

William Neal

SNHU

CS330: Comp Graphic and Visualization

Professor Enkema

9/24/2023

1. **Justify development choices for your 3D scene**. As you write, think about why you chose your selected objects. Also consider how you were able to program for the required functionality.

The choices I made for my scene were no different than anyone else’s code before me. Even though I greatly struggled in this course, I learned someone else has done this before me and someone else has done it better. The resources were there I just needed to find them. If we look at camera movement and camera, that is all really standard par for the course code. Glm::vec3 is just x-y-z vector components. Made meshes and indices for all my objects, created my window, built out my lighting, connected up my mouse and textures, and created my breakdown. There was nothing revolutionary about my project. I think the only thing that could have been slightly different on my project compared to others was my lighting. I spent a lot of time trying out different lighting sets and adding and subtracting various lights.

1. **Explain how a user can navigate your 3D scene**. As you compose your thoughts, discuss how you set up to control the virtual camera for your 3D scene using different input devices.

Well, I think one of the first things I messed up during this class was moving backwards. My logic was basically the same for backwards as it was for down. The reason I bring this up is because our input effects our camera, especially when we start factoring in mouse movements. I really could not figure out how to move freely in the space as if it was a 3D video game. I hit a point were I could not turn around and look behind me. I also do not have a functioning scroll on my mouse (thanks Razer!) which limited my ability to QA check things.

1. **Explain the custom functions in your program that you are using to make your code more modular and organized**. Ask yourself, what does the function you developed do and how is it reusable?

Again I was most proud of my lighting, maybe this was because I had finally got stable objects with a plane finished. I felt like I was flailing all class so when I was finally getting something to click I felt good about it. I had my calculate direct light which took in our directional light source, the fragment direction and our normal vector of the fragment. It was a function that calculated the resulting color of an object when light hit it. Now I know I added texture now, but my first submission I did not. I still do not think the textures look good on my objects but I understand they are in the requirements and rubric, so I added them. The point I was getting at is, when I would move around the 3D space I could see the light changing the color of my objects. Is the code reusable? Yes, one hundred percent. Anytime directional light is on an object calcdirlight is going to calculate the effects. This of course is just a small example. I think a lot of the code in the project was super reusable and a lot of sections were copy+paste+add in your own variable name. Also grouping various functions makes sense, whether it’s on an OpenGL project, a console program, or a web-based application. That’s just good basic readability practice. Overall, towards the end of the course I felt like I was finally seeing the patterns with the verts and indices, I felt like all the information I had tried biting off during mod 2/3 was finally making sense. I tried throwing a bunch of stuff at the wall to see what stuck, and it all stuck just my aim was bad and I needed to layer it better. I can appreciate the patience and organizational thinking needed to be good at this. I was looking at OpenGL jobs on google and saw Lockheed Martin, Argo AI, and Roblox. So, there is still need and some varying industries in demand for this kind of code, it’s just not a strength of mine yet. I am not saying without prolonged time and practice I wouldn’t get better, just that in 6 weeks I am not much more adept than what I was at week 1.