CS 330 Reflection (Design Decisions)

These last eight weeks have been challenging and eye-opening. I never fully appreciated the work behind the game engines, libraries, and software we use that depend on OpenGL to display visual elements to its users. Being exposed to the process of using OpenGL to create objects, using it to navigate 2D and 3D space in a window, and using my prior experience to modularize the rapidly growing code for a seemingly basic 3D scene also made me realize how much I take the libraries I use at work for granted. Even though I was challenged throughout the duration of this class, I feel good about the progress and decisions I made for my 3D scene. As I was nearing the end of my project, I was looking at it and realized just how amazing it is that I could create a 3D world with just images and vertices. Although I have much more to learn, I feel excited about what I can create given more time with what I already know.

When developing my 3D scene, I had a lot of choices to make given the limited time I had, especially when I was already a week behind. I knew there were different shapes that could create what I needed but I also knew that depending on the shapes I used, it would vary the complexity of the object I was trying to depict. I also wrote the code in a way that it would be easy for me to understand at a later time what I needed to change if anything. For example, at first, I was creating the same cylinder objects in multiple vertex arrays and then decided that a better decision would be to use the same one for some of my objects and then use the scale function to change the size of the cylinder. This reduced the amount of GMesh instances I had, made it easier to navigate the code, and was much easier to tweak the size without changing a ton of values in an array. As a result, the two cylinders used to make the support for my monitor’s display, the flattened cylinder/circle I used to make the leg for my monitor, and the two cylinders for my mouse bungee are all using the same VAO even though they have different sizes and rotations. If I had created a mesh for each of these objects, I could have had 450 more lines of code that would need to be maintained and tweaked which would have wasted a lot of development time. I know its possible to put multiple objects in the same VAO, but I refactored the code so that I could create a GMesh object for each 3D shape that I wanted in my scene. I found this to be better (at least for this project) because it allowed me to make changes faster by organizing the code so that each array was responsible for a single shape. I used comments and gave the GMesh object a name field so that I could easily search and modify anything I wanted and find it quickly. Even though I am new to OpenGL, I feel confident that my design decisions made a better development experience and will prove useful even I ever need to come back to this project in the future.

Extensive scene navigation options are important to me because not only does it make development easier, but it also adds to the experience of viewing the scene in general. First, I worked I worked on the forward, backward, left, right navigation (WASD). I wanted it to feel “good” moving around the scene by default. I didn’t want it to be so slow that it took forever to navigate large portions of the scene, but I also didn’t want it to be so fast that you couldn’t make precise movements to view the parts of the scene that you wanted with ease. I believe if it doesn’t feel at least somewhat right out of the box then you are doing something wrong with the design of the program. I made the up and down movement (Q and E) the same speed as the WASD movement speed because I feel like the default feels smooth for up and down movement too. The mouse cursor lets you control the way the camera is looking, there wasn’t much tweaking with that, it makes it easy to view up and down angles to make sure that the vertices were done correctly. The scroll wheel allows you to change the speed of movement, and I made sure to tweak it so that the speed difference was noticeable quick enough, but without leaping too much that it felt hard to control the variation. All in all, I am happy with my decisions in this area.

Modularization and separation of concerns is a big thing for me and an important step in good code practice. I implemented the common user defined prototype functions and separated each 3D object to its own mesh to keep the code organized and easy to change. I didn’t make too many new functions as they weren’t exactly necessary for the size of my code, but I also made sure not to put everything into just the main function like a lot of online resources do. Everything was spit into creating the shaders, creating the meshes, and rendering the meshes with the shaders attached. I feel good about the modularization of the code, and I am confident that I can come back at a later date and modify it without much hassle if I chose.

In conclusion, although I have limited experience in OpenGL programming, I believe I made the correct design decisions for a great project. I also believe my prior programming experience helped me come up with ways to organize the code in a way that will make it easier to edit later. I am happy I had the opportunity to gain some experience is this type of development.