Reflection:

Design Decisions

David Vega

Southern New Hampshire University

CS-330: Comp Graphic and Visualization

Professor Malcolm Wabara

02/25/2024

**Justify development choices for your 3D scene**

My development choices for my 3D scene rested on several factors. Since the premise was to render an approximation of the object, it isn’t critical to be precise. However, to achieve an aesthetically pleasing rendering, precision was factored in as best as possible. Aesthetics is another factor when considering my 3D development. I wanted to make sure that the objects had sufficient contrast and complimented each other well. Modeling technique was crucial in this development phase as the more complex the object was the more polygons were necessary and it can get rather complicated, so I tried to keep the polygon count low. The use of textures also played an important role in development as I didn’t want something that was too detailed that it wouldn’t render properly especially when using multiple polygons. Unfortunately, there is only so much detail you can see with a black object, so the final factor I considered is lighting. Given these objects were placed in a very well-lit area with plenty of natural light, I thought additional lighting wasn’t necessary. However, as the day progressed, you could see the change in intensity as the sun starts to go down so I added some additional lighting where the sun was coming in from the window from the left of the objects.

**Explain how a user can navigate your 3D scene**

A user can navigate my 3D scene via Mouse and Keyboard. WASD keys were set up to control horizontal, vertical, forward, and backwards navigation. In addition, the user can also use the scroll mouse button to speed or slow the movement of the camera. The mouse can also be used to move the camera in the world space.

**Explain the custom functions in your program that you are using to make your code more modular and organized.**

The custom functions I used in my program were primarily for initial setup such as initializing a window, rendering, creating shader programs. Most importantly, were the use of custom functions to create textures and different kinds of meshes. For example, the meshes I used in my program were a box mesh, cylinder mesh, and a plane mesh. Each of these implementations had arrays that contained vertices and/or indices data. Each primitive can contain a few vertices to hundreds of vertices depending on the object you are trying to replicate. The benefit to making these custom functions is that they are reusable and allow me to implement a new primitive just by calling the function. This made development more efficient since I did end up having to implement multiple boxes and cylinder meshes. This would have been time consuming having to enter in the vertex and/or index data each time I wanted to use the same primitive. It would also make the code harder to read due to the length of the code blocks. I think in a subsequent project, I would also modularize the structure of the code. In other words, I would have created the custom functions in one .cpp file and then just <include> it in the main .cpp file. I would also do this for the initialization of the window. Doing a modularized structure helps to organize the code and make it easier when debugging.