CS 330

7-1 Reflection

Mihir Patel

[mihir.patel@snhu.edu](mailto:mihir.patel@snhu.edu)

Southern New Hampshire University

**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.**

I selected to create a shelf for this final project since it seemed difficult enough to challenge me but not so complex as to overwhelm me. This sort of coding isn't something that comes readily to me, and there was a significant learning curve for me throughout this lesson. Having said that, I was able to program for the essential capabilities. The finished code submission shows a 3D shelf textured using wood, with a source of light to the right of the 3D shelf. Furthermore, when the software is launched, the user may freely move the camera as well as the shelf using the WASD keys. It took a lot of trial and error to get the light source to operate properly, but I'm pleased with how it turned out.

**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.**

As previously said, the user may move the camera around with the mouse to traverse my 3D shelf. If the user presses any mouse button, the camera will freeze for the duration. Furthermore, using the WASD keys while trying to hold down either mouse button will enable the user to move the shelf in the desired direction, depending on which key is pressed. The console will show the user precisely what they are doing by displaying the key they are pressing or holding or releasing when executing all these activities.

**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?**

To go along with my vertices[] array, one of the big things I did for the code to simplify things for myself was to establish an indices[] array. This allowed me to reuse vertices in several locations, removing the need to create numerous more vertices unnecessarily. This aids in optimization and is something I'd expect an OpenGL-powered game to accomplish as well. Aside from that, I made certain that my code was well-organized in terms of space, the number of comments, and the chronological order of my functions. For example, in the start of the.cpp file, I specified all relevant variables/prototypes, then created the shaders/vertices, then the keyboard functions/texture function, and finally the main method. This seemed like the most reasonable method to structure my code to me. My software also contains an optional function that may be toggled on or off (by default, it is commented) that will continually show the precise coordinates of the user's mouse within the bounds of the window whereas the program is running. Because of the commenting approaches I used, I feel if another developer took my code and wanted to build on it, they'd be able to rationally follow my thought process and simply add new functions and change the code without breaking anything.