Eric Boilard

Southern New Hampshire University

CS-330 Final Project Reflection

December 12th, 2021

### **Reflection**

**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 chose my scene because it was simple enough that if I struggled too much, I could still create it with the basics but then also if I felt confident in my ability, I could create some additional complexity in my scene with the same objects. The objects also covered many of the bases that I needed to hit in respect to the type of primitive shapes being used. I was able to create a plane, a cube, a cylinder, a sphere, and a complex shape using triangles, cubes, and pyramids. For example, the spheres that I created to represent tomatoes were created by taking a pyramid and rotating and translating it in a for loop until it made a ring. Then I made another for loop that took that ring and multiplied it and then spun it on a single axis. Doing this I was able to create three spheres and place them on the cutting board planes. Another example is the knife which was an altered pyramid to form the blade and then a cube stretched to form the handle. The cylinder was two circles connected and those circles were made by taking a triangle and rotating it into a circle, like how a pizza is cut. So, the scene itself along with the objects were simple enough but also flexible enough for me to take different approaches in coding them.

**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.

The user can navigate using the wasdqe keys on the keyboard and the mouse/scroll wheel. This allows the user to fly in any direction and to view the scene from any angle. The virtual camera was setup outside of the main loop and was not included in the rendering of the images, instead keyboard and mouse input connected to the camera happens outside this main function and is recognized by a callback function which means that whenever a key is pressed that is coded for or a mouse movement, it is then noticed and the proper movements are then made in the scene based on camera variables set up at the start of the program.

**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?

Some examples of custom functions would be the functions for importing textures, mouse and keyboard input, and vertex buffer and array objects. By making the mouse and keyboard input and controls at the bottom of the program and all lumped together it makes it easier to understand its purpose but it's also out of the way. The vertex buffer and array objects really help to make the program simple because when drawing each object, I can simply just call the VAO that I created and it imports all the information regarding that object like vertices, color, textures, Normal's. The keyboard and mouse input are quite simple and can just be copied and pasted into any other opengl project to add the functionality. The VBO and VAO functions can also be copied and pasted with only changing a view item to make it work with almost any other opengl program. The function for importing textures is also quite simple to use if you are also using the soil2 library in which you could also copy and paste and change a view variable. If one really understands OpenGL like at the end of this course than you could expect the ability to copy and paste the entire program and only change a few things like vertices, textures, colors, Normals, and the information for model view projections and it should be very useable as a template for getting started.