Nicholas Glover

4/15/2023

**CS-330: Final Project Reflection**

When deciding what objects to include in my 3D scene, I thought about different areas I find myself in, such as my workspace. When I work on my schoolwork assignments, I sit at my desk and work on my laptop. I’ll usually have a snack of some sort and a drink. At the beginning of the course, when we were assigned to create the 3D scene, I was enjoying some praline pecans, and Pure Leaf iced tea, so I included them in my 3D scene along with my laptop. Since I needed to include another item to meet the requirements, I also included a box of my morning tea. I decided to use these items because they were diverse and were complicated enough to challenge me without being too difficult. I was able to program for the required functionality by setting up basic shapes and duplicating, then manipulating them through translation, rotation, and scaling in order to recreate the 3D objects.

Deciding how to navigate the 3Dscene was actually fairly easy. I like to play an online game called Warframe. In the game there is a piece of equipment called an Archwing, it allows the character to fly in all directions in the mission space. I utilized a simplified version of the Archwing controls for navigating my 3D scene. The W, A, S, and D keys move the camera forward, left, backwards, and right respectively. The left Control button moves the camera down and the Space bar moves the camera up. Moving the direction of the camera is controlled by the mouse, moving it forward rotates the camera up, moving it backwards rotates the camera down, and moving it right and left rotates it right and left. With these controls, the camera can easily and fluidly navigate through my 3D scene.

While my code is not the most modular, it is well organized. I have multiple methods to control the way the camera moves and several other operations that occur during the simulation loop. The other methods I created mainly handle the creation of the shader programs I use to render the 3D scene. I created the methods because the compilation of shader programs would clutter up the rest of my code if it wasn’t kept modular. It also makes it much easier to compile the multiple shaders I need to maintain the lighting and effects in my scene.