Final Project

Brandon Cook

Southern New Hampshire University

CS-330: Computer Graphic and Visualization

Professor Eugenio Rodriguez

February 18, 2025

Final Project

When creating the 3D scene, I separated each individual object into their own functions. This cleaned up the code considerably, allowing easy access to each object to make adjustments. Two of the objects are “complex,” using two objects to create one. The mug uses a half-sphere and a half-torus to simulate the mug and handle. To match the image (found at the bottom of this paper), a green metal texture was used. The pie is the other object, though it does not use two different shapes to create. Both objects are prisms, stacked on top of each other with different textures applied to give it a layered appearance. It is sitting on a plate, which was made from a half-sphere with the y-axis scaled very low. Finally, there is the cup, which is made from a tapered cylinder with a clear glass texture. While a backdrop was not required for this scene, I believe that my decision in adding one helped to add a layer of depth. Lighting is minimal. There is a directional light positioned behind the scene, and two point lights at the front of the scene to provide enough visual detail. The combination of shapes and chosen textures helped to match the provided image to meet requirements.

The camera is controlled by mouse and keyboard. Moving the mouse allows the camera to look around, while the WASD keys are used to move the camera within the 3D space. There is support for full 360 degree motion around the world to view the objects from every angle. The camera can be raised and lowered with the Q and E keys, respectively.

Each segment of code was isolated to its own function, allowing for reusability. Each object is its own render function, so the objects can be pulled from this project and implemented into other OpenGL projects. Additionally, scene lighting, textures, and materials occupy their own function. These keeps the code clean and organized. Functions are called when they are needed, preventing extra lines of code from being run and making the impact on memory minimal. Modular code can be reused in other projects to prevent the need for the developer to begin completely from scratch each time they begin coding, saving time, energy, and headaches that could arise from problems encountered that have already been solved previously.

The picture matched in the project is below:

