Devin Wheeler

7-1 Final Project Submission

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

From my 2D image, I chose to make a mug (including the tea tag and string), cutting board, grapes, sausages, counter, and tea box. The reason for these items is because they made up most of the original image. And it is important to capture the overall feel of the scene. To achieve the required functionality, I focused on using basic shapes such as cylinders, boxes, spheres and toruses. For example, the mug is created using a cylinder for the body and a torus for the handle. The Mug also uses different textures, one for the outer body and one for the top, which will make it appear closer to the image. Approaches like this allowed me to create realistic objects easily.

1. **Explain how a user can navigate your 3D scene.** I added basic but fundamental controls to help the user in navigating my 3D scene. With the W, A, S and D keys you can move the camera forward, backward, left and right. Pressing the Q and E keys will cause the camera to move up and down. The mouse also provides controls such as camera orientation. This allows the user to look around with the cursor, changing the horizontal or vertical orientation. The speed of the camera can be changed using the scroll wheel. This combination of mouse and keyboard controls make it really easy for a user to move throughout the scene.
2. **Explain the custom functions in your program that you are using to make your code more modular and organized.**

To keep my code organized and modular, I made functions for each object. For example, the DrawMug function handles everything needed for drawing the mug. This includes setting its sizes, applying textures, and rendering the different parts that make up the mug including its body, handle, tea tag, and string. This makes it easy to adjust any of these parts without messing up the rest of the code. I also created functions like DrawSausages and DrawGrapes to better handle rendering multiples of the same object. These functions use arrays to handle multiple parts of the objects like their sizes and positions. They render the objects using a for loop instead of declaring each object separately. This way I can easily change the number, size, or position of any of the grapes or sausages in a single easy to find location. All my functions focus on a specific object or task, making it modular and easy to maintain! This also makes it easy if I need to modify part of any object like size, I can easily find the object by its function instead of looking through a huge code block. This approach keeps the code clean and makes it easier to debug.