Austin Layman

CS-330

Professor Diesch

December 11, 2022

**Final Project Design Reflection**

In the beginning, the result of this project was expected to be a cup of tea, some tea bags, tea rusks, and a tangerine on the side, but this vision evolved over the course of multiple weeks. I personally prefer coffee over tea, so it only seemed natural to create a scene that reflects my preference. Following that change, I had to slightly alter the rest of the scene to more closely fit the coffee motif. This left the construction of the objects relatively unchanged, with the exception of the tea rusks which were stretched to resemble biscotti biscuits instead. The entire redesigned scene consists of a coffee mug constructed with a cylinder and a torus, two biscuits created by using a stretched cylinder, two packets of stevia created with cubes, and of course a table to place everything on which is created using a plane. One thing to note is that the tangerine is missing entirely with no object in place to replace it, this decision was made once I started looking at the complexity of the shapes in my scene. Per the program requirements the scene must have less than 1,000 total triangles, so adding a sphere that I would be happy with would have put the program over that limit. Of course, I could have simply reduced the polygon size of the other objects, but I wanted the mug to be the center of attention in the scene so I utilized the triangles that would have been used in the sphere to add to the detail on the mug. The lighting in the scene consists of three fill lights to illuminate every bit of the mug, as well as a top-down key light with a slight yellow tint to create a more warm feeling in the scene.

Perhaps one of the simplest pieces of this program in terms of length of code and custom function implementation is the camera used to navigate the scene. On the face, it is a simple camera that any user that has slight familiarity with video games or computer visualization in general would be able to operate. The same common controls are applied to this camera as a number of other media. If you take a closer look at the camera, however, you will see a number of matrices and vectors added and multiplied together to get a final transformation matrix for the scene objects that is run every single frame. The camera utilizes multiple callback functions to take the appropriate user input and update in real-time.

Modularization of the program was one of my highest priorities while thinking about how to structure the code. To accomplish this, I created classes for the major objects as well as functions to create primitive objects within the Mesh class. Creating the complex objects in the scene like the cylinder and torus is done by first generating an array of vertices and indices. The cylinder takes in the number of sectors, radius, and height. Vertices for two circles are then generated, one at half the height and one at negative half height, these two sets of vertices are then connected by generating a corresponding index array. The torus is a bit more complicated, but the premise is the same. This function accepts arguments for the number of main segments, tube segments, main radius, and tube radius and subsequently generates a vertex and index array. Additional modularization was achieved by creating functions for translation, scaling, and rotation of the meshes as well as several other commonly used code chunks such as drawing and generating textures. Calculation of normal vectors, perhaps one of the most complicated parts of this program, is also a custom function. This function takes no inputs and simply calculates a weighted average by taking the cross product of two triangle edges and adding the value to the corresponding vertex normal. This allows even complex objects to incorporate the specular element of the Phong lighting model.