Ariana Williams

Design Decisions

I choose to use two rectangular prisms to create the book. I made one slightly smaller than the other to give the illusion that the book has pages. Next, I used a cylinder to create the coster and the cup that is sitting on the book. Then, I used a sphere to create the globe, a cylinder that is holding the globe and a cube as the base of the globe. I added texture to the plane to make the floor, it is a carpet. In addition, I added texture to the book pages to make them gold and the stand of the globe. I used a separate class to create the cylinders, I chose to do this because it makes the code more cohesive. If there is an error with the cylinders I can go directly to the class and quickly resolve the error.

Users can navigate my scene by using the WASD keys or a mouse. The user can use the W key to move forward. The S key to move backwards. The A key to move to the left and the D key to move to the right. Users can zoom in and out and rotate around the scene. The camera is positioned in a first person point of view. I allowed the orientation of the camera to change but the physical camera to move. As the user moves around the scene they can see how each object in the scene connects to each other. To enhance my scene I added lighting. I added ambient light, this is considered a “natural light”, but it can be created artificially. I created ambient light using an artificial source. Diffused light is light that is scattered, this light is evenly distributed and leaves no sharp shadows. I created diffused light in my scene by creating a spot light. Next, I used the x, y, and z coordinates to place the objects on the plane. I had to use precise coordinates to make sure that all the objects connected in the way that I wanted.

Code modularity is used to help programmers read, understand, and reuse code in future projects. There are several ways to create code modularity. One way I ensured that my code was modular by creating classes. Another way that I ensured that my code was modular was creating functions, these functions can be reapplied throughout the code. I also made sure that I used proper naming and grouping the code together. In addition, I used best coding practices and comments to make it clear what is happening in each part of the code. Finally, I used a consistent naming scheme and limited code line length. There are downfalls with code modularity as well. Some of them include the modules are too small. This causes the program to have too much overhead. Another downfall is centralized dependency, this causes issues if there are multiple users using the program.

There were many things that I learned during this project. I learned that I can take an image and break it down to basic shapes to recreate the scene. I also learned that I can enhance the scene by adding textures and lighting. Lastly, I learned how to make my code modular so that it can be easy to use and reapplied to future projects.