Trevor Corliss

CS-330 2025

Final Project Reflection

I chose to design the objects I did because there are two common place objects and two objects that are not as common, with that combination of objects two are simpler in design while two are complex in design. This gave me a more challenging experience when designing the scene this term. The objects I chose, the book and the rubik’s cube were the two non complex objects, the challenge in designing these lied in the texturing of the objects. To create the book I created a box that was relatively close in perspective to the size of the and gave that a texture to represent the cover, to create the pages in the book I created a slightly smaller box and offset it by a tiny amount to allow it to look like pages of a book with the shader color I chose for it. To create the rubik’s cube I laid six cubes on top of each other each with a separate texture for each color side, then I offset the cubes in each direction on each axis to not have artificing due to overlapping textures. I ran into a problem when I first applied the texture to the cube faces because I didn’t use any texture coordinates first, one I used the texture coordinate the cube looked just like the real thing. When designing the complex objects, the soda can and the amplifier, the process for these was a lot of trial and error to get the dimensions and location right and true to the source image I created. The can was the most difficult to recreate accurately due to the curves in the shape of a can, the lack of fine tuning on the torus shape didn’t allow for me to make the best representation in my scene. I was able to effectively add textures and texture coordinated to every object in the file by referencing the code in the 1-2 OpenGL sample.

In order to navigate the scene users can move the camera using the following commands:

W - Camera forward

S - Camera backward

A - Camera left

D - Camera right

Q - Camera up

E - Camera down

To change perspectives in the scene:

O - Front orthographic view

2 - Side orthographic view

3 - Top orthographic view

P – Perspective view

The custom functions I created inside my final project are functions to create the objects in my scene. It is incredibly easy to implement new object creation functions into this program ad you just need to plug in the function into the SceneManager.cpp file and the SceneManager.h file. After the function call is referenced in both files it will execute and render the additional object into the scene. Since the object creation functions are some of the bulkiest functions in the file, grouping each object into its own function makes the debugging, revision process more streamlined. In my SceneManager.cpp file the object rendering functions took up about 500 lines of the file, once I condensed them navigation in the file was significantly easier. The file was also much easier to read and comprehend after breaking them into functions and adding eye catching comment blocks to signify start and end points to specific functions.