Skylar Walker

10/18/25

CS-330

It took me a while to figure out what I wanted to use as my scene. I thought about my desk or a playmat with a deck of magic cards on it but the ideas I thought of just seemed too simple. The playmat with just a deck of cards and a die seemed to be basic with only one complex shape being the die. I settled on my computer because I knew it would be a challenge for me. Being able to figure out what shapes were needed to make a basic rendition of the computer was fun.

The different light sources that would come about with the fans made it more complex. As well as the number of shapes it took to create them. Each fan has 17 different parts. I would code one shape and then decide how big it needed to be with how it would line up with the other pieces of the fan in order to create an accurate object. While I was looking for textures, I came across a site that has a wide selection of different textures from wood to brick, to even the chip texture I used on the motherboard. I was able to make some glass panels for the front and one side of the computer case just like my own by changing its color values to a low white as well as its opacity. I found out it has to be rendered last or else I was not able to see the objects on the inside of the case until I took the camera inside.

The controls were easy due to some of the functionality already being in the program we were given to use with our project. The camera.h file was helpful in understanding how the camera controls worked. I was able to use the camera.h functions within the viewmanager.cpp file to allow Q and E to move the camera up and down respectfully as well as utilizing the mouse scroll for adjusting the speed at which the camera moves within the scene.

There was a lot of repeat code that I had to shrink down to the best of my ability. I was able to cut out things such as the objects size from each object if it was the same as the previous one drawn. The same goes for a few objects’ positions. The middle supports for the fans all share a position and are just rotated. The ability for certain aspects of the objects to carry over helped in cleaning up the code to some extent. I even used this method when it came to the textures. A lot of items within the scene used a black plastic texture. Because of this I was able to play the texture at the top of the code and allow it to be used for all parts underneath until I came to a part that needed something different. I also did the same with the material. I used relative code for searching the directory for the textures since the project would be run on other devices besides my own. This helps ensure there are no missing textures when someone tries to run the program.