Design Decision

The choice for my scene was honestly due to us buying a home and remodeling the kitchen. I thought about how it would be interesting to have something like this when preparing a remodel. I’m sure there are programs out there that probably create this kind of thing much easier, but I wanted to see how it would look through OpenGL. Truthfully, for a couple weeks there I thought I bit off more than I could chew but seeing the final environment I am very happy with the outcome. I’m sure I could have chosen something easier, but it made me learn more about OpenGL through doing much more than just simple shapes. I think the most fun I had was creating the chairs because it had so many pieces that it involved but it came out looking very good. There was a lot of trial and error getting the rotation right to have the legs properly turned inward.

Navigating throughout the scene is very simple and anyone who has ever played a computer game on mouse and keyboard will be able to pick it up quite quickly. I made use of the custom camera header file for creating the movement. The WASD keys are used for Forward, Left, Backward and Right camera movement, respectively. The custom camera header file also had Up and Down functions to use as well so I set Q for Up and E for Down movement. The mouse cursor is also utilized by allowing the user to look around the scene as well as directional pointing. Whichever way you turn your mouse you will go towards where the cursor is pointing, even though we don’t see the cursor because the window locks it in. The scroll on the mouse is used for movement speed increase. Scrolling up will make the camera speed increase while scrolling down will decrease the speed. All these things make up the movement in the scene.

The code was written with OOP programming structure and made us of early declaration of functions and then defining them later. I used classes for cube, pyramid, sphere and torus to create different objects in the main code. I wanted to keep the main function clean, so I called functions only instead of writing full code in there. The code is written in pieces through these functions. This makes it easier to locate where something fails because only that function will need to be fixed instead of everything together. It also makes code reusable because all the header and class files can be used in other code but the functions in the main source code can also be reusable. The meshes, all callback functions, system initialization, and texture creation function can all be reused. Just creating a new rendering function will make use of all the rest of the functions.

I’m happy with the overall outcome and, even though it’s very primitive, it looks very similar to the 2D image I chose. I did cut some corners with the cabinets by making the rectangular prisms act as cabinets without any breaks to show the cabinet doors but I think the basic idea is there, especially since the cabinets in the 2D image aren’t very complex anyway. I learned a lot from creating this image and I’m going to use this knowledge in the future because I would like to work in game development so creating such a scene was really fun.