Reflection

I chose the selected objects because I believed that they best fit the vision that I had for my project. While I couldn’t quite figure out how to create wires or have time or energy to create the controller, I still believe that the shapes that I chose to make my objects were the best that I had available. For instance, using box meshes to create nearly all of the dresser seems as though it was the best choice. It made the legs, the drawers themselves, the top of the dresser with an added protrusion on top and a lip gave it that “carved” look that the dresser in the image had. As another example, the shapes to make the pizza were also good decisions. The initial cylinder shape was used to take the form of the inner part of the pizza, while the extra torus shape gave way for the crust. Finally, using multiple, different sized cylinders and a box mesh to create the PlayStation not only made the console look as if there were spaces between the cylinders and the box mesh, but it also allowed for different parts of the object to give off shimmers from the specular lighting effect to really emphasize the color and reflectiveness of the light.

I set up the camera using multiple variations of maneuverability using simple lines of code, usually by copying a few parts of the other ones. For example, implementing the straight up and down movements was fairly easy, as I simply added two new button presses and addressed them to move the camera straight up and down. Although, adding the ability to control the speed of the camera was much trickier. Since there’s no specific, built-in method to use the mouse wheel as a button press, I had to get creative and create a new Mouse Scroll Callback method that took the action of the mouse wheel being offset by a certain amount and set that amount in either direction to either speed up or slow down the rate of the camera speed. In addition to this, I had to make sure the movement speed didn’t go haywire if someone were to increase it to infinity or decrease it enough to go into the negative by setting an upper and lower limit to the speed change. Finally, I introduced another method to change the camera view between orthographic and perspective view. This allows for those who wish to see an orthographic view without the distortion of perspective view, as perspective makes objects further away seem smaller than those that are closer. In some cases, like dealing with the need for precise sizing or alignment, dealing with UI overlays, grids, tiles or CAD-style visuals, or even simply not wanting depth to influence scale, orthographic view can be very beneficial.

This was a very interesting and challenging project to work on. The Mouse Scroll Callback was very fun to figure out and implement and took quite a few tries to get just right. I also believe it was a great idea to begin with, as there was not much to work with when it comes to pure OpenGL, rather than using other programs like Blendr or Cinema 4D to implement things like shadows. One more method that I am proud of has to be the for loop that I used to make the stack of games. By the looks of my code, I couldn’t quite figure out how to simplify my mesh building code. At least with this loop, I was able to make the stack without resorting to repeating the same block of code over and over again. Not only that, but with a bit of manipulation of the color shifting portion of the code, I was able to repeat a pattern of different colors of the shapes in order to give it that look of many different games, rather than just a stack of black or red cases, or what have you.