Week 7: Project Reflection George Kaline III

For my scene design I chose four objects to use from my everyday life. The reason I say everyday life for simple shapes was the objects are toys that my daughter plays with daily. Ironically enough the items in my scene were already on the kitchen table and felt like fate that those were the objects I would use in my scene. These objects are a pyramid, a cube, a cylinder, and the kitchen table(plane). This also allowed for the programming to follow a simpler path forward and was not a complicated object to create. The pyramid, plane and the cube all used the method GL\_TRIANGLES to draw the triangles. GL\_TRIANGLE\_FAN and GL\_TRIANGLE\_STRIP were used for the cylinder. Those objects have their vertices and indices laid out in each section of the code along with their color formatting and texture coordinate system.

For the project and navigating the scene, there are seven functions programmed and mouse movements. Those functions are key presses. Six of the key presses are standardized functions in the camera at each header file. The 7th key press was built to change from an orthographic to a projection mode. That was coded with an if and then statement referencing the perspective function in the camera section of the code. Here are the following key presses and their function “W” is used for a forward motion. “S” is used for a backwards motion. “A” is used for a left motion. “D” is used for a right motion. “Q” is used for an up movement in the Y axis direction. “E” is used for a down movement in the Y axis direction. “P” is used to change from a perspective mode parentheses default) to an orthographic view. The key press for “P” is not a toggle key press but is a switch key press so to maintain the orthographic view you must hold the “P” button down otherwise upon release it will go back to perspective mode. The mouse movements allow for a “free look” camera motion. There is also an included wheel mouse function that allows the user to zoom in and out of the viewed scene. The default camera view has you looking directly into the center of the objects. The default has an elevated view that puts you at +3 in the Y direction. In some cases, and in some programs you are often given a height adjustment in the event that you are performing a walkthrough of what the average person of a height would you use. Further incorporating a settings function would allow the user to adjust their height so the camera would view at their height so that they can field our perspective like theirs in real life.

The functions developed in my app allow for compiling in two separate classes in the event to make the main file smaller. I kept everything together in order to follow the code easier for me to follow. The functions listed are well documented in comments and have some special formatting for some generic add-ins to be used. Extensions used to allow for special color highlighting within visual studios can help break this out further and allow users to color coordinate coding. I found this very helpful when breaking down comments and sections. I viewed the functions as separate main tasks that had smaller subtasks assigned to them. Adjusting functions this way allowed for faster navigation through the programming along with being able to collapse or expand the functions to eliminate vertical scrolling when searching the cpp.