**Reflection**

I chose the selected objects in my scene due to them being difficult and allowing me to grasp a beginner understanding of OpenGL. This is the third time I have taken this class and game design was something I may want to pursue in my endeavors. This class was the perfect way to see if I enjoyed working with graphics. I chose objects in my day-to-day life that would be seen in a game and I wanted to know how hard it is to implement them. Since the sphere and box were already drawn with vertex values, I did not have much to change other than scaling, rotation, and placement with them. The complex shape which was the child’s toy that is shaped like a torus with the depth of a sphere caused a lot of issues. Since it was difficult to add more than one shape together and make them seem as one, I had to create four separate shapes for the final product. I was able to program for the functionality by understanding how objects interact on the x, y, and z-axis. For stretching an object, you must understand what sides will be affected while maintaining the correct ratio for the shape. Rotation also was a big factor as if you scale it incorrectly the rotation will not work correctly. Lastly, placement is the most important due to it matching the scene for my project proposal. All of this created my shape in the correct place with perfect ratios and rotation. Navigating the scene required me to understand how the event handler worked in OpenGL. I used frames to determine when to check if the user had pressed any hardware. As for example using WASD was how my program would move the camera up, down, left, and right. When running my program would check every frame if the any event was pressed and if so, it would them move the camera to a certain distance in the direction chosen. This speed could be increased or decreased using the scroll wheel on the mouse which used a function inside the camera object. A lot of the custom functions I created are modular and reuseable. This is due to coding practices that work only inside the program without grabbing information from my personal machine. Most of it is also building into creating an environment which has the basics of what would be needed to do so. For example, my DefineObjectMaterials() function works with lighting to create realistic lighting that would match how it reflects on the object texture. My object materials are types of objects which make them usable on a broad of similar objects in different shapes. As my gold object material could be used on any shape that is gold. This is also seen in the work I have done in my LoadSceneTextures() function. As I have created a path in the program for users to put their JPG or PNG files and then call them through that function to load the textures onto their shapes. All the code in my program was created to be used on any OpenGL project. As now if I wanted to I can now create a new scene with everything I have used or even add onto this scene without having to change any functions to do so. If I want more complex shapes I have all the basics needed to do that, and if I feel as if I do not I can work inside my ShapeMeshes.h to add onto it. Everything is modular and can be grown without having to completely redo functions and I feel like I have learned a lot creating this project.