**Design Decisions**

I started off my design process by making a lot of mistakes. At first, I decided I wanted to re-create an image of the Bass Pro Shop pyramid in Memphis, Tennessee. Now that I have started working on my project, I can see how that image really wouldn’t have been possible for me to recreate in this class. The angle and complexity of the image would have been just too complex to accurately recreate in the scope of this class. Thankfully, I had a small glass pyramid at home I was able to take a picture of, so not all my work had to be scrapped. With the pyramid as the basis of my project, I added several objects I found in my house that I thought I would be able to recreate using OpenGL. I chose a snow globe, a pyramid, a rectangular box, a saltshaker, and cube shaped piece of Tupperware. All these objects could be categorized into one of the three-dimensional shapes you provided us with in the meshes.cpp file.

Now that I had my image, I began to build my three-dimensional models. I started by using vertices to draw a series of triangles that linked together to form a pyramid. This was one of our first assignments, and seeing how a pyramid was the foundational element of my image, I thought it was a great place to start. However, when moving from the assignment to the milestone, which was the beginning scene for my project, I continued to use this method of drawing shapes using my own custom vertices. This was a mistake. At the time, I hadn’t realized that Professor Brian provided us with modular functions to implement various meshes that he had made. I had to go back a recreate the shapes using the meshes that Professor Brian provided. Unlike Professor Brian’s meshes, my shapes didn’t have proper texture or shader mapping. Using the new meshes I created a pyramid sitting on top of a flat plane.

With my mesh in the scenes view, I could now begin to implement rudimentary camera controls. I started by assigning WASD to their associated movement directions using the “UProcessInput” function. I also had to set up the camera to update each frame and use a formula to control the speed of these movements. Next, I assigned the E and Q keys to move the camera vertically along the y axis in a similar way. Now I needed to implement mouse movements and scroll wheel functionality to my camera. First, I used the “UMousePositionCallback” to track my cursor’s movement, then I used the “ProcessMouseMovement” function to rotate the camera on the x and y axis depending on how the cursor is moved. I used the “SetScrollCallback” function to assign the scroll wheel the ability to change the movement speed of the camera by using the “ProcessMouseScroll” function to track the user’s movement of the scroll wheel. If the user scrolls up, the movement speed will be increased until it hits the upper cap, or if the user scrolls down it will decrease until it hits the lower cap.

The last and most difficult camera control function to implement was the ability to switch the camera’s perspective between orthographic(2D) and perspective(3d) views with a simple key press. First, I had to assign variables to the camera’s positions. Then when the O key is pressed it will look at the variables and move the camera to the position assigned by the variables and set the view matrix to orthographic mode. I also had to make sure to disable mouse movement in this mode, so you couldn’t accidentally touch the mouse and mess up the 2d perspective. Finally, I assigned the P key to allow you to switch back to the perspective view matrix.

The next step was to create a complex three-dimensional object from my image. Originally, I was going to try to use a pyramid with a rectangular walkway for this, but that was two objects not one complex object. So, I decided to use a snow globe as my complex three-dimensional object instead. I was able to combine a cube, a cylinder, and a sphere together to make the snow-globe, and I think this ended up being a much superior choice because it used multiple shapes to create a single object.

At this stage I began to add textures to my objects. Using Professor Brian’s meshes made adding textures simple because these meshes already had texture mapping. Here is a list of the steps I took to map textures to the objects:

* Find a seamless, square, royalty-free image with a high enough resolution.
* Add the texture to my file system.
* Assign the texture to a variable within the code with an associated number.
* Assign a meshes texture value to true and use the correct number to assign that mesh the desired texture.

Taking this approach allowed me to add textures in a modular way. This was especially useful at the end phase of my development when I needed to change my original distorted plane texture to a texture that was seamless. All I had to do was overwrite the old texture with the new texture in the file system which allowed me to quickly change that texture.

A common theme you will find in this paper is that I made many mistakes at every point in this course. Thankfully, Professor Brian is very knowledgeable and good at teaching. Once I reached out to him, and started taking advantage of the meetings he offers the students, I was able to learn and implement the concepts for each week quickly and correctly. If there is one piece of advice I could give to future students of this class, it would be to take advantage of these meetings sooner rather than later. This course is progressive in nature, so building things on a sloppy foundation will greatly hinder one’s future progress.