**CS-330 Project 2 Reflection**

**David Waid**



**Justify development choices for your 3D scene.**

When deciding what object, I chose for my 3D scene I first reviewed the shapes that OpenGL was capable of creating. This gave me a basic idea of what I could choose from when looking around my house. Then I chose to start with objects that only would require one of those primitive shapes to create. That is where I chose the sphere, coaster, and charger block. All of these shapes require just one OpenGL primitive. The gold sphere is self-explanatory, it only requires a sphere to make. Next the coaster I decided would just take a flattened rectangular prism. Then looking at the charger block it would also take an altered rectangular prism which was easy to make with the scaling feature. After finding those, I thought about what items I would do that combines two primitive shapes and that is when I see my candle. Using two cylinders, one normal size and one flattened, I was able to make an object that resembled my candle.

**Explain how a user can navigate your 3D scene.**

When you start the program, I have various input devices such as the mouse, and the keyboard that will allow you to move around and explore my 3D scene. For the keyboard, we can move forward with W, backwards S, left A and right Dusing the keys. Then, we also have the ability to change the view of our 3D scene by pressing the P button we can shift from perspective view and orthographic view. We have the ability to use our mouse in multiple ways also. First, using the cursor we can look around directionally changing the way we look at the 3D scene. Then with our mouse scroll wheel we have the ability to speed up the movement speed of our WASD keys.

**Explain the custom functions in your program that you are using to make your code more modular and organized.**

Starting from the top down we have multiple functions that allow our code to be reusable. For example, we have our flipimagevertically() which helps with transforming your textures so that they are rendered vertically. Next, we have our CreateDummyWhiteTexture() function which is very helpful so that we can use just colors on certain primitive shapes. We have our UCreateTexture() function which will let us create a texture by using an image for input. We have our uProcessinput() function that allows us to process all our inputs for our keyboard in one function. Lastly, We have multiple examples of functions that allow us to process our mouse input also. These examples are all examples of making sure our code is modular and allows us to reuse it and scale our project while keeping it organized throughout the process.