**Justify development choices for your 3D scene. As you write, think about why you chose your selected objects. Also consider how you were able to program for the required functionality.**

I chose a coffee mug, coaster, notebook, pencil, and glasses case to fit a study desk theme. I chose the coffee mug and pencil to be used as my complex objects using more than one shape. I was able to program the scene by utilizing the provided shape meshes and OpenGL functions.

**Explain how a user can navigate your 3D scene. As you compose your thoughts, discuss how you set up to control the virtual camera for your 3D scene using different input devices.**

Users can navigate my 3D scene by pressing select keys on their keyboard. For moving forwards, left, right, and backwards, a user can utilize keys W, A, S, and D respectively. For moving up and down, a user can utilize keys Q and E respectively. Users may also utilize their mouse to look around the 3D scene and the scroll wheel for zooming in and out.

**Explain the custom functions in your program that you are using to make your code more modular and organized. Ask yourself, what does the function you developed do and how is it reusable?**

I used UInitialize to initialize GLFW, GLEW, and create a window. I used UProcessInput to implement user input within the 3D scene. I used UMousePositionCallback, UMouseScrollCallback, and UMouseButtonCallback to handle the mouse controls. The URender function held everything required to draw the shapes including the transformations and applying textures. The UCreateTexture function generates and loads the textures. The UCreateShaderProgram function holds the logic for creating, retrieving, and compiling the vertex and fragment shaders as well as attaching them to the shader program. These functions provide a clear organizational structure within the code to keep each task separate.