**3D Scene Project**

CS-330-T1189 Comp Graphic and Visualization 22EW1

by. Corey Nance

10/15/2022

As a developer for Triangle and Cube Studios that works with C++ and OpenGL, the task is to recreate a 3D version or a 2D image. To accomplish this task, the first step is to create a low-polygon 3D representation of a real-world object. Then next is to apply textures to this 3D model and apply lighting to help create a polished look. After creating the 3D scene, the next task would be to give user control with camera functionality that can traverse the X, Y, and Z axes.

The chosen scene is an image of a computer desk that has a laptop, computer screen, mouse pad and computer mouse. This was chosen to best display a 3D environment that can be modeled after an image of an actual workspace. Even though most of the shapes are similar, the placement and depth of the image can be well showcased. For the Laptop, a stretched cube is used for the bottom and top sections of the laptop. A cub shape is also used for the computer screen base and the computer screen itself. The computer screen stand is help up by a thinned cylinder with a small block cube being used for the computer mouse. Lately a thin plane is used for the desk as well as the mouse pad. To program these objects, OpenGl’s mesh is used to set the vertices and ambient lighting is used to show lighting on the objects. To try and get a more realistic look, OpenGl’s texturing features were used to apply realistic images for each of the objects.

Part of the design of this program is to give control access to the user. The 3D scene can be zoomed in or out, moved up or down or rotated either using the keyboard hot keys or using a computer mouse to change the view. The mouse scroller can be used to control the sensitivity of the movement. The A, S, W, and D buttons can be used to move the object left, or right, and zoom in or out with the escape button allowing for program termination. The Q and E buttons can be used to move the object up or down with safeguards in place to keep the scene from flipping upside down. The P button allows the user change projection from 2D to 3D. Overall, using the specific control keys along with the mouse gives the user full control to move the camera completely around the scene.

To have a more reusable code, each program functionality is broken up into their own functions. Each mesh for each of the objects are separated into their own functions properly named to reference what it is creating. The rendering, texturing, and lighting code is all broken up into their own functions. Doing this helps for easier maintainability as well as easy reusability. Each naming convention is referencing the objects its creating or the action it is doing. Code comments along with the naming conventions help to fully explain how or what the code is doing in that area. Overall, within this program, the main function calls the render function that then calls the mesh functions, texturing functions and lighting functions to fully render the 3D scene.

Overall, the created 3D scene displays each functionality to the client. Through this code, the client can experience how a 2D scene is recreated into a 3D scene. Functionalities display complete camera control as well as applied lighting applications. Through this scene potential clients can see how applying texture can give a more realistic feel. Putting everything together allows the client to get the full experience and see how it all works together.