Jarrod Helmers

Dr. Aly

CS-330

20 October, 2024

7-1 Project Reflection



In my 3D scene, I recreated a desk setup with a monitor, keyboard, mouse, and coffee mug. I chose these objects to practice modeling real-world items using basic shapes. For instance, I used a cylinder and torus to create the mug, boxes, and other primitive shapes for the keyboard and mouse. The monitor was made using a plane for the screen and a tapered cylinder for the stand. I kept the objects simple to ensure they each had fewer than 1,000 triangles, as per the project guidelines.

To add realism, I textured the desk and mug using royalty-free textures with 1024x1024 pixel resolutions. The wood texture for the desk and the ceramic texture for the mug create a nice contrast, making the scene visually pleasing. I carefully mapped the textures to avoid stretching or distortion.

For lighting, I used a point light to simulate a desk lamp and a directional light for ambient room lighting. I applied the Phong shading model, including ambient, diffuse, and specular lighting, to give the scene a polished look. The specular highlights on the mug enhance the illusion of a glossy surface.

Users can explore the 3D scene using various input methods. The WASD keys control forward, backward, left, and right motion, while the QE keys adjust upward and downward movement. The mouse cursor controls the camera’s orientation, allowing the user to look up, down, left, or right. The mouse scroll wheel adjusts the camera's movement speed for finer control. The camera is set to orbit the scene to ensure all objects remain visible from every angle.

I also included functionality to switch between perspective and orthographic views, allowing users to toggle between 2D and 3D representations of the scene with a single keystroke.

To ensure the program was modular and easy to maintain, I used custom functions to handle specific tasks. For example, I created a function to build the mug by combining two primitive shapes into one reusable object. Another custom function was used to handle camera movement, encapsulating all the logic related to translating and rotating the camera based on user input. Similarly, lighting and object placement were handled in their own functions. Each function was commented on, making it easier to understand the purpose of each block of code.