Shawn Frye

Computer Graphics & Visualization

Final Project

Southern New Hampshire University

Reflection

**9**. In developing the 3D scene, I selected a globe, pencil, pencil holder, and a stack of 3 post-it notes to create a cohesive and interactive environment. The globe was chosen as a centerpiece, symbolizing a sense of global connection and discovery, while the pencil, pencil holder, and post-it notes were chosen as essential tools for note-taking and brainstorming. To program the required functionality, I incorporated realistic sizes when compared to the photo and texturing for the objects, ensuring that users can interact with them in a natural and intuitive way. The GLUT library was utilized to efficiently create and render all objects, lighting, and other elements within the 3D scene. This allowed for seamless integration of complex geometry, textures, and shaders, while simplifying the handling of user input and window management.

**10**. In my 3D scene, users can effortlessly navigate through the virtual environment using a combination of keyboard and mouse inputs. I’ve designed the camera controls to be intuitive and user-friendly, allowing for seamless interaction with the scene. Using the 'W', 'A', 'S', and 'D' keys, users can move the camera along the X and Z axes, while the 'Q' and 'E' keys enable vertical movement along the Y-axis. This setup mimics familiar first-person shooter controls, providing an intuitive experience for users. Also, I have implemented mouse controls to enable panning around the scene. Users can click and drag the mouse to change the camera's pitch and yaw, offering smooth and precise control over the viewing angle, with adjustable mouse sensitivity using the '+' and '-' keys for a more personalized experience. The scroll wheel further enhances the user experience by allowing them to adjust the camera speed on the fly, ensuring a comfortable and customizable experience while navigating the 3D scene. By leveraging both keyboard and mouse inputs, our virtual camera system provides a robust and flexible solution for exploring and interacting with the 3D environment.

**11**. In my code, I have created several custom functions to make my code more modular and organized. These functions are designed to perform specific tasks and can be reused in different parts of my code. One of these functions is the "setupLights" function, which sets up the lighting in my scene by enabling lighting, specifying the position, ambient, diffuse, and specular properties of two lights, and enabling these lights. This function can be used whenever I need to set up lighting in my scene. Another function I have created is the "loadTexture" function, which loads a texture image and generates a texture ID that can be used to bind the texture to an object. This function can be used to load any texture image I want to use in my scene. Additionally, I have created functions to draw different objects in my scene, such as a globe and a pencil, which take in parameters like radius, height, and number of segments to draw the object with. These functions can be reused whenever I need to draw the same object but with different dimensions. Overall, these custom functions make my code more organized, easier to read and maintain, and allow me to easily reuse code in different parts of my program (MacDonald, 2020).

**References**

MacDonald, M. (2020, July 28). Modular Programming: Definitions, benefits, and predictions. Blueprint - Blog by Tiny. Retrieved April 15, 2023, from https://www.tiny.cloud/blog/modular-programming-principle/