April Nixon

Southern New Hampshire University

**7-1 Submit Your Project**

Professor Jeff Phillips

February 24, 2024



* **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.

A: Creating a 3D scene based on this image in Visual Studio entailed thoughtful selection of objects and consideration of the functionality necessary to bring the scene to life. The objects chosen for the 3D rotating cube included the desk, chair, lamp, and bookshelves with books, as these are central elements that define the workspace environment depicted in the image. These objects were selected not only for their visual impact but also for the atmosphere they contribute to the overall scene, representing a studious and intellectual space.

The lamp is a functional item that can be programmed to cast dynamic lighting within the scene, showcasing the lighting effects possible within a 3D environment. The books and bookshelves add depth and realism, and their textures contribute to the authenticity of the scene. The desk and chair are essential for grounding the user's perspective, providing a focal point around which the camera can navigate. Programming for functionality might include interactive elements such as the ability to 'turn on' the lamp or select books, which required event-driven programming techniques and an understanding of the user interface within a 3D space.

* **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.

A: For user navigation within the 3D scene, the virtual camera is pivotal. The camera setup involves programming the ability to rotate around the cube, zoom in and out, and pan across the scene. This was achieved using various input devices such as a mouse and keyboard. Mouse input can control the camera's angle of view, with left-click and drag to rotate, scroll wheel to zoom, and right-click and drag to pan. Keyboard controls might include arrow keys for rotation and specific keys for zooming and panning, allowing users with different preferences or disabilities to navigate the scene comfortably.

* **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?

A: Regarding modularity and organization of code, custom functions played an important role. For example, a function named ‘RotateCamera’ encapsulated the mathematics and logic required to rotate the camera around the scene based on user input. This function took parameters such as rotation speed and direction, and it updated the camera's position and orientation accordingly. By abstracting this functionality into a single function, the code remains clean, organized, and, more importantly, reusable. This means that the same function could be called whenever camera rotation is needed, possibly with different parameters, thus adhering to the DRY (Don't Repeat Yourself) principle in programming. Other custom functions could handle lighting changes, object selection, or user interface updates, each encapsulating a distinct feature or behavior that can be easily maintained or expanded upon as the project grows.

**References**

Cordeil, M., Hurter, C., Conversy, S., & Causse, M. (2013, September). Assessing and improving 3D rotation transition in dense visualizations. In *HCI 2013, 27th International British Computer Society Human Computer Interaction Conference: The Internet of Things*.

Schroeder, W. J., Martin, K. M., & Lorensen, W. E. (1996, October). The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *Proceedings of Seventh Annual IEEE Visualization'96* (pp. 93-100). IEEE.

Top of Form