CS 330 7-1 Submit Your Project

Larry E Williams Jr

Southern New Hampshire University

Prof. West

08/13/2023

In the creation of my 3D scene, my development choices were guided by the pursuit of an engaging and intuitive user experience. The selection of objects was a deliberate process aimed at both aesthetic appeal and practicality. By incorporating objects like pencils, cases, bottles, and pods, I sought to strike a balance between visual diversity and thematic cohesion. To realize the required functionality, I programmed a comprehensive navigation system. Users can traverse the scene using a combination of input devices—WASD keys for translation and the mouse cursor for orientation adjustments. This dynamic control scheme enables users to navigate seamlessly along the X, Y, and Z axes while also enabling precise adjustments to their viewpoint. I further implemented the ability to switch between perspective and orthographic views, enhancing user exploration while maintaining camera orientation.

Navigating my 3D scene is an intuitive experience that encourages exploration. Users can effortlessly control the virtual camera using different input devices. Utilizing the WASD keys, they can move forward, backward, left, and right, while the QE keys enable vertical motion, allowing them to traverse the scene's objects in three dimensions. This setup ensures complete coverage of the scene. Additionally, the mouse cursor enables users to modify the camera's orientation, enhancing the viewing experience. By scrolling the mouse, users can adjust the camera's movement speed, providing a customizable exploration pace. Switching between perspective and orthographic views is a simple key press away, seamlessly adjusting the visual presentation without disrupting the established camera orientation.

To enhance code organization and reusability, I developed custom functions within the program. These functions serve specific purposes, making the codebase more modular. For instance, the UProcessInput function encapsulates the handling of various input events, streamlining the process and enhancing readability. By centralizing input processing, this function can be reused throughout the codebase, simplifying maintenance and minimizing redundancy. Similarly, the UMousePositionCallback and UMouseScrollCallback functions isolate mouse-related interactions, making the code more focused and efficient. These custom functions contribute to a cleaner and more organized code structure, enhancing readability and easing future modifications.