Module 4 Assignment

Steve Hill

The artifact is a C++ application that serves as the entry point for a 3D graphics program. It initializes OpenGL through GLEW and GLFW, sets up shaders and rendering logic, and manages the scene and view for rendering 3D graphics. This code was originally created for the CS-330 course on Computational Graphics and Visualization as part of a final project developed on or after November 1st, 2023.

I selected this artifact for my ePortfolio because it represents a key milestone in my growth as a developer, specifically in graphics programming. It demonstrates my understanding of the OpenGL pipeline and how to structure a graphics application from initialization to rendering. By enhancing this project, I was able to apply principles of clean code, error handling, memory management, and debugging—skills that are essential in real-world software development.

The improvements I made to the code highlight several of my abilities. I implemented robust error handling for OpenGL and GLFW operations, which increases the application's stability and helps prevent crashes. I also modularized the code by separating logic into dedicated functions, improving reusability and readability. Logging was introduced to aid in debugging and performance tracking, and I ensured proper cleanup of dynamically allocated resources to avoid memory leaks.

The enhancements directly support the course outcomes I aimed to achieve in Module One. I focused on applying efficient code management, practicing error-handling techniques, and showcasing my proficiency in creating and refining a graphics-based application. At this point, I don’t see a need to revise my outcome-coverage plan, as this artifact meets all the original goals I set out to accomplish.

Reflecting on the enhancement process, I gained deeper insight into the importance of structure and maintainability in graphics code. I learned how even small improvements like logging and modularization can make debugging and future development much more manageable. One of the challenges I faced was ensuring that error handling was implemented effectively without overcomplicating the main logic. Another was verifying that resource cleanup occurred properly even in the event of early termination or failure, which required careful attention to control flow and pointer management. Overall, this was a valuable experience that reinforced good development practices and pushed me to think critically about how to make my code more professional and resilient.