Nar Bhattarai

Module 3

**Artifact Description**

This artifact is a 3D viewer application, built using OpenGL, as part of CS-330 (Computational Graphics and Visualization). The original code was created in [month/year]. It displays a scene composed of basic shapes, such as a plane, box, cylinder, and cone, all of which have textures. The application also allows the camera to move.

**Reason for Inclusion**

I chose this project because it shows good object-oriented design. It clearly separates scene management, view management, shader control, and mesh loading. This project highlights the use of graphics APIs, matrix math (GLM), and resource management.

**Enhancements Made**

For Milestone Two, I made several improvements:

* Robust Startup and Cleanup: I added checks for `glfwInit`, window creation, and `GLEW` initialization, and ensured that allocated objects are correctly cleaned up.
* Resource Management Fixes: I corrected the `DestroyGLTextures()` function to delete GPU textures correctly, which previously had a bug.
* Prevented Data Growth: I moved the material setup out of the render path to stop pushing materials every frame.
* Defensive Camera and Timing: I added checks in camera callbacks and limited `deltaTime` to avoid erratic camera movement during spikes.
* Improved Logging and Maintainability: I included logs for texture loading and added comments to clarify the function's intentions.

These improvements demonstrate my skills in software design and engineering, focusing on making the code robust, managing resources well, and ensuring maintainability. They align with program outcomes and technical communication, designing computing solutions, and using well-established tools and practices.

**Outcome Coverage**

The enhancements mainly support these outcomes:

Better documentation and logging.

Improved design choices, such as avoiding repeated allocations and reducing costly per-frame changes.

Proper use of APIs and effective resource management for shaders and textures.

**Reflection**

As I made these improvements, I learned how important it is to manage resources carefully in graphics applications and how minor bugs—like using the wrong GL function—can cause hard-to-find problems. I faced challenges in making sure my changes did not break compatibility with other helper classes (like ShaderManager and ShapeMeshes). To solve this, I made specific fixes while keeping the rest of the system intact to maintain build compatibility.