Steven Blathras

Software Engineering Narrative

**Artifact Description**  
The project I chose to enhance is a 3D scene rendering program written in C++ using OpenGL. I originally created this during my CS 330 course on Computational Graphics and Visualization. The main file I focused on is SceneManager.cpp, which handles the loading of textures, drawing of meshes, and setting of shader values for lighting and materials. It’s one of the key files that controls how the scene is built and displayed.

**Why I Selected This Artifact and What It Shows**  
I picked this project for my software design and engineering enhancement because it was a good example of real-world logic that could benefit from cleaner structure and better organization. There were several repetitive sections and hardcoded values that made the code less flexible. That gave me an opportunity to apply techniques I’ve learned about maintainability and modular design.

The biggest change I made was creating a helper function called LoadSceneTextures() to clean up how textures are loaded. Before, the code had a bunch of repeated lines for each texture, and now it uses a loop with a set of constants, which makes it easier to read and update. I also added a debug flag that lets you turn console logging on or off, which is useful when moving between development and production. On top of that, I went through and added clear, professional comments that explain what each part of the code is doing. These updates not only improve the structure but also make the code easier to understand for others who might look at it in the future.

**Outcome Alignment**  
This enhancement supports the outcome focused on using solid design techniques and tools to build solutions that have real value. My plan in Module One was to show that I could use clean, modular design, and I believe I followed through with that successfully. At this point, I don’t have any updates to the original outcome plan.

**Reflection on the Enhancement Process**  
While improving this project, I realized how much of a difference small design changes can make. Turning the repeated texture-loading code into a single function helped reduce clutter and made the program easier to maintain. I also appreciated the benefit of using constants instead of repeating strings throughout the file. It helped avoid mistakes and made things more scalable.

One of the only real challenges was making sure that the changes still worked with the existing OpenGL setup and didn’t break anything else. It took a bit of trial and error to get the order and structure right, but I was able to figure it out without too much trouble. Overall, this process gave me a better appreciation for clean, scalable code and showed me how valuable thoughtful design can be, even in smaller enhancements.