

***A Frenetic First-Person Shooter (FFPS)***

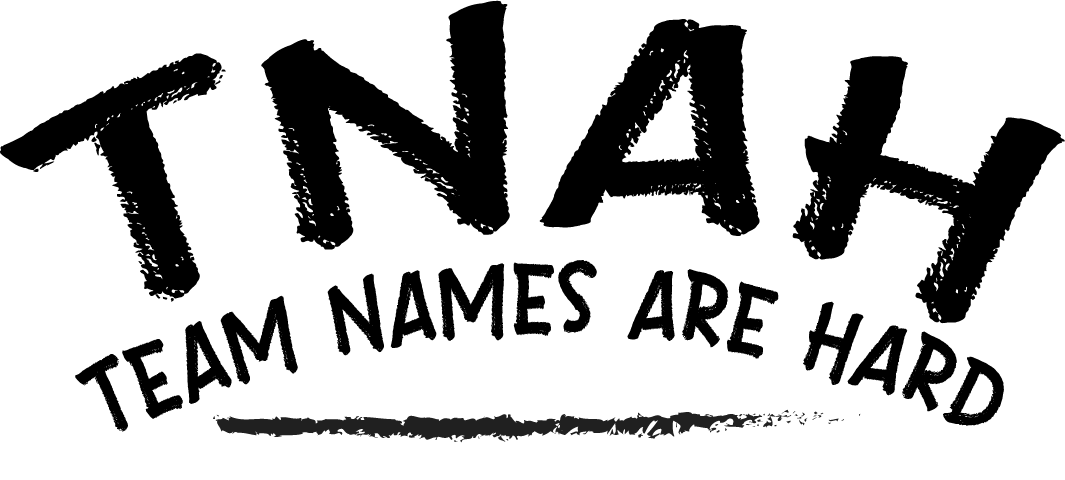
**Discussion of issues Encountered**

Version 1.0

Produced by Dylan Blereau, Bryce Standley, and Christopher Logan

Created: March 31, 2021

Updated: April 7, 2021



# Introduction

While we would love to say that we as a group had a stress-free time with this assignment, that is simply not the case.

Developing an engine with the Model-View-Controller pattern and keeping software reuse at the forefronts of our mind wasn’t easy, but we feel that the engine and resultant tech demo game that was made from it is proof that we overcame most of our issues. Despite being able to overcome our issues, we still had plenty of frustrating problems throughout development.

This is a comprehensive list of the biggest issues encountered while developing our game engine for assignment 1, as well as what did to overcome them.

# Issues Encountered

## Amount of work required

I think it is fair to say that the sheer workload of this unit took us all by surprise. Dylan having experienced the first few weeks of the unit last year allowed us to perhaps be a bit more aware of how quickly things ramp up in the unit than other groups, but it still shocked us just how many hours need to go into this unit.

The solution to the issue of the overwhelming workload was to dedicate time each day to look at things and work on small aspects if possible. While each day contributions may be minor, keeping invested in the project each and every day made it easier to comprehend and deal with the workload. Another solution was that towards the end of the assignment we all had a greater understanding of the engine and how to develop for it, which in turn made aspects of development that took days initially only take a few hours at most in the end. In the end, we feel like we have overcome our struggles with the workload and feel better prepared for the second assignment in that regard.

## Lack of confidence in our MVC design

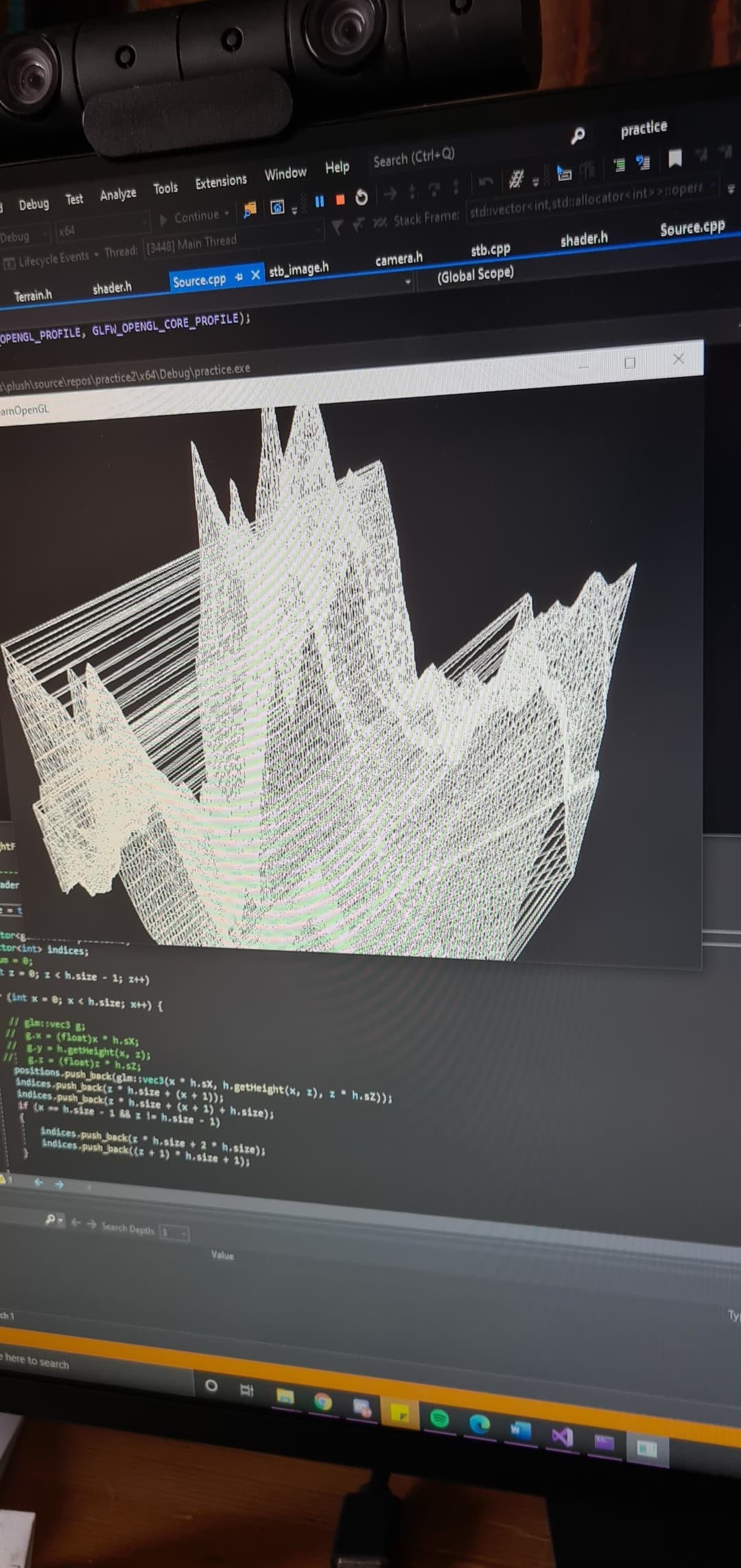
Initially when beginning to plan out how our engine would be designed, we all had rough ideas, but none of us were confident that our design was good enough. Chris, who was the team member assigned with leading the development of the engine took time to research the MVC pattern, coming back to the group with a plan that made us more confident in the architectural ideas we’d established. While the lack of confidence in our engine design was an issue to begin with, Chris’ commitment to researching and recognizing the design pattern we needed to follow to succeed ultimately fixed the problem the team was having.

## Unsure of how to best implement lua scripting

Each member of the team scored highly in the Lua scripting lab, but despite the basic understanding of what lua does and how it works, none of us were too sure on how to best implement the lua scripting into our engine. We weren’t sure where in the engine it should go and we also weren’t sure to use luabind or luabridge. The solution to this problem was Dylan’s luaManager class, a powerful class that uses the singleton design pattern and provides instances of lua and luabridge (which we determined was easier to use than luabind) to any class that wants to use it. The luaManager handles scripts for the terrain, skybox, scene and more. The luaManager class was the team’s solution at it finally made sense how to best make use of lua scripting in our engine.

## Terrain drawing incorrectly

When it came to the programming side of the assignment, the first issue we had to deal with was the fact that our terrain class wasn’t building terrain as intended. The issue we had with the terrain is that we were trying to develop it for use in modern opengl, but we were using the code from the lecture notes that was legacy opengl code. The solution was ultimately to go through each function, determine how they work and if they will work with modern opengl, and redesign the functions accordingly. Because modern opengl was new to three of us it took a while to get off the ground, but after a while we were able to get the terrain to draw, albeit incorrectly.



As you can in the image above, the tip of some points would draw to the other side of the terrain, which was an issue none of us could understand. The fix was to make use of glPrimitiveRestartIndex, which in turn removed the unnecessary additions to the terrain.

Other aspects of the terrain such as the multitexturing and lighting gave us headaches, but as a team we managed to work through them.

## Assimp not drawing models correctly

We had numerous issues getting the Assimp model loader to work throughout the project. One of the first issues we had was textures not drawing properly on some models. This was quickly fixed by setting the stbi\_set\_flip\_vertically\_on\_load flag to false. While this fixed the issues with some textures, some models with textures on them still wouldn’t draw with textures attached. We soon realized that the design we had for our assimp class didn’t allow for fbx models with embedded textures.

Because we needed to move on with the assignment, we decided to provide our fbx models with a separate texture file, which is then placed onto the model. In the end, we managed to get Assimp drawing models are we intended.

## Random exceptions being thrown on different computers

One frustrating occurrence that caught the group off guard was the fact that an access violation reading location error would occur on one device, despite the code functioning perfectly fine on another. While the code ran perfectly on Dylan’s PC, Chris and Bryce were unable to run the same code on their respective computers. We managed to overcome this issue by making Bryce and Chris update the drivers on their AMD graphics cards, which in turn fixed the issue. It turns out that AMD graphics cards are far more finicky when it comes to passing data to the gpu in comparison to NVIDIA cards.

# Conclusion

Despite running into a few issues throughout the development of our engine thus far, we as a group arguably haven’t encountered as many issues as we thought we might. When we did run into an issue, we worked together as a group to slowly but surely find a resolution to the problem.

While we are guaranteed to have some issues in the second assignment, the way in which the group handled all the problems we faced has us confident that we will succeed.