Table 1: Revision History

Date	**Developer(s)**	**Change**
December 18, 2019	Sasha Soraine	Original

Reflection Report on Lights, Camera, Models!

Sasha Soraine

1 Project Overview

The original intent of this project was to understand lighting models well enough to be able to generate them on the fly when the user gives a description of a scene and a named lighting model/shader set. The requirements for this were to have the system act as a plugin for the Unity editor and parse Unity scenes as JSON files to manipulate and render.

It quickly became apparent in the second half of the semester that this wasn't going to happen the way I originally envisioned it. I settled on creating a Unity executable with easy to read Shader/Lighting files. While the original intention was to create something that showed off how well I knew lighting models, I think that this approach helped to solidify my knowledge of implementation elements in a way I wasn't expecting.

2 Key Accomplishments

Overall I'm quite proud of the coding that I've done. I'm not a strong coder and took on a project that I wanted to learn how to do (coding custom Shaders and Lighting Models) even though I was hesitant about my ability to learn both the details of the theory behind it and the new languages for implementation. While what I developed is exceptionally simple, it took me a long time and a lot of reading, Googling, and Youtube watching to figure out how to tackle the problem. A big stumbling block was trying to figure out how to turn off Unity's built-in light system so I could create my own. It took me 2 whole days of reading to realise that in the context of Unity Shader files are both the lighting model and shader code. That revelation was the only way I was able to create a working part of this project. I'm glad that I have something functional that can be played around with on the web, and I think that if I want to expand this work I could very easily create new components for it now that I know what I'm doing.

3 Key Problem Areas

Generally the thing that I think went wrong was trying to learn the topic as I went. I was unfamiliar with writing my own Shaders and Lighting Models, as

well as with the details of the math for the topic. As such I flip-flopped between implementation details like what to implement it in (OpenGL or Unity), what it should be (a library or an executable), and how best to present it. I think that I spent a lot of time thinking about details and having deadlines slip away from me as I was caught trying to understand things. While I did suffer from issues of time management and coding problems, they would have been more manageable if I had a better grasp on my project and its vision from the get go.

4 What Would you Do Differently Next Time

I think next time I would stick with an SRS from the start. Scoping instantly to one executable would eliminate a lot of the variables that I spent time thinking about. I also think I would pick something that I understood better; I spent a lot of time wrapping my head around Meshes, NormalMaps, and figuring out the way things are currently done that I got lost trying to design my own. I think a simpler project where I could have done more on the documentation because I wasn't worried about the implementation would have been a better idea.