**Final Project Report**

By Zhiyuan Liu

In the previous Final Project Milestone Report, I mentioned that the plan for me is to implement a technique regarding rasterization called shadow mapping using WebGL. In the original plan, I wanted to show shadows of objects in a scene as well as showing the light source, such that the person who sees the project will see the shadow casted by the objects.

Previously, using the models and resources already provided in pa2, I was able to construct the scene with a teapot or with a bunny. In addition, I successfully changed the position of the light source to higher than the objects in order to showcase the effect of shadow mapping more obviously. However, I encountered some large problems that forced me to change to another topic. As I was implementing shadow mapping, I noticed that the codes in the provided pa2 was extremely complex with all the UIs and different shaders in the html file. I was not familiar with most of the codes as pa2 only focused on functions regarding shaders. In addition, most UI part of pa2 was not related to what I planned to be doing in this project, but each part of the codes is connected, and it was extremely difficult for me to change anything substantial without breaking the whole scene completely. It was also very hard for me to debug when something went wrong. In the end, I couldn’t figure out a way to use pa2 models and codes as a template to complete the final project, and I did not have time to learn WebGL from scratch well enough to understand codes in pa2 and use the codes as a base for my final project. As the deadline approached, I decided to change the topic to a simpler one, fog shader, and learn the basis of WebGL just well enough to implement the algorithm for this shader.

In order to learn WebGL, I found this site <https://sites.google.com/site/csc8820/educational/read-webgl-programs>.

This site has the tools and code base I need to build a scene from scratch, and from that, I will be implementing fog shader algorithm. The plan for me is to construct the scene using the code examples in this website tutorial. The object in this scene is very simple and it would be consisting of a large cube in the center of the scene. To show the effect of my fog shader to the person who sees the scene, the person will be able to press the keys “W” or “S” to either increase or decrease the distance of fog.

In order to complete the goal, I will be using a few resources involving WebGL. They will be “webgl\_utils”, “webgl\_debug”, “cuon\_utils”, and “cuon\_matrix”. The content in my “Final.html” file is simple and is based on the example from the previous website. I added two lines of instructions in the html file for the user to see. In the Fog.js file, I constructed lines to recognize keys that are pressed. As for the cubic object, I just used the coordinates from another example in this website, <https://sites.google.com/site/csc8820/educational/move-a-camera>. The main part of my implementation happens on the top of the Fog.js file, the two variables that are the most important are VSHADE\_SOURCE and FSHADE\_SOURCE. They stand for vertex shader and fragment shader, and they function as small programs and will be used in later parts. This method of constructing both kinds of shaders is also from the tutorial. As for the fog shader, there are several important aspects about the fog. The color of the fog and distance of fog. When the distance is the greatest, the cubic object is completely shown, and when the distance is the smallest, the cube completely vanishes. When the distance is set to between 40 and 90, and this range is good enough for the demonstration. In the FSHADE\_SOURCE variable, important aspects of the fog will be implemented. One important variable is the factor of fog, called fogFactor, and this variable will be larger as it becomes closer to the viewpoint. In the main function, the shaders are initialized, and I could change the variables according to the plan there.

For this project, although there are not a lot of techniques used, and they are relatively simple as well, I did learn very much about the basis of WebGL, and how to construct scenes and objects from scratch without the provided codes from pa2. I also learned the relations between each part of the codes and the relations among different files in a WebGL project.