**Module 8 Critical Thinking**

Alex Stampfl

Colorado State University Global

CSC405 Graphics

Bingdong Li

January 9th, 2025

**Lessons Learned**

This semester I was introduced to the subject of graphics and virtualization. I learned about OpenGL and its API, WebGL, which is written in JavaScript and runs on the GPU and can be displayed on the browser. Upon learning the basics of WebGL, I was introduced to concepts pertaining to the behavior of light, Cathode Ray Tube (CRT), orthographic projection and projectile projection. I discovered that there is a lot of setup code involved in WebGL, because it works at a very low level, where we have control over the GPU, and all its processes. This is just a few of the things I have learned, and I will discuss what I have learned in further detail.

WebGL is an API that involves writing code to render geometric shapes onto the canvas, which is a tag used in HTML5. WebGL does not magically produce cool and interesting shapes on its own. It is up to the developers to write the code specifying every geometric shape, vertex and every line. The setup code is pretty tricky and takes time to get used to as you are writing directly to the GPU, which in turn, renders the geometric shapes onto the canvas. WebGL contains two shaders, which specify how the vertices should be rendered onto the screen. There is the vertex shader, which determines where the vertices will be drawn, and the fragment shader, which specifies the color of every pixel on the viewport.

The buffer is a type of tool used to store and hold data pertaining to the shaders, vertices and GPU. There are various types of buffers such as the frame buffer, the color buffer, and the depth buffer. The buffer is the part of the code which specifies the vertices of the triangles, which are one of the fundamental geometric shapes in WebGL. The buffer is a part of the initialization code or the setup code that must be established before any drawing or rendering of geometric shapes can be created. The frame buffer is the last stop before the geometric shapes can be rendered onto the screen.

Another concept we learned about was that of viewing the object. We, the users or developers, are the source of the viewing or where the camera is. In WebGL, when we rotate an object, the camera is actually the one moving around the object, while the object remains still. It just appears that the object is the one moving around the object. When creating a 2D object, the 2D object is actually part of the third dimension. All that is required to transform a 2D object to appear as a 3D object is a bit of tweaking of the viewing camera and of the object.

Lighting and shading were also fundamental concepts I learned this semester. Objects appear very bland and 2D without any lighting or shading. 2D objects without light are merely a solid color with no changes. Lighting and shading have the power to transform an object into something much more dynamic and can make it appear in 3D. When you combine the lighting and shading with translation and moving the object in space, it makes the object become much more interesting.

I stated before that triangles were one of the most fundamental shapes in WebGL. At the start of the semester, one of the projects I did was creating the Sierpinski’s gasket, an apparatus that consisted of triangles being created recursively into infinity. It is a large triangle which contains repeatedly smaller and smaller triangles. Another project I did was creating an orb or sphere which was shaped from triangles, and the more triangles you added, the smoother the sphere became.

In conclusion, this semester taught me a plethora of interesting concepts related to graphics using WebGL. Lighting, shading, geometric shapes are all concepts that I believe will carry with me and help me to contribute to building all kinds of graphics and animations for images, video games, and virtual reality. There are also JavaScript APIs such as three.js which abstract away much of the WebGL complexity and allow you to build some really cool stuff. I definitely plan on exploring that as I would love to create some interesting graphics and games.