CSC 405-2

Module 2

Critical Thinking Reflection

Nastacio M Cabral – Tafoya

04/27/2025

Prior to WebGL, the Internet used Java Applets, Adobe Flash Player, and SVG to embed graphics on web pages. Each platform made web pages more interactive or improved graphics beyond what HTML, CSS, and JavaScript could do alone. When browsers began to end support for Java applets in 2015, and Adobe Flash support ended in 2020, I was introduced to WebGL as an HTML5 alternative that utilizes the <canvas> element to create embedded graphics. I was overwhelmed by the possibilities when I found that WebGL was an API that allows JavaScript to interact directly with graphics hardware like OpenGL. Because of life, however, I never got the chance to set aside time to learn WebGL before this class. I am excited to learn as much as I can about WebGL because of the possibilities that it opens up for web developers.

In this module, I was initially intimidated by the idea of having to draw a Sierpinski Gasket using WebGL as the first assignment. I am not at all familiar with how WebGL or OpenGL works. I found an example of drawing a triangle on Tutorials Point. From there, I found an example of drawing points using WebGL. At that point, I was able to begin experimenting with changing the vertices and shader values to change the color of the background and the color of the shape being drawn. I was already familiar with the process of drawing a Sierpinski Gasket because I had completed an assignment in my introduction to programming class at Morgan Community College that utilized Python Turtle to draw the Sierpinski Gasket. Fortunately, I am also already familiar with HTML, CSS, and JavaScript, so I was able to focus on experimenting with WebGL concepts.

Once I figured out how to draw as many points as I wanted, it was as simple as generating the vertices with a JavaScript function and passing them to the gl buffers and shaders. I still do not understand what most of the gl.bufferData and gl.bindBuffer functions are doing. I could not figure out how to draw a triangle and the individual points in the same canvas element in time to turn in the assignment. I compromised by defining two separate canvas elements, drawing a black triangle for the triangle-shaped background, and setting the opacity for the canvas background to 0.0 so that only the black triangle would show. I then overlaid the second canvas element over the triangle and gave it a background opacity of 0.0 so the black triangle could be seen. I then generated the vertices for the individual points that would draw the Sierpinski Gasket, utilizing green dots one pixel at a time. I look forward to learning more about WebGL and what is happening between defining the vertices and getting them to appear on the screen.

References

Geeks for Geeks (2025), Java Applet Basics, Retrieved on 04/27/2025 from,

<https://www.geeksforgeeks.org/java-applet-basics/>

Adobe (2020), Adobe Flash Player End of Life, Retrieved on 04/27/2025 from,

<https://www.adobe.com/products/flashplayer/end-of-life-alternative.html>

MDN (2025), SVG: Scalable Vector Graphics, Retrieved on 04/27/2025 from,

<https://developer.mozilla.org/en-US/docs/Web/SVG>