Name: Braden Maillet
Email: braden_maillet@student.uml.edu
Professor: Krishnan Seetharaman
Class: Computer Graphics (COMP.4270)

The attached source code includes a .html as well as a .js file. Together they compile as a small web application that demonstrates some of the capabilities of Webgl. The assignment required me to take starter-code of a sierpinski gasket provided by Professor Seetharaman and alter it so that it accomplished the following. In a continuous loop the program was to change the color, size and number of points within the gasket.

I believe that the source code I provided does exactly that. Not to say that it wasn't a bit challenging getting my head wrapped around Webgl. I had a lot of issues with every step of the process. I started by looking into how to change color within Webgl. It was obvious that it was related to the fragment shader. Which I later learned was responsible for determining the color of each pixel. Furthermore it all came together in terms of color when I discovered uniform variables in glsl. Uniform variables allow you to change attributes within your javascript source code of things like color. The next biggest hurdle was figuring out how to animate the gasket. The professor did give a hint towards this. That hint being to look at a future assignment's source-code which involves animation. This revealed the recursive technique of calling "requestAnimationFrame(render())" within the render loop. This function tells the browser that you would like to perform an animation. In this case repeatedly calling the render function. Within the render loop I have a series of Webgl and user supplied functions that change the data within the buffer as well as some uniform attributes relating to scale and color. Figuring out how to scale the gasket came easier due to the knowledge that I had gained trying to figure out the other aspects of the assignment. I assigned a uniform function within the vertex shader that scales the gl_position variable. Additionally I have a function that varies the scale continuously between a constant maximum and minimum value. Changing the number of points displayed on the gasket was fairly easy as well.

Are there any remaining bugs? The way the program currently works is that it resets the gasket with every iteration. It wasn't specified in the assignment whether or not it was required to work on the same base gasket with every iteration. It's something that I could implement if required. Also the program does run in a continuous infinite loop. It does not only run 10 times. I found this to be a more entertaining visual. Whether or not this is considered a bug or extra credit I will leave for the grader to determine.