

Graphics

Kynan Justis

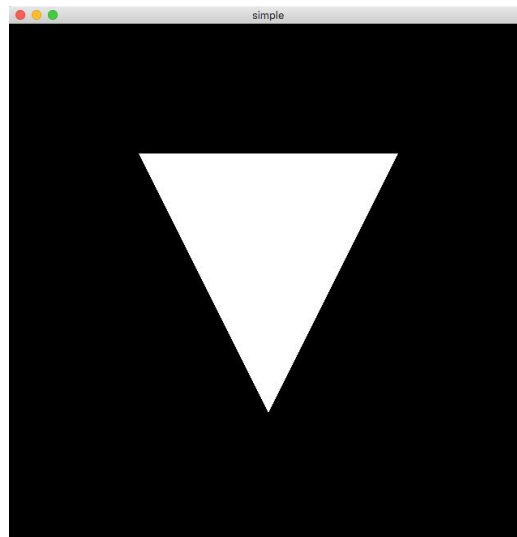
Assignment 1

02/12/2018

Overview

There were three main parts for this programming assignment. In Part A, the task was to merely download the sample code, compile, and run it in our own development environment. Using OpenGL, GLEW, and GLFW, a simple OpenGL context was created that displays an upside down white triangle. In Part B, the task was similar in that the sample code just had to be built and ran. However, instead of a triangle, the display would show a Sierpinski Gasket. Finally in Part C, the task was to change the code in Part A to display a square instead of a triangle while also changing its overall color.

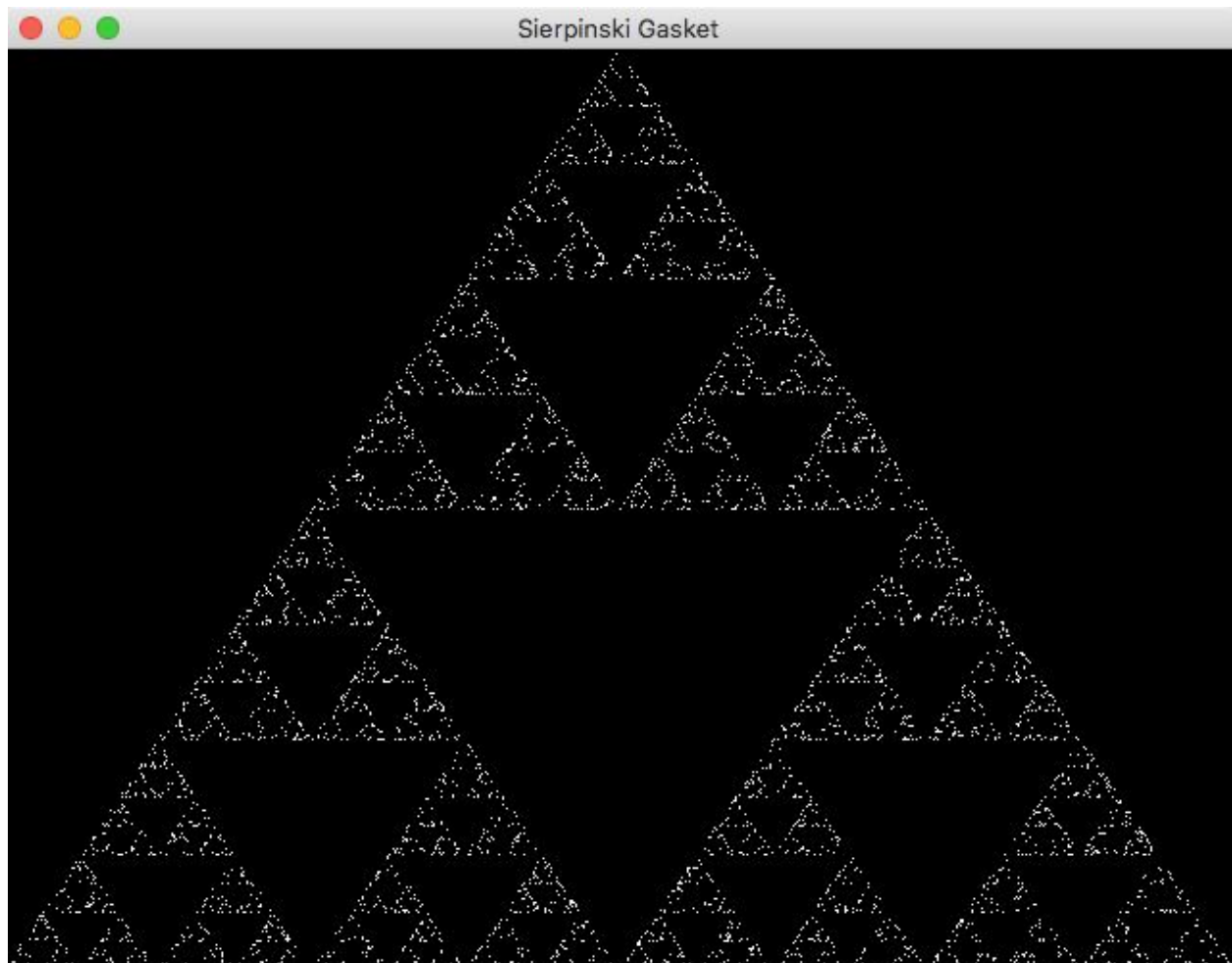
Part A



Here is the resulting display from compiling and running Part A. The basic triangle shape is shown upside down and centered with an overall white color. The most challenging part was setting up the development environment so everything would compile. Since I use a Mac, most of it turned out to be easier than expected however. All of the libraries were installed using Homebrew, which allowed for CMake to find them with a simple function call. Also. the section

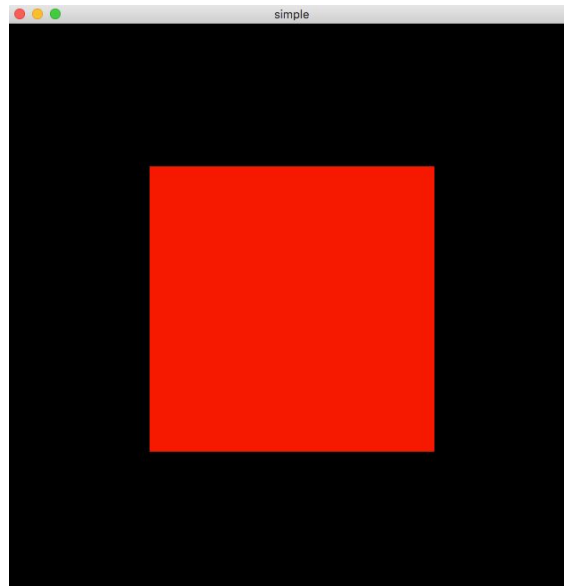
of code with the preprocessor “`#ifdef`” turned out to be necessary since Apple does not seem to default to the latest supported version of OpenGL.

Part B



Given that this part was mostly the same as Part A, there wasn't any new obstacles in terms of compiling and running. The same build setup was used, just separated into a different project directory to keep all of the folders clean. Again, it was necessary to keep the preprocessor directives uncommented since this code was running on my MacBook. Other than that, everything was pretty much the same.

Part C



Compared to the other two parts, I would say this section was the most challenging. That being said, there weren't many changes that needed to be made to the original code for everything to work. First, the point array in the *init()* method needed to be updated so that six points were defined instead of three. This is because the square itself is actually made up of two different triangles. After that, the *vec4* value in the fragment shader had to be updated so that the final color would be something other than white. I changed this one to red, but the color version of this image is only available in the GitHub repo. Finally, the last step was to modify the *glDrawArrays()* function in *main()* to draw the six points defined in the points array. Once all of this was done, everything rendered as expected.

Note

The final version of the code, the full-color result images, and the CMake build instructions can be found online at <https://github.com/kfjustis/graphics-1>.