Graphics

Kynan Justis

Assignment 2b

03/12/2018

Overview

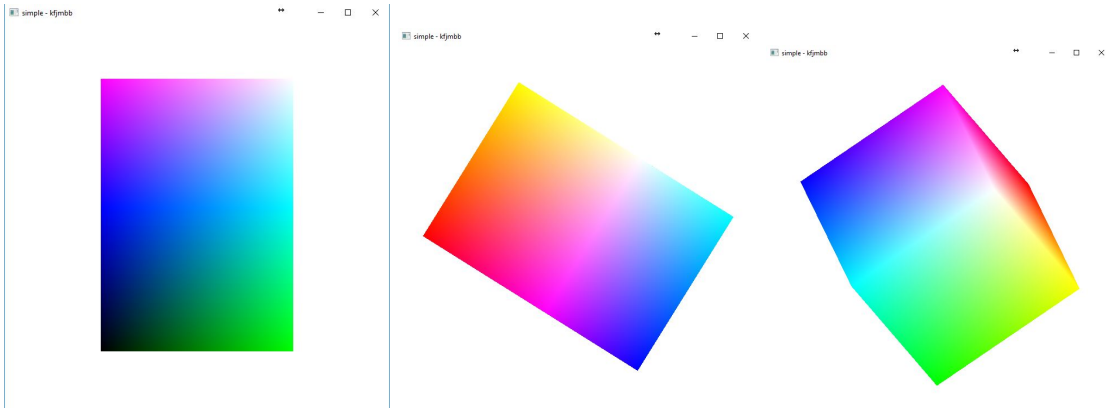
Again in this assignment, there were two main programming parts. In Part A, the task was to compile a similar sample program to one found in the first assignment. This sample program contains code that generates the Sierpinksi Gasket. However, the challenge this time was to modify the program to support placing multiple gaskets around the screen using the mouse pointer's location as a rendering input. With the new program, the user could place several gaskets around the screen in the same positions that the mouse clicked. In Part B, the task to was compile a different sample program that renders and rotates a three dimensional cube. However, instead of adding a mouse callback, a keyboard callback needed to be added to control the cube's rotation and animation.

Part A

As mentioned previously, Part A involved running the Sierpinksi Gasket program. Instead of just compiling the program, however, the requirement was to modify it so that the gaskets could be placed based on user input from the mouse. Using a mouse callback, the position of the mouse after a mouse click is recorded. Since these coordinates are retrieved from screen space, they have to be converted to world coordinates before the gaskets can be rendered

in the window. Once the coordinates are converted, the VBO is updated and finally rendered in the main loop. That being said, gaskets can only be rendered as many times as there are points in the overall point array. To save memory, a limit of 10 gaskets was enforced in this program.

Part B



In part B, the task was to compile the sample program that renders and animates this three dimensional cube. Already, mouse callbacks were implementation that allows the user to change the rotation of the cube during run-time. However, the challenge for this part was to implement the same functionality using keyboard callbacks for the X, Y, Z, and S keys. X, Y, and Z would cause the cube to change its rotation to the given axis whereas the S key would stop the rotation all together before allowing the user to resume the animation again. Since the logic was mostly the same between callbacks, all of the changes were pretty straight forward.

Note

The final version of the code, the full-color result images, and the Visual Studio build instructions can be found online at https://github.com/kfjustis/graphics-2b.