Graphics

Kynan Justis

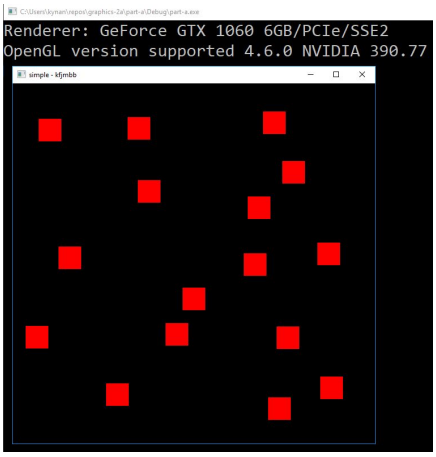
Assignment 2a

02/26/2018

Overview

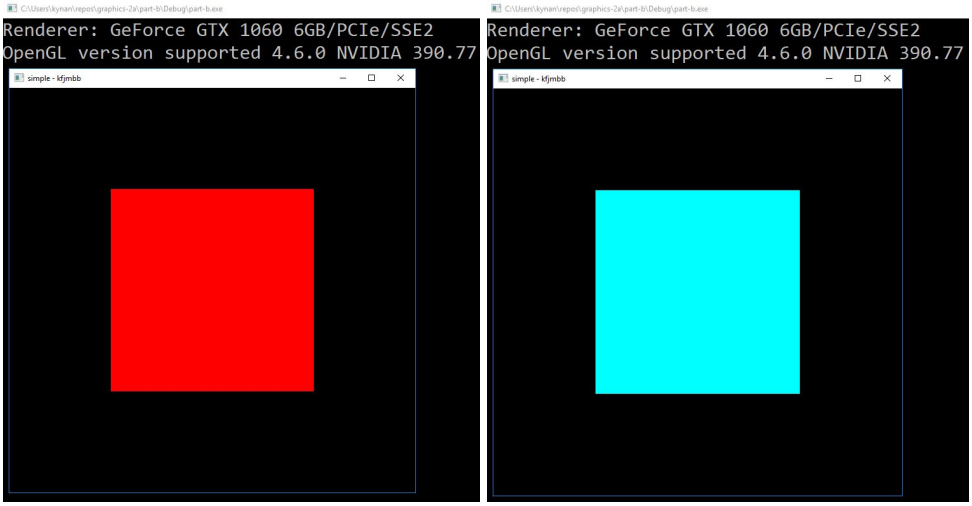
There were two main parts for this programming assignment. In Part A, the task was to compile another sample program. This sample program contains code for rendering multiple squares on the screen. However, instead of drawing all of the squares at once, they can be placed as the program is running through the use of a callback function. To help with this process, additional header files had to be included in the project solution. In Part B, the task also involved compiling and running a program, but this time to produce a cube rendered in three dimensions. Then, two callbacks had to be implemented. One callback would handle a right mouse button input and terminate the program. The other callback would handle any input from the keyboard to cause the cube's color to change.

Part A



As previously mentioned, Part A just involved compiling and running another sample program. As you can see in the result image, many squares are drawn on the screen at one time. Through the use of a callback function, each square is placed after a left mouse button press at the current location of the mouse pointer. Since the shader has the color value hard coded in, all of the squares remain a simple shade of red.

Part B



In Part B, most of the code was again provided. Since compiling and running was very similar to the first part, no major obstacles were encountered here. The tricky part was getting access to the cube's color value during runtime. To accomplish this, a global variable was added to hold the id of a uniform variable from the shader. Using *glfw*, this id could be retrieved during runtime and then used to bind new color data to the shader. Once this framework was established, all that had to be done was to write a callback function for handling the keyboard input. Then, when a key is pressed, a random RGB value is generated and fed into a new *vec4* value before finally being passed to the shader. This allows the cube's color to be changed for each key press during runtime.

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-2a.