Graphics Programming Project 1 Report

Luke Miller
010785453

Problem Statement:

The goal of the programming assignment is to generate the seven pieces of the game Tetris and display them on screen to be regenerated when the screen resizes. This is to give us a first look into graphics programming and the OpenGL API for the C language. The inputs of the program are the user changing the size of the window and the program should redraw the pieces in different places on screen. There isn't any error handling because there shouldn't be any errors due to the fact that the user's input is limited to resizing the window.

Design:

When I designed my program I first tried and hardcoding each shape and as soon as I complete the first O shape, I realized this way wasn't going to cut it. So I decided to create a function called drawSquare(). This function takes in the x and y coordinates for the exact middle of the square making it easy to scale the length of the sides for both the outer polygon and the inner border. Then from there I can have 7 functions that draw 4 different drawSquare()'s relative to its midpoint. This made it extremely easy to then just call all 7 functions in the display callback feeding each two random positions on the screen.

Implementation:

To get started I simply tried to run two or three of the provided source code example files just to make sure my computer was good to go. Then I looked at the simple square.cpp file and went from there in how OpenGL draws its graphical images. Then I copied the default code from this file and used the polygon in it as a starting block for drawing my first square. After that I decided to start over in my overall implementation because hardcoding the O piece was really tedious and I knew there was a better way. So, after I wrote and function for drawing squares it took me about 1.5 more hours. In total it took approximately 3 hours to complete due to a bug I was having where my random number generator wasn't working quick enough, and the pieces were all sticking together in the same spot.

Testing:

After each piece was programmed, I checked to make sure it was drawn correctly. Then after each was implemented, I implemented the random coordinate generator feeding it random seed based on time and this did not work well because it was not quick enough and therefore all 7 shapes were being fed the exact same coordinates and globing together. After I commented out the seed it worked like a charm even when resizing the window. This program is pretty simple so there weren't really any special cases to test for and that leads me to believe everything seems to work as expected.

Conclusions:

In conclusion, this program was a success. Everything works as expected. I am very proud of the modularity of the program and I believe this was a great start for my first assignment in the class. Knowing what I know now I wouldn't have wasted time trying to hard code each piece. If it weren’t for that the assignment would have taken me less than 2 hours, especially if I had known about the seed problem ahead of time. Three hours with debugging isn’t wholly that bad and I am proud of how it turned out.