Project Report - Project 2

Alexander Ahlbrandt

010726372

Problem Statement

The goal of this programming assignment is to use OpenGL libraries to make rotations and capture input from the user, and create a program that allows a user to add and remove cubes from virtual 3D space. Then inputs to the program come from the keyboard, a user will press different keys to interact with the program. Output of the program will come in the form of a graphical window, showing what the user has made with their combination of keystrokes. There was no required error handling.

Design

I designed the program execute in the following steps:

1. Listen for keyboard inputs.
2. Provide feedback via console if necessary.
2. Calculate the next state of the blocks.
3. Calculate the next rotations to make.
4. Graphically display the new state to the user.

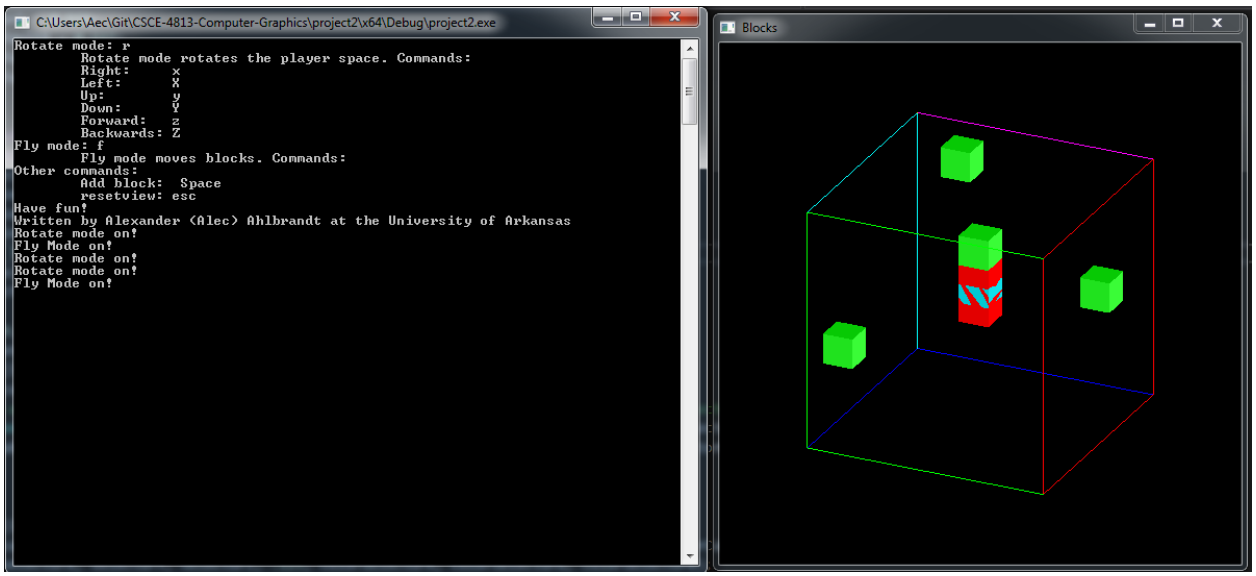
Most of this logic resides in the "keyPressed" method. It will update variables in the program to be referenced by the display method. Some data structures that I used to help me do this are: Coordinates. A coordinate is a simple data structure that holds three floats - x, y, and z. A vector of coordinates is assigned to a variable, which holds the coordinates of our cubes. Some other data structures I have are ints, which can hold the number of degrees to rotate the game space. I used one algorithm to find a cube to delete. It's a simple loop over the cubes to find a cube within a certain coordinate range to be deleted. I check if the coordinate is in the range of tolerance so the selector cube doesn't have to be exactly on the cube you want to delete, because sometimes the cubes would line up but the cube wouldn't be deleted. Once the cube's found, it's deleted and the loop exits. A pro to this choice is that you don't have to be directly on the cube, eliminating the chance for error. A con to this is that you may delete more than one blocks on accident if you find yourself in the range of tolerance for more than one cube, but it's better than not being able to delete it at all.

Implementation

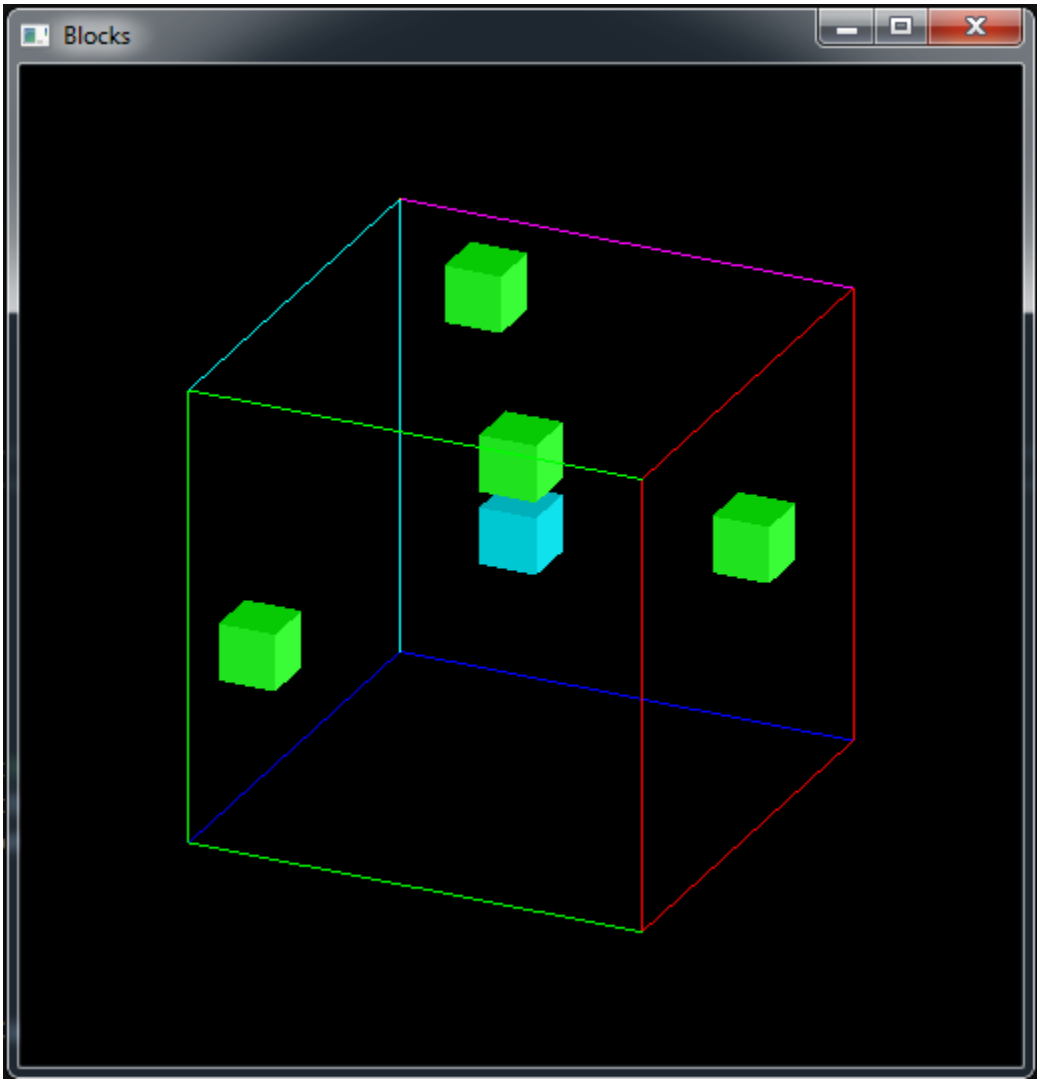
To write this program, I used C++ and OpenGL. I used sample code from the course website to learn about rotations (office3.cpp and cube.cpp). I used the cube method from cube.cpp to make a cube and show a boundary of the player space. I also used it to make my selector cube and the player cubes. I modified it to not only support custom colors, but I subtracted and added a fixed amount to each color to make it lighter and darker for each face. This way, the user can distinguish each face, and therefore, the edges and shape of the cube. My development timeline was five days, but of course, not all of this time was spend programming!

Testing

There was not much testing to do for this project, but I did ensure that the program was able to endure a string of random input combinations. I rotated in different directions, added cubes, removed cubes, and moved the selector in many directions many times, and never had an issue with the objects going invisible, or losing any functionality for that matter. Here are some sample inputs/outputs for the program.



Cubes added, rotated among the x and y axis. Ready to delete at this position.



Screenshot after deletion.

Conclusion

The overall result of this project is a success. I have used C++ and OpenGL to make a block game that uses rotations and captures user input to render graphics in 3D space.
In the future, as always, I plan to start earlier, so I can implement more cool features into my program! To complete this project I took about five days, over thirty commits, over 26,733 additions, and over 803 deletions. I enjoyed and learned a lot from this project, and look forward to the next one and learning more OpenGL!