**Graphics Programming Project 5 Report**

Luke Miller

010785453

**Problem Statement:**

Graphical user interface

Description automatically generated with medium confidenceThe goal of the programming assignment is to use OpenGL texture maps to interactively translate, rotate and display 3D Tetris pieces made up of four cubes each. This is to give us a first look into the 3D texture mapping. The inputs of the program are the user pressing specific keys on the keyboard and the program should draw, translate, rotate, and move the specified Tetris piece in the place/position the user desires in a 3D space. There are few places of error handling that keep the pieces on the board and don’t accept non desired input from the user. For example, if the user presses the ‘u’ key the program does nothing because that is not a specified key for the program and if the user is on an edge and does an action like move try to move a piece past the edge then the program will test and make sure before moving said piece to make sure it won’t move off the board, and if it would move off the board the program won’t accept the input.

**Design:**

When I designed my program, I first decided to tune up my second homework and almost fuse it with the texture mapping parts of textures3, the provided code that uses texture mapping. In tuning up my second homework, I converted it to allow for a third Z dimension. I also decided to create a data structure to keep coordinates for xyz placement as well as color for a cube. Then I used a vector to store coordinates and colors for each block that has been locked into place and the four cubes that are currently being placed by the user. Then to display the pieces the program just needs to loop through both vectors and display their cubes. I also added functions to rotate the pieces in all three dimensions. Allowing the piece to be placed outside the screen. Finally, I angled the board so you can see the pieces are 3D. I added a function that displays a 3D border around the board, but it tented the textures the color of the border, so I ended up commenting out the function call (this is why the pieces are discolored in the photo). This design suited the projected even better than before and allowed for an easier implantation than the other projects we’ve completed.

**Implementation:**

A picture containing scatter chart

Description automatically generated To get started I tuned up my second Tetris program with some design element changes from the second solution to get it up to snuff. Then I copied the some of the default code from one of the ‘texture3.cpp’ files for the texture mapping I needed. After that I decided to pretty much get at it with how I implement rest trying to just get it to redraw a vector data set so I knew it would work when redrawing old shapes. The hardest part was trying to figure out the 3D rotation stuff which I saved for last. I ended up coding my own functions for the rotation based off the Transformation slides provided, using cosine and sine. The first parts took maybe two hours then the texture mapping stuff was a little harder for me to understand so maybe had me finishing out in less than 3 hours total.

**Testing:**

After each transition was programmed, I checked to make sure it was moving the pieces correctly. Then after each piece was implemented, I retuned how each function drew its shape to allow for the dimensional blocks and textures. The rotation function is where I ran into some problems for it was doing some calculations wrong and rotating around the screens center axis not the pieces axis. After I discovered my math’s logic bugs it worked a treat even when rotating or shifting near an edge. This program is relatively simple but there were some special cases to test for like trying to move past the board. I tried throwing every weird instance at the program and it seems to handle it well with no known bugs to report upon, other than the program does allow the user to rotate at edges. Since the assignment said this is ok, I did not do error checking for this case.

**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 next step for the Tetris assignment in the class. Knowing what I know now I wouldn’t have wasted time trying to understand every single thing about the OpenGL texture mapping, and just used the code after a normal understanding of it. If it weren’t for that the assignment would have taken me less than 3 hours. Three hours with debugging isn’t wholly that bad and I am proud of how it turned out.