**Programming Project Report**

Rhett Brandon

010722727

**Problem Statement:**

The goal of this programming assignment was to create a program we can use to implement the next ‘Angry Bricks’ project. The goals for project 2 just require us to create a 3-D brick that we can click, then drag the brick around the screen. The input for this program is the mouse pointer’s x-y coordinates. This input is used to calculate where to re-draw the cube. On a click and drag input, the user should see the cube follow the mouse pointer around the window. No error handling was required for this program.

**Design:**

There were only a handful of design choices made when designing this program. The major design decision was coming up with a formula to translate the mouse callback coordinates received into drawable coordinates in the view. To create this formula, I used the translation equations we did in class. Upon making this change I noticed that a click resulted in a redraw, however the cube would only redraw itself around the edges of screen, this was happening due to my values being integers instead of floats. Upon changing my integers to floats, the mouse callback was redrawing the cube where the mouse is clicked. Past this decision, all other design choice boiled down to which pieces of source code to use as an example from Professor Gauch’s library.

**Implementation:**

Most of this project was implemented using Gauch’s source code library. At first, I started with the cube.cpp, this program displays a 3-d cube, the only change made were to change the color of the cube face purple to green, this is just my preference. The next step was to create a program that would redraw the cube at the location of the mouse when the mouse was clicked. To find a useable mouse click I used the office3.cpp file, I only pulled out the mouse function, as well as the mouse callback from the main function. With this piece of code, I removed any reference to anything other than the GLUT\_DOWN function. The only other change I made was when translating the mouse callback coordinates to coordinates I could use when drawing the cube. The final step was to redraw the cube whenever the mouse is clicked down and drug across the screen. I used the motion callback from office3.cpp to accomplish that, I gutted the motion function and inserted the redisplay function in this motion callback. In total this program only took about an hour or two to complete.

**Testing:**

To test this correctness of this program I incrementally tested as I was programming. The project description gives clear steps and checkpoints when trying to implement this project. The first checkpoint was just displaying a cube on the screen, this was already done for us in the cube3.cpp .

The next checkpoint was to create a program that would redraw the cube when we clicked on the mouse.

The final checkpoint being to see the cube being drug around the screen when we click and drag the mouse. Everything worked as expected when testing my program. The only thing I would have liked to change is to center the cube on the mouse whenever I click and drag the cube.

**Conclusions:**

The result of this project was a success. The end result was a program that allows for a user to click and drag a cube around the view window. If I were to implement this program again I would try to center the cube on the mouse cursor when redrawing the cube. Overall this program took about 1 to 2 hours to implement.