Peter Lesslie
Project 3

I have been playing tournament chess since kindergarten, and I have always wanted to make a chess engine so designing a chess board seemed like the perfect project. Designing each of the pieces is sophisticated enough to show off the use of interesting geometries, while the board will eventually be textured. The one aspect that a chess board does not really show well is different types of lighting since they do not matter for the board; I want, however, in the later projects to put a candle next to the chess board to show a positional light source.

The program generates the view based upon a bounding sphere as specified in the requirements. The view aspect ratio is preserved when the user changes the size of the glut window. The three different projection types can be seen by pressing 'q' for oblique, 'p' for perspective, and 'o' for orthogonal. The program starts with perspective projection by default since that is the most realistic. The oblique projection is projected along d = <0.0f, 0.0995037, 0.995037>, which I arbitrarily chose this direction, and the program does not allow the user to change it. A zoom is implemented which can seen by pressing 'z' or space bar. It is not implemented correctly because we had not discussed how to implement zoom yet, so I implemented it by moving the projection plane forwards and backwards.

I have improved upon my previous models by adding a sphere and cylinder to the pawn and by creating a bishop. The king is the same as the previous project. I want to make a egg shape (rounded) for the top of the bishop, but I did not quite get the drawing done in time.

The phong lighting model is implemented in the vertex shader for this project. The positional light source is above the center of the board and it's effects can be seen on one of the pawns. For the later projects I want to make a candle to be the positional light source. I do not know how to make a flicker, flame like, light source though.

The main unique piece of this project was the implementation of a sphere class. I borrowed code from p. 633 in *Interactive Computer Graphics* by E. Angel, but I had to modify the code so that the sphere could be moved to any arbitrary position. I borrowed from your cylinder code, and also modified it so that it could be moved to any arbitrary position.

For the next project I would like to finish the rest of the pieces and create a function to “place” the pieces on a square so that it will be easier to move them around on the board. My final idea is to have a board that will animate a game.