**Checker Game Final Project Proposal**

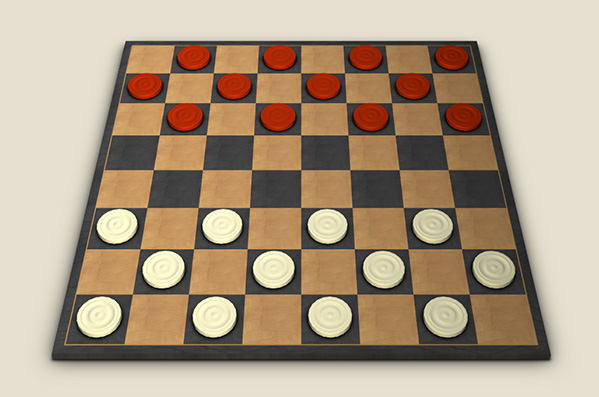
**Jordan Bertke**

**Checkers Game**

For my final project, I will be making an interactive checkers game. This will consist of a tiled surface to be used as the checkers board and 24 round playing pieces that will be used as checkers (12 for the player and 12 for the computer). When a player wishes to move a checker, they will select the occupied square with the mouse. Then, the board will illuminate the valid squares that the piece can move to. A second mouse click on the space that the player wishes to move to will result in the checker being "lifted" off the board and moved to the appropriate space. If the player jumps one of the computer's pieces, it will disappear from the board. When it's the computer's turn, it will make it's move automatically. The same rules that apply to jumping from the player's perspective will apply to the computer. Play ends when either the player or the computer run out of checkers.

**Sample Images**

**Checkers Board**



I plan to create a checkers board similar to the one seen here.

Source: http://www.osd.net/blog/web-development/3d-board-game-in-a-browser-using-webgl-and-three-js-part-2/

**Checkers Piece**

I will create the board pieces ("Checkers") to look something like this - hopefully with some of my own personalization - in both scarlet and grey.

Source: http://www.bcgameworks.com/store/index.php?main\_page=index&cPath=2

**Implementation**

I will use an integer matrix system to implement the board, and use a '1' to flag is a space is occupied and a '0' to signal that a space is not occupied. There will be a boolean variable to determine whether the piece occupying a particular space is yours or the computer's. Player pieces will be closer to the bottom of the screen and move "away" from it. Computer pieces will be placed further away and move "toward" the screen. If the board is represented logically by a matrix with **i** rows and **j** columns, then a given player piece would be able to move to [i-1][j-1] or [i-1][j+1]. A computer piece would be able to move to [i+1][j-1] or [i+1][j+1]. If the space is occupied by an opposing piece on the diagonal, and the space beyond is open, it could move to [i+-2][j+-2]. All of this assumes that i and j are greater than -1 and less than 8 (will not fall off the board).

**Timeline**

I am quite busy with all of my classes, but I plan on working on the project for **at least** half an hour every day until it is due. I plan on having all of the pieces, the board, and the motions done by the halfway point; and I plan to complete the logic for the player and computer moves during the second half of the time allotted to work on the project.