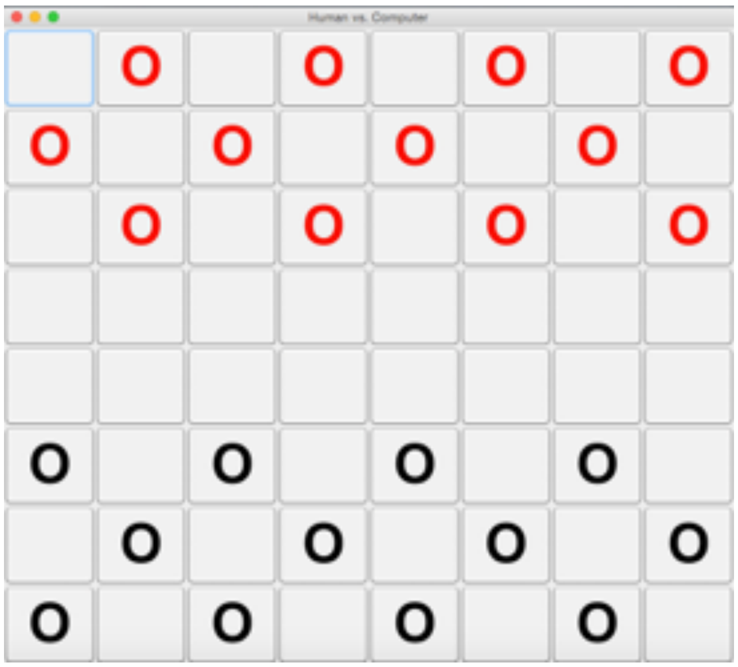
Checkers

In the modern era, some of us are simply too busy for these things some call “friends”. We saw this problem, and we decided to come up with a solution. The age-old game of checkers has been recreated in the modern language of java, but now you don’t need friends to have fun!



Do you think you can take on the AI? Implementing the MiniMax algorithm and Alpha-Beta Pruning, it sifts through all possible scenarios and finds the best move to make you feel bad about yourself. If you think you have the guts, just run CheckersDriver.java and you are on your way to your own demise.

Depending on the computational power of your computer, you may wish to adjust the depth of moves that your AI searches. To do so, just go to *line 29 inside the CheckersAINode* and change the if(currHeight==X) statement. X is the number of moves the AI will search ahead. Most computers can search 5 or 6 moves ahead.

To make a move, just click the piece you want to move and the place where you want to move it to. If it is a valid move, your wish will be granted and the AI will then respond with its own move. Neither side is forced to jump. If you successfully make a jump, however, and there is another available jump with that same piece, you will be required to make that jump. Get to the end of the board and your piece will be upgraded to a King and gain the ability to go backwards. Until then, you are stuck going diagonally and forward. You win when the AI runs out of pieces or the AI can't move. Vice versa, you lose when you run out of pieces or moves.

Brought to you by yours truly:

Aivant Goyal
Eric Tang
Jack Hirsch