Name: Abiyaz Chowdhury

ECE469 Fall 2015, Checkers Write-up

The American checkers program has been written in the Processing language, an open source language that specializes in media art and visual design, and is based on Java. All user input when playing the game is entered via the mouse. Pieces are moved and the board is edited via mouse clicks, and not mouse drags. The game displays a colored black and white checkerboard with all the pieces in the usual starting position, with white to move. Each square is indexed by a number, which is also displayed on the checkerboard for convenience. At any time in the game, the player whose turn it is to move is displayed at the top of the screen. If a player has won the game, this is also displayed in large letters at the top of the screen. At the right of the screen, the game displays every legal move for the player whose turn it is to move. For multi-capture moves, the full sequence of captures is displayed, as each legal multi-capture sequence is treated as a single move.

To capture a piece, a user simply selects the capturing piece, and then selects either the target of the capture, or the destination of the capturing piece. There is no difference between these two. For a multi-capture move, the user performs each capture sequentially, until no more captures are possible with the capturing piece (or if the capturing piece is kinged), upon which the turn is handed over to the opposing player.

The game features an edit button, which allows the user to edit the board by clicking on any board square, resetting or clearing the board, and also switching the player whose turn it is to move. Although the default starting position of the game has the pieces all in the black squares, the edit board feature can be used to utilize the white squares, if the user so desires.

The AI starts with a default time of 1 second, which the user can change using the buttons at the bottom of the screen. The minimum time is 1 second. The AI does not make its moves automatically, but rather displays its desired move when the AI button is pressed. This way, the AI can be consulted at any point in the game, and can advise both white and black, allowing the user to have the AI player against itself. This also lets the user keep track of what moves the AI has actually made. It also displays the depth it searched, the value of the best node it found, and the time elapsed for the search. The AI performs iterative deepening with minimax and alpha-beta until it runs out of time. If there is only one legal move, the AI will immediately display this move, without performing a minimax search. Sometimes, one move is so good (even if it does not guarantee a win) that the AI will very quickly choose this move, without expending all of the time it was allotted. Otherwise, the AI will take its time exploring as deep of a game tree as possible.

The heuristic ranges from 0 to 9999, where 0 is a win for black, and 9999 is a win for white. 1000 (and anything close to it) is an evaluation that the game is very even. Kings are currently valued at 1.4 times the value of men. Each man receives a bonus based on how close he is to being kinged. When the AI is at a material advantage, it will try to close the distance between its pieces and its opponent’s pieces. It will also try to force the opponent’s pieces out of corner positions.

I hope you enjoyed playing my AI.