Main

The main class is merely there to start off the chess program, and does not have any methods itself.

GUI

The purpose of the GUI class is to present the current state of the chess board in an intuitive, informative way to the player.

Taking Turns

In order to take their turn, the player must click on a piece which belongs to the team that is currently to move. This piece’s tile will turn red once selected, and locations to which this piece may move will turn blue. Following this, the player may then click on any of those blue tiles to move the selected piece to that location. If the player wishes not to select this piece, they can simply select another piece instead. By default, both the player and the AI will be able to take moves for either side, at any time. This allows for player vs. AI, player vs. player, and AI vs. AI matches, without the need to decide which match will be held beforehand. To make the AI take a move, one must simply click the “Next Move” button, and the AI will calculate its next move according to an alpha-beta pruning algorithm.

Special Moves

Special moves are those which extend beyond the commonly known, standard chess moves (namely en passant and castling). The AI will take these moves automatically whenever such a move would be beneficial according to the heuristic score and the decision algorithm. For the player to take an en passant move, they would click the button labeled “En Passant”, which will automatically determine where on the board such a move is possible, and take that move. If there is no such move possible, this button will not be possible to click. Likewise, with castling. If there is only one castling move possible, one must only click the button labeled “Castling”. This, like “En Passant” will automatically detect the possible castling move and make it. Alternatively, if there are two castling moves possible, the player will be able to choose which one they would like to make by use of the “Castle Left” and “Castle Right” buttons. These buttons will only become operational when there are two possible castling moves.

Heuristic

The simple heuristic used was a linear weighted sum of all the pieces on the board. The existence pieces on the white side increase the value of the fitness of a board. The existence of black pieces decrease the fitness value of the board. It follows then that higher scores benefit the player on the white team and lower scores benefit the player on the black team.

Game

The purpose of the game class is to handle all operations that span an entire game of chess. This class is also responsible for writing all moves made to the move log. This is done by the writeLog() method, which takes in two boards, (the one before a move was made, and the one after), and through use of a helper method, writes the move which represents the difference between these two boards to the move log.

Game Termination

The Game is also responsible for determining whether the conditions for game termination have been reached. This is done by detection methods kept in the board class. A checkmate is declared when a particular team is in check, it’s their turn to move, and they have no more legal moves left to make. A stalemate is determined when the player to move cannot move and is not in check, or there has been no capture in the last 50 moves, or the same move was made by either side for three turns in a row. If the conditions for game termination are reached, the ending of the Game is written to the log, and the Game is reinitialized.