Nathan Ly Mini-Camelot

CS 4613

The program runs with Python 2.7 and does not need anything else installed as the imported libraries should have been included with the standard Python package.

My program shows the result of the user's move and the AI's subsequent move with a visual game board. The player has a decision to go first or second when the program begins. In either case, the player is given a list of possible pieces to move which conforms to the obligatory capturing move. After a piece is selected, the player could go back to the list of potential pieces to move or pick a number from the list of possible moves for the current piece. The user's decision is summarized at the end and they can see the result of their move on the visual board. Input is required to let the AI begin its turn.

The AI runs the alpha-beta pruning algorithm to determine the best move and the position of all pieces after applying the move. The evaluation function used in the program is split in two parts. The function checks if the AI's pieces are on the player's castle. If one piece is on the castle, then the function calculates a score intended to get the closest piece to the adjacent castle for the win by using Chebyshev's distance. If no pieces are on the player's castle, then Chebyshev's distance is used to calculate the score. A positive increase in score is attributed to AI's (black) pieces being closer to the player's castle. A negative addition to the score is attributed to the player's (white) pieces being closer to the AI's castle. The board configuration after the AI's move is shown at the end and then it is the player's turn again.

The user has a choice of difficulty levels in this game which affects when the alpha-beta pruning uses the evaluation function. The difficulty level determines the cutoff (depth limit) of the alpha-beta pruning algorithm. There are 3 different levels (1, 2, and 3) which equates to a depth limit of 3, 4, and 5. The higher the limit, the more nodes are generated for a more in-depth search for moves that would likely lead to a better board configuration for the AI.