Applied Ai Assignment - 03

Report

Spring 2025

Prepared by: Waleed Abdullah
22i2714

Table of Contents

Chess game description

SO basically there are 2 main classes defined within the gane which have totally different functionality and used for diff purposes.
Most of the functions are done via the built in libraries in python which are chess and other imported to really help out with the implementation.

Compooterchess Class

This is where the computer decides its moves.

1) __init__(self, thinkinlevel: int = 3)

This function makes the ai which decided as many moves ahead as the number of thinking level which is 3 for deault cuz more than enough that much

2) bestMoveornot(self, board:chess.Board) -> chess.Move

This function makes sure the computer makes the best move. This is done via the minimum and maximum algorithm. It works by first checking the best possible move from all legal moves and then check what this move would in turn cause and then undoes it by poping from queue.
Then beta and alpha are compared to check and eliminate bad moves.

3) minimax(self, board: chess.Board, depth: int, alpha: float, beta: float, checkbest: bool) -> float

This uses recursion to evaluate and keep checkin the board again and again
It keeps on checking how many more moves to look ahead and the best possiblr moves for both players.
It makes sure the compooter chooses the best move by using max function and for our turn it uses min function to check mina dn update beta in oru case whereas alpha in compooters case.
Atlast to make use of alpha beta pruning if a bad move for sure is predicted.

4) checkboard(self, board: chess.Board) -> float

it gives scores to the board and so higher the score the tougher it is for us to wina dnbetter for compooter that is playin. It checks the probability of stale or checkmate and then assigns scores to pieces.
Capturing pieces puts th eopponent under pressure and so I have added a feature to give bonus point sfor eliminations. And also for being at the center positions;

Wholechess class

In dis class the working of game and also how it looks at the cli is handled.

1) __init__(self)

In this funcvtion a new board is started and also a new computer is made to play against. It is only a basic function used to initialize only.

2) showboard(self)

This function shows the board at the terminal and it then makes rows and columns via loops.
It then uses symbols to shows pieces and also coordinates via alphabets and numbering.

3) piecesymbol(self, piece: Optional[chess.Piece]) -> str

This function is used to finally convert a normal lookin piece to a symbol it also makes sure it is white or black to differentiate on the board and hence avoid any confusions that may occue.

4) play(self)

Used to run the main loop that is for the game. First of all shows a welcome message to give ujer good feels and then gets input and uses it to check if the move the user trynna make is possible and legal or not. It is also considered illegal if the user is checked or something like dat.
For the computers turn this function calls the bestMoveonot function to make sure the best possible move is played rom computers side.
It also keeps and eye on the result if it's a stalement or any other type of draw.

5) if __name__ == "__main__":

Finally used to run our whole script.