

Report of Assignment #2

Muhammad Ahmed Baig

20i-1884

CS-B

Introduction: This report aims to explain a Python code that implements a chess AI using the Minimax algorithm and the Python-Chess library. The program allows a user to play against the AI, which computes its moves based on the evaluation of the chessboard's material balance.

1. Importing the required library: The Python-Chess library is imported to work with chess data structures and to perform chess-related operations.
2. Display function: The Display(b) function takes a single parameter, b, which represents the chessboard. The function prints the board's current state using the Python-Chess library's built-in string representation.
3. Evaluate function: The Evaluate(b) function takes the chessboard, b, as its input and calculates the material balance by summing the piece types for both white and black pieces. It returns the difference between the white and black material, representing the current material advantage for white.
4. Minimax function: The Minimax(b, d, is_maximizing, a, be) function is a recursive implementation of the Minimax algorithm with alpha-beta pruning. It takes five parameters:
 - b: the current chessboard
 - d: the search depth
 - is_maximizing: a boolean indicating if the current node is maximizing or minimizing
 - a: alpha, the best value found so far for the maximizing player
 - be: beta, the best value found so far for the minimizing player

The function evaluates the board if the search depth is 0 or the game is over. For maximizing nodes, it iterates over legal moves, updating the maximum evaluation and alpha values. For minimizing nodes, it iterates over legal moves, updating the minimum evaluation and beta values. It also implements the alpha-beta pruning technique to reduce the search tree's size.

5. BestMove function: The BestMove(b, d) function takes two parameters, the chessboard b and search depth d. It iterates over all legal moves and uses the Minimax function to evaluate each move. It returns the move with the highest evaluation.
6. Main function: The main() function sets up the chessboard and enters a loop that continues until the game is over. It alternates between the user's turn and the AI's turn. The user is prompted to enter a move in UCI notation, and the move is validated before being pushed onto the board.

On the AI's turn, the BestMove function is called with a search depth of 3, and the move is pushed onto the board.

Conclusion: The provided code implements a simple chess AI that allows a user to play against the computer. The AI uses the Minimax algorithm with alpha-beta pruning to compute its moves and evaluates the board based on material balance. The user interacts with the program by entering moves in UCI notation, and the game continues until a terminal position is reached.

Outputs:

```
r n b q k b n r
p p p p p p p p
. . . . .
. . . . .
. . . . .
. . . . .
p p p p p p p p
R N B Q K B N R
Enter your move (e.g., 'e2e4'): e2e4
r n b q k b n r
p p p p p p p p
. . . . .
. . . . .
. . . . P . . .
. . . . .
p p p p . p p p
R N B Q K B N R
AI is thinking...
r n b q k b . r
p p p p p p p p
. . . . . n
. . . . .
. . . . P . . .
. . . . .
p p p p . p p p
R N B Q K B N R
Enter your move (e.g., 'e2e4'): e4e5
r n b q k b . r
p p p p p p p p
. . . . . n
. . . . P . . .
. . . . .
p p p p . p p p
R N B Q K B N R
AI is thinking...
r n b q k b r .
p p p p p p p p
. . . . . n
. . . . P . . .
. . . . .
p p p p . p p p
R N B Q K B N R
Enter your move (e.g., 'e2e4'): e5e6
r n b q k b r .
p p p p p p p p
. . . . P . . n
. . . . .
```

|| meet.google.com is

```
AI is thinking...
r n b q k b r .
p p p p p p p
. . . . . n
. . . . . P . . .
. . . . .
P P P P . p p p
R N B Q K B N R
Enter your move (e.g., 'e2e4'): e5e6
r n b q k b r .
p p p p p p p
. . . . . P . . n
. . . . .
. . . . .
P P P P . p p p
R N B Q K B N R
AI is thinking...
r n b q k b . r
p p p p p p p
. . . . . P . . n
. . . . .
. . . . .
P P P P . p p p
R N B Q K B N R
Enter your move (e.g., 'e2e4'): e6f7
r n b q k b . r
p p p p p P p p
. . . . . n
. . . . .
. . . . .
P P P P . p p p
R N B Q K B N R
AI is thinking...
r n b q . b . r
p p p p p k p p
. . . . . n
. . . . .
. . . . .
P P P P . p p p
R N B Q K B N R
Enter your move (e.g., 'e2e4'): 
```

In []: