

## Artificial Intelligence Assignment 3 – AI Chess

Mahad Ahmed 20i-0426 Section F

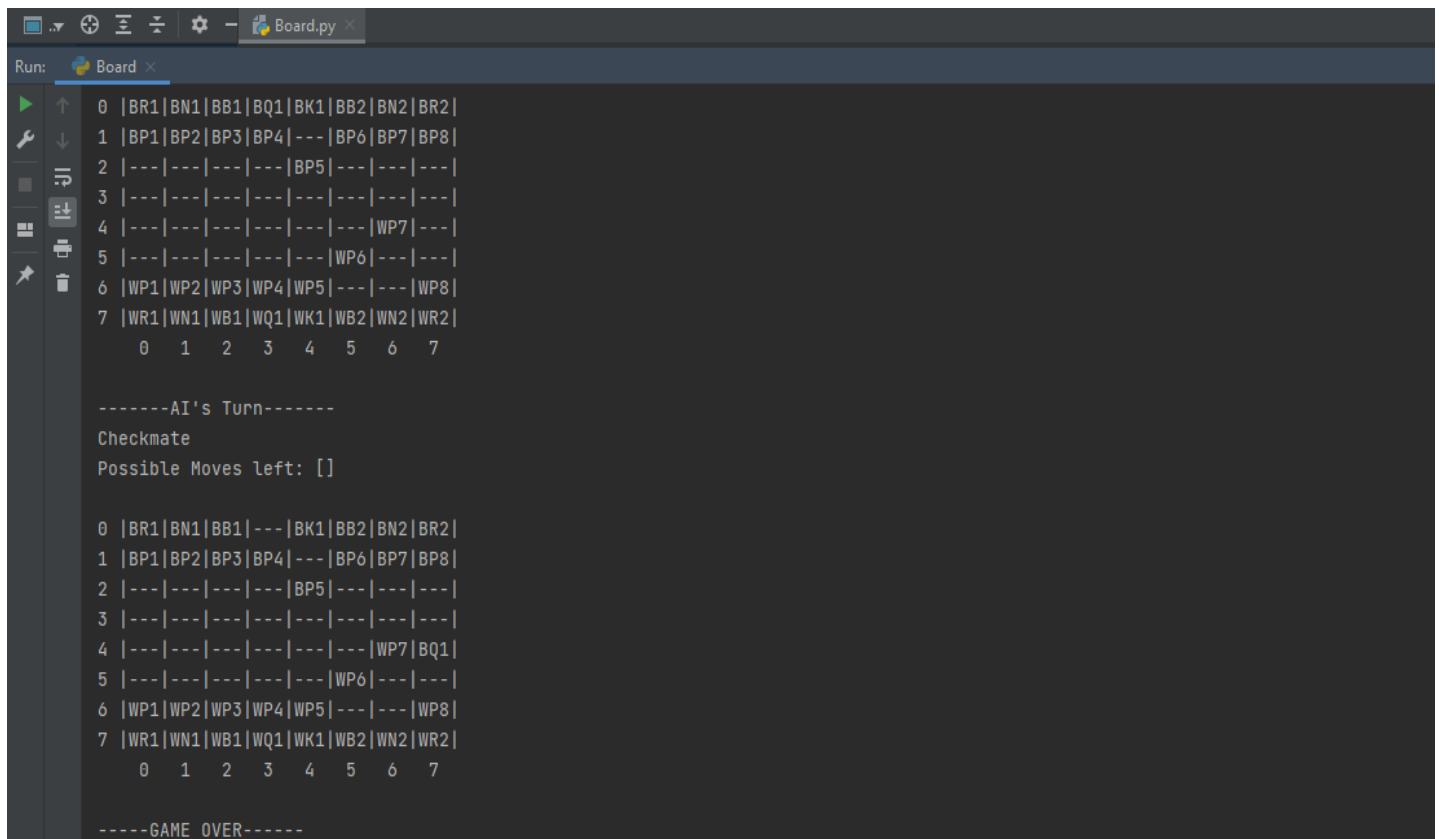
### Question 1.

In this program I have made a chess game in which one player is human and the other is the AI. Initially the human has the choice to play as White or as Black and the AI is given the other color.

In this program we represent our pieces in 3 characters. The first is the color, the second is the piece and the third is the serial number. E.g., 'WB2'. For the player module the player chooses his move by specifying the piece they want to move and the new coordinates. The piece and coordinates are then validated by checking if the move is valid. We use the `get_all_moves()` function which returns all the possible moves for that player. This function checks the moves of all pieces one by one and keeps in mind the rules of the chess game such as legal moves, attacking moves, own piece blocking etc.

The player and AI alternate performing their moves. The AI's moves are generated using the alpha beta pruning algorithm. In this we specify the depth to which the AI should go and analyze the moves. Depth 1 will only see the immediate move whereas depth 3 will recursively check out the next few moves and predict the best moves from all scenarios. To help the AI make the best decision we use two heuristic functions. One is the piece advantage function which rates the pieces on the board for both players and then returns the difference. The other is the mobility heuristic which sees the many possible moves are possible for both the players and returns the difference. If a move has a large heuristic, it means the move is good for the player to perform.

The AI checks the heuristics for all moves. The `get_all_moves()` function will return all moves for a player. If a player is in check, it will only return the moves that can save the player from check. If a player is in check and has no moves left it will be a checkmate which is checked in the `is_check_mate()` function. A stalemate is seen if the player is not in check but has no moves possible by `is_stale_mate()` function.



```
Board.py x
Run: Board x
0 |BR1|BN1|BB1|BQ1|BK1|BB2|BN2|BR2|
1 |BP1|BP2|BP3|BP4|---|BP6|BP7|BP8|
2 |---|---|---|---|BP5|---|---|---|
3 |---|---|---|---|---|---|---|---|
4 |---|---|---|---|---|---|WP7|---|
5 |---|---|---|---|---|WP6|---|---|
6 |WP1|WP2|WP3|WP4|WP5|---|---|WP8|
7 |WR1|WN1|WB1|WQ1|WK1|WB2|WN2|WR2|
  0  1  2  3  4  5  6  7

-----AI's Turn-----
Checkmate
Possible Moves left: []

0 |BR1|BN1|BB1|---|BK1|BB2|BN2|BR2|
1 |BP1|BP2|BP3|BP4|---|BP6|BP7|BP8|
2 |---|---|---|---|BP5|---|---|---|
3 |---|---|---|---|---|---|---|---|
4 |---|---|---|---|---|---|WP7|BQ1|
5 |---|---|---|---|---|WP6|---|---|
6 |WP1|WP2|WP3|WP4|WP5|---|---|WP8|
7 |WR1|WN1|WB1|WQ1|WK1|WB2|WN2|WR2|
  0  1  2  3  4  5  6  7

-----GAME OVER-----
```