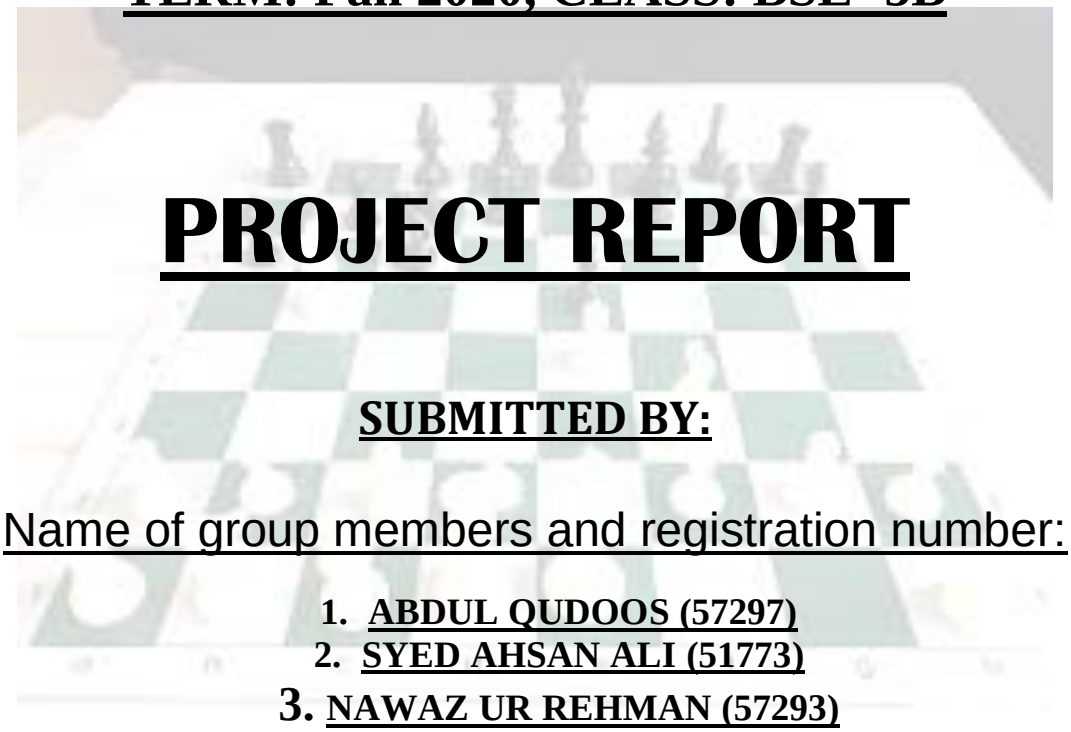
Bahria University
Karachi Campus

COURSE: CSC 411 ARTIFICIAL INTELLIGENCE
TERM: Fall 2020, CLASS: BSE- 5B



PROJECT REPORT

SUBMITTED BY:

Name of group members and registration number:

1. **ABDUL QUDOOS (57297)**
2. **SYED AHSAN ALI (51773)**
3. **NAWAZ UR REHMAN (57293)**

SUBMITTED TO:

Engr. Muhammad Rehan Baig

SIGNED _____ **REMARKS:** _____ **SCORE:** _____

Table of Contents

TABLE OF FIGURE:

INTRODUCTION:

The semester project that we have made for this course is a Recreational and a Competitive board game played between two players known as CHESS GAME. The Chess game follows the basic rules of chess, and all the chess pieces only move according to valid moves for that piece. Our implementation of Chess is for a player and an AI Agent. It is played on an 8x8 checked board.

ARTIFICAL INTELLIGENCE ALGORITHMS IMPLEMENTED:

- Min-Max Algorithm.
- Alpha-Beta Pruning.

MIN-MAX ALGORITHM:

Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory. It provides an optimal move for the player assuming that opponent is also playing optimally. Mini-Max algorithm uses recursion to search through the game-tree. Min-Max algorithm is mostly used for game playing in AI. Such as Chess, Checkers, tic-tac-toe, go, and various two-players game. This Algorithm computes the minimax decision for the current state. In this algorithm two players play the game; one is called MAX and other is called MIN.

ALPHA-BETA PRUNING:

Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm. Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prunes the tree leaves but also entire sub-tree.

The two-parameter can be defined as:

Alpha: The best (highest value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is -∞.

Beta: The best (lowest value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is +∞.

The Alpha-beta pruning to a standard minimax algorithm returns the same move as the standard algorithm does, but it removes all the nodes which are not really affecting the final decision but making algorithm slow. Hence by pruning these nodes, it makes the algorithm fast.

EXPLAINATION OF GAME:

Chess is played on a board of 64 squares arranged in eight vertical rows called files and eight horizontal rows called ranks. These squares alternate between two colors: one light, such as white, beige, or yellow; and the other dark, such as black or green. The board is set between the two opponents so that each player has a light-colored square at the right-hand corner.
Chess begins with the division of pieces – white and black sets. Each set contains 16 pieces, as follows:

- 1 king
- 1 queen
- 2 rooks
- 2 bishops
- 2 knights
- 8 pawns

MOVES

The board represents a battlefield in which two armies fight to capture each other's king. A player's army consists of 16 pieces that begin play on the two ranks closest to that player. There are six different types of pieces: king, rook, bishop, queen, knight, and pawn; the pieces are distinguished by appearance and by how they move. The players alternate moves, White going first.

KING

White's king begins the game on e1. Black's king is opposite at e8. Each king can move one square in any direction, e.g., White's king can move from e1 to d1, d2, e2, f2, or f1.

ROOK

Each player has two rooks (formerly also known as castles), which begin the game on the corner squares a1 and h1 for White, a8 and h8 for Black. A rook can move vertically or horizontally to any unobstructed square along the file or rank on which it is placed.

BISHOP

Each player has two bishops, and they begin the game at c1 and f1 for White, c8 and f8 for Black. A bishop can move to any unobstructed square on the diagonal on which it is placed. Therefore, each player has one bishop that travels only on light-colored squares and one bishop that travels only on dark-colored squares.

QUEEN

Each player has one queen, which combines the powers of the rook and bishop and is thus the most mobile and powerful piece. The White queen begins at d1, the Black queen at d8.

KNIGHT

Each player has two knights, and they begin the game on the squares between their rooks and bishops—i.e., at b1 and g1 for White and b8 and g8 for Black. The knight has the trickiest move, an L-shape of two steps: first one square like a rook, then one square like a bishop, but always in a direction away from the starting square. A knight at e4 could move to f2, g3, g5, f6, d6, c5, c3, or d2. The knight has the unique ability to jump over any other piece to reach its destination. It always moves to a square of a different color.

CAPTURING

The king, rook, bishop, queen, and knight capture enemy pieces in the same manner that they move. For example, a White queen on d3 can capture a Black rook at h7 by moving to h7 and removing the enemy piece from the board. Pieces can capture only enemy pieces.

PAWNS

Each player has eight pawns, which begin the game on the second rank closest to each player; i.e., White's pawns start at a2, b2, c2, and so on, while Black's pawns start at a7, b7, c7, and so on. The pawns are unique in several ways. A pawn can move only forward; it can never retreat. It moves differently than it captures. A pawn moves to the square directly ahead of it but captures on the squares diagonally in front of it; e.g., a White pawn at f5 can move to f6 but can capture only on g6 or e6. An unmoved pawn has the option of moving one or two squares forward. This is the reason for another peculiar option, called en passant—that is, in passing—available to a pawn when an enemy pawn on an adjoining file advances two squares on its initial move and could have been captured had it moved only one square. The first pawn can take the advancing pawn en passant, as if it had advanced only one square. An en passant capture must be made then or not at all. Only pawns can be captured en passant. The last unique feature of the pawn occurs if it reaches the end of a file; it must then be promoted to—that is, exchanged for—a queen, rook, bishop, or knight.

CASTLING

The one exception to the rule that a player may move only one piece at a time is a compound move of king and rook called castling. A player castles by shifting the king two squares in the direction of a rook, which is then placed on the square the king has crossed. For example, White can castle kingside by moving the king from e1 to g1 and the rook from h1 to f1. Castling is permitted only once in a game and is prohibited if the king or rook has previously moved or if any of the squares between them is occupied. Also, castling is not legal if the square the king starts on, crosses, or finishes on is attacked by an enemy piece.

END OF THE GAME

Games can be won in the following ways:

- **CHECKMATE:**
  The player whose turn it is to move is in check and has no legal move to escape check.

- **RESIGNATION:**
  A player may resign, conceding the game to the opponent. Most tournament players consider it good etiquette to resign in a hopeless position.

- **WIN ON TIME:**
  In games with a time control, a player wins if the opponent runs out of time, even if the opponent has a superior position, if the player has a theoretical possibility to checkmate the opponent were the game to continue.

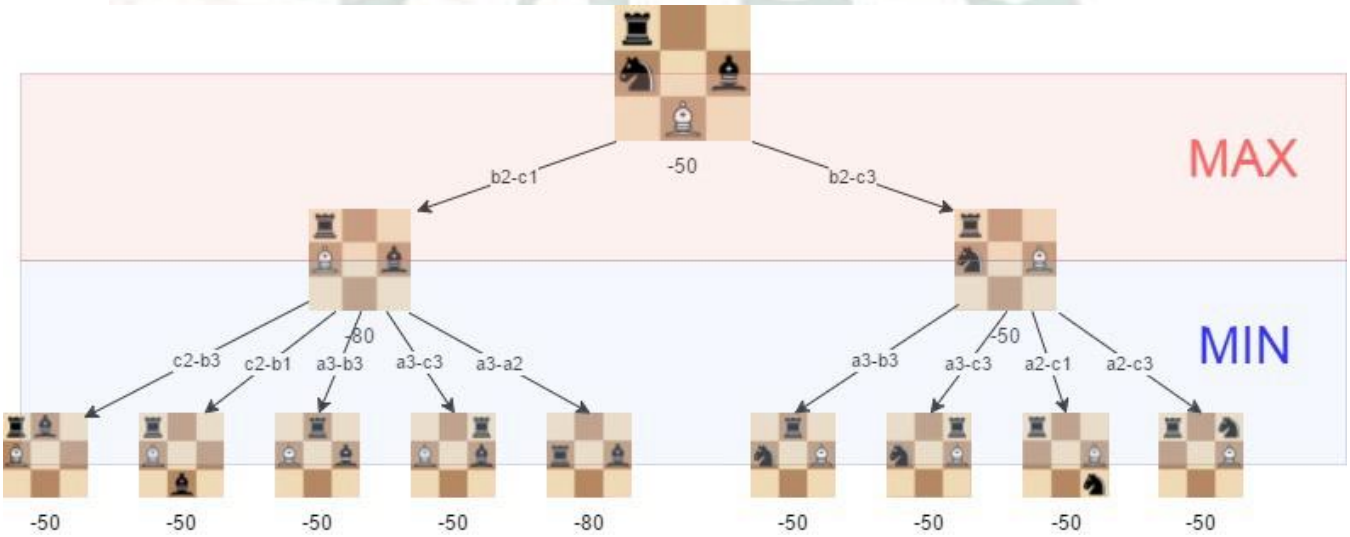
- **FORFEIT:** A player who cheats, violates the rules, or violates the rules of conduct specified for the tournament, can be forfeited

EXPLAINATION OF ALGORITHMS:

MIN-MAX ALGORITHM:

we are going to create a search tree from which the algorithm can chose the best move. This is done by using the Minimax algorithm.
In this algorithm, the recursive tree of all possible moves is explored to a given depth, and the position is evaluated at the ending "leaves" of the tree.

After that, we return either the smallest or the largest value of the child to the parent node, depending on whether it's a white or black to move. (That is, we try to either minimize or maximize the outcome at each level.)



A visualization of the minimax algorithm in an artificial position. The best move for white is **b2-c3**, because we can guarantee that we can get to a position where the evaluation is **-50**

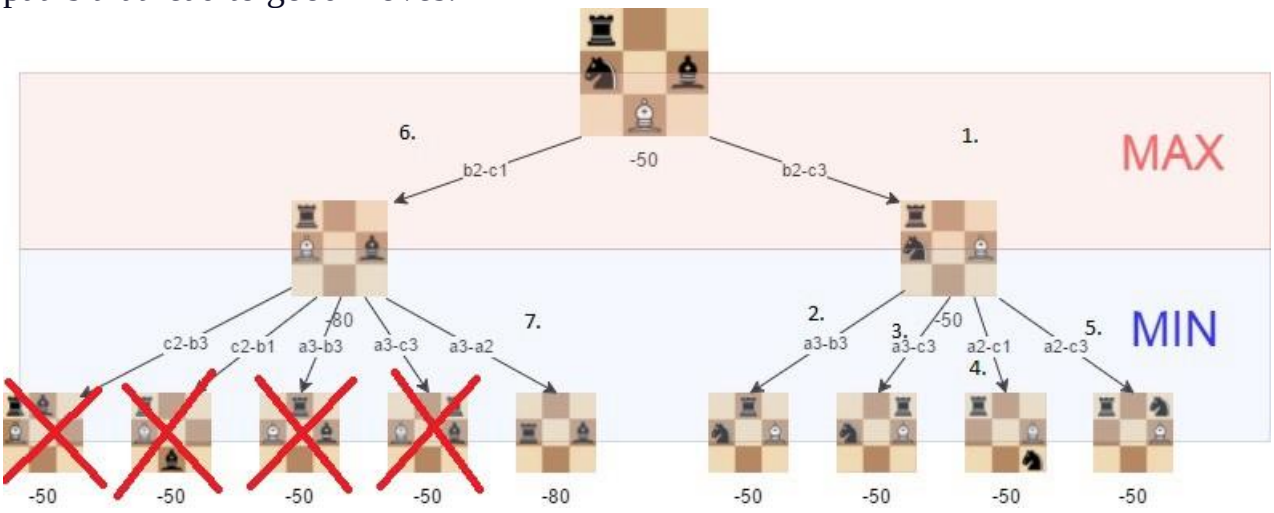
With minimax in place, our algorithm is starting to understand some basic tactics of chess:

ALPHA-BETA PRUNING:

Alpha beta pruning is an optimization method to the minimax algorithm that allows us to disregard some branches in the search tree. This helps us evaluate the minimax search tree much deeper, while using the same resources.
The alpha-beta pruning is based on the situation where we can stop evaluating a part of the search tree if we find a move that leads to a worse situation than a previously discovered move.

The alpha-beta pruning does not influence the outcome of the minimax algorithm it only makes it faster.

The alpha-beta algorithm also is more efficient if we happen to visit **first** those paths that lead to good moves.



The positions we do not need to explore if alpha-beta pruning issued and the tree is visited in the described order.

CODE AND ITS IMPLEMENTATION:

The code of our game is divided into four different python files.

- Board.py
- Pieces.py
- Ai.py
- Main.py.

We have uploaded our complete code on git with all the basic comments so that any one can understand it easily.

GIT-HUB URL:

- ***ABDUL QUDOOS***

https://github.com/qudoos183055/ARTIFICAL-INTELLIGENCE-PROJECT.git

- ***NAWAZ UR REHMAN:***

https://github.com/nawaz079/A.I-Project.git

- ***SYED AHSAN ALI:***

https://github.com/ahsan-dev99/AI-Project-.git

OUTPUT:

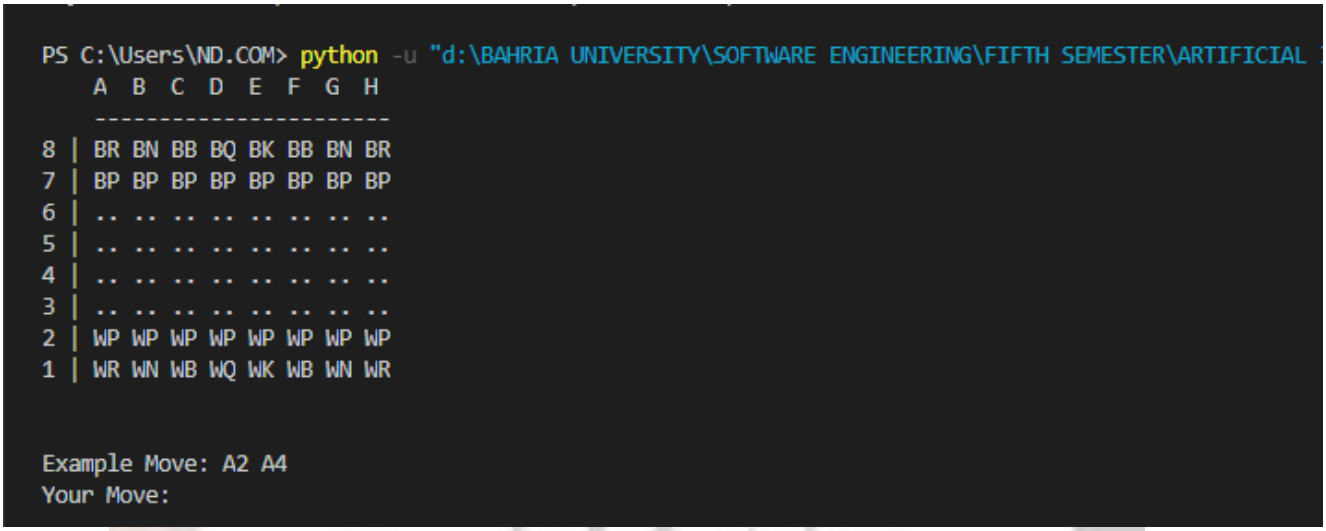


Figure 1 Initial Point From where our game start.

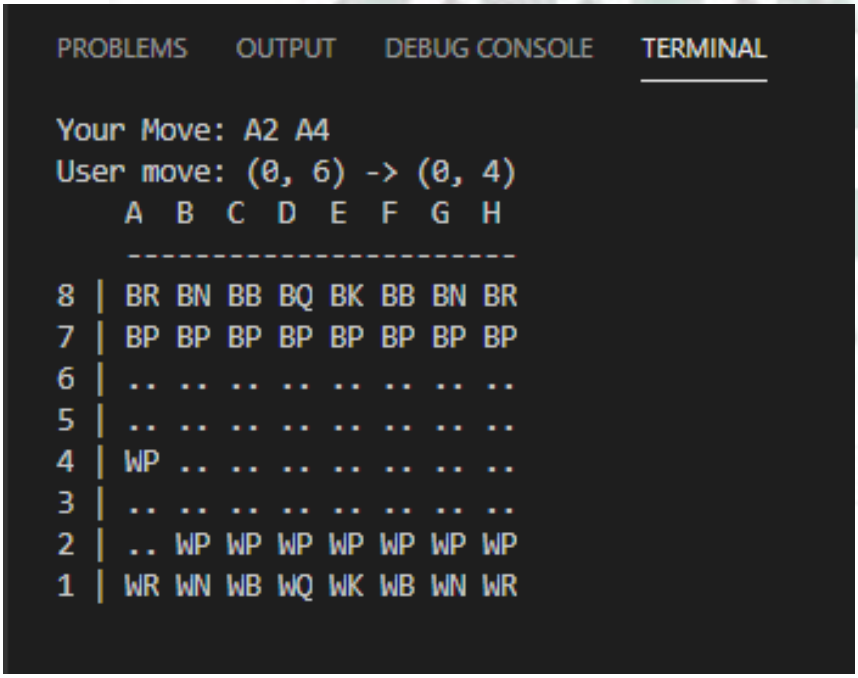


Figure 2 PLAYER MOVE

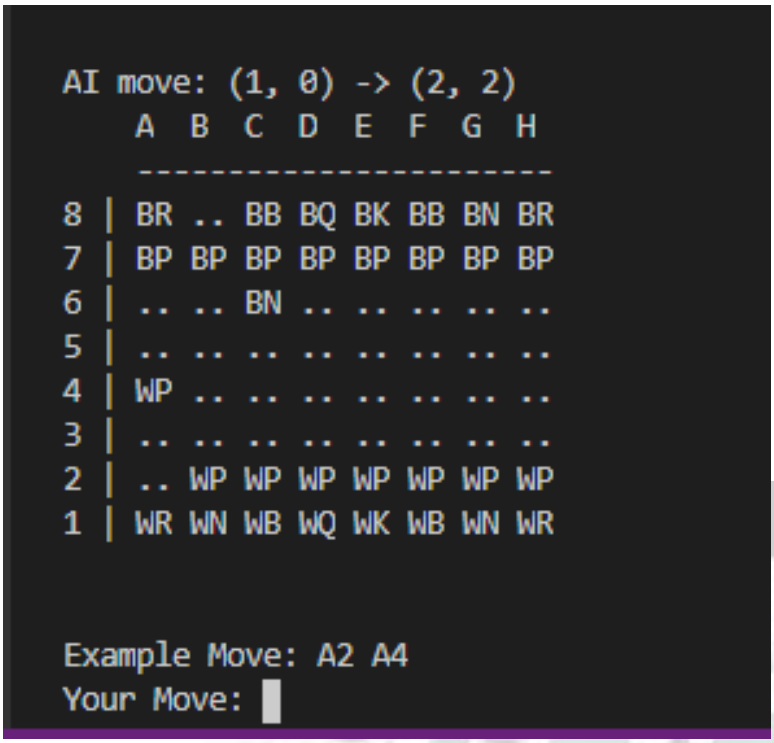


Figure 3 A.I MOVE.

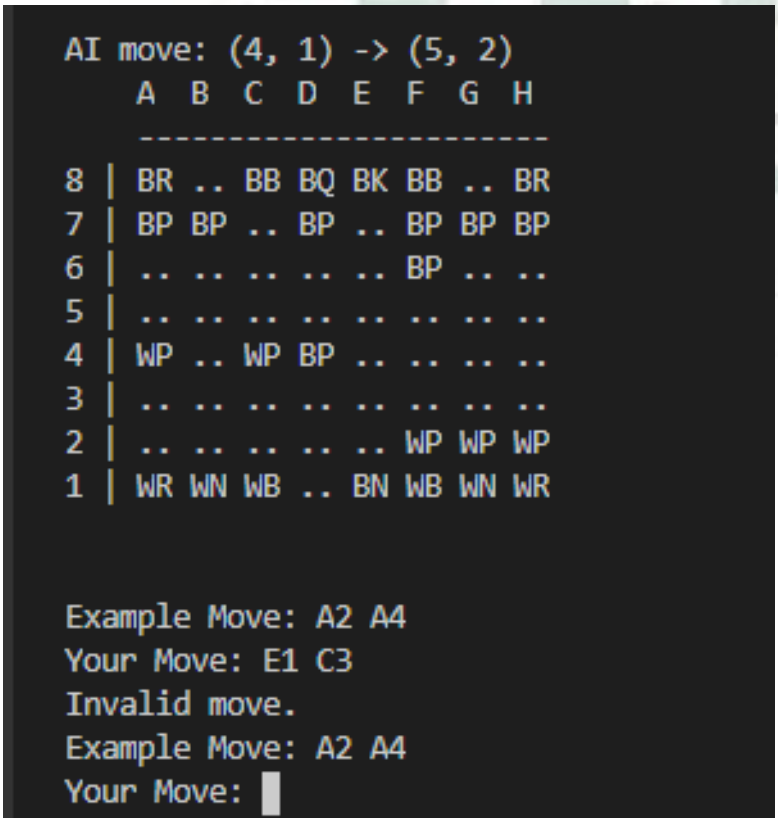


Figure 4 INVALID MOVE

CONCLUSION:

This project was completed in fulfillment of the course requirement of Artificial Intelligence. It provided a very good opportunity to obtain both the theoretical knowledge and practical experience. It helps us to get more familiar with the Artificial Intelligence Algorithms. The result of our project meets both the requirement of the course and our expectation. A competitive game between human user and the machine is realized.

REFRENCE:

https://www.javatpoint.com/ai-alpha-beta-pruning

https://www.javatpoint.com/mini-max-algorithm-in-ai

https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/

https://www.youtube.com/watch?v=l6ex08F92F0&t=101s