COSC 3P71 - Chess Project

Prepared by Karina Bissessar and Shivdeep Khangura
2019 December 19

A DESIGN OF PROGRAM

The chess-playing program was developed using Python 3.7. The program respects the following rules of chess:

- The movement of pieces (including *castling* and *en-passant*)
- Piece promotion - a pawn reaching the other end of board is promoted to a queen by default
- Check
- Checkmate
- Stalemate

The program uses a game tree search scheme with *alpha-beta pruning*. In addition, the program permits two user supplied parameters:

1. The depth of the search
2. The board setup to be used initially (I.e. either the standard starting board or a user specified starting board.

Useful ideas for developing the chess playing program was obtained from the website: https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/

The program allows the computer (AI) to play against a human player. The human plays white while the AI plays black.

The program dumps out the current board as an ASCII table with upper-case characters representing black pieces and lower-case characters representing white pieces. The traditional start-up board looks like this:

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R | N | B | Q | K | B | N | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P | P | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | p | p | p | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
```

White pieces are: **r** for Rook, **n** for knight , **b** for bishop, **q** for queen, **k** for king and **p** for pawn.
Black pieces are: **R** for Rook, **N** for knight , **B** for bishop, **Q** for queen, **K** for king and **P** for pawn.

The board is an 8 by 8 grid. The columns are labelled A to H while the rows are labelled 1 to 8 as shown above. The locations of pieces on the board are given via coordinates (column and row). For example, the queen-side white knight on the board above is at coordinate B1 and can move to either position A3 or C3. To move the knight from B1 to C3 both the to and from co-ordinates need to be specified e.g., B1 C3.

The program also allows you to dump out all of the game's moves when the end is reached.

Brief Outline of how the program works:

There are two users of the program - the human user (you!) and the computer (the AI). You as the user will be playing with the white pieces while the AI plays with the black pieces.

You first need to choose whether you wish to start the game with a standard traditional starting chessboard or your own individual starting board setup. You'll also need to specify the depth of search of the game search tree to determine each of the AI's moves.

Naturally in a game of chess, the white side starts the game, thus you will have the first move and every other move thereafter until the game comes to an end. The AI has the second move and every other move thereafter until the game comes to an end.

You specify your move in the form of a start and end co-ordinate, including a space in between. The program will validate the move to determine if such a move is possible. If such a move is possible, it will be made, otherwise it will return a message saying invalid play. Then it's the turn of the AI to make its move

For the AI's move, the program creates a **game tree** starting at the current position of the chessboard and containing all possible moves from each position. Game trees are important in artificial intelligence because one way to pick the best move in a game is to search the game tree using any of the numerous tree search algorithms, combined with minimax-like rules to prune the tree. The search algorithm used in this chess application to determine the best move for the AI was **alpha–beta pruning** which decreases the number of nodes that are evaluated by the minimax algorithm in the search tree. It stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. Such moves need not be evaluated further. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision. (Explanations obtained from Wikipedia).

After the AI's move is determined, it is executed and a test is performed to determine whether the white king is in check. Then it's the human player's turn to play again. If the move specified by the human player will result in the white king getting into check or remaining in check, then the move is deemed invalid. In such a case, the human player will have four additional opportunities to get it right or forfeit the game.

The human player and the AI player play alternatively until the game comes to an end. At the end of the game, the user has the option to look at a dump of all the moves that were made during the game.

Validation of moves

The pieces.py program contains information that helps determine all the possible moves of pieces on the current chessboard. Once a user-entered move is deemed valid and does not result in the player's king getting into check or remaining in check, it will be executed.

Heuristic for determining the score of the current chessboard

In order to determine the AI's best move, the program uses a heuristic to determine the score of the chessboard after a certain potential move is made by either the AI or the human player. This heuristic is based on both the values of the pieces on the board and the location of the pieces on the board. The values of pieces used in this application were : Pawn (100), Knight (320), Bishop (330), Rook(500), Queen (900) and King (9000).

We add an addition factor to the evaluation that takes into account the position of the pieces. For example, a knight on the center of the board is better (because it has more options and is thus more active) than a knight on the edge of the board. The tables showing the values of the relative positions for each piece type can be seen in the source code for ai.py.

The heuristic value is equal to the calculated values for the white pieces minus the calculated values for the black pieces.

See the User Manual below for instructions on how to run the program. **A sample of the program run is shown in *Chess Game Sample Run.pdf*.**

B USER MANUAL

The software for this assignment comprises four programs:

chess.py	This is the driver program which consists of the user interface, the steps to set up the initial chess board and the functions that allow the alternate moves of the human player and the AI. The program also contains the functions to determine whether the board is in check before and after the human player's move.
ai.py	This file contains the definition of the board and allows for the creation of either a new chess board (the traditional starting board) or a user-specified staring chess board. It also contains the code for performing the different moves for the individual chess pieces, determining whether a particular colour is in checkmate.
board.py	This file defines the different pieces on the chessboards and the possible moves that they can make. The moves for Rook, Knight, Bishop and Queen are straightforward. However for the King, there is a castling move (either queenside or kingside) and for the Pawn, there is a en-passant move. This file also contains the function displaying the board after each move.
pieces.py	This is the AI part of the application. It uses a game tree search scheme with alpha-beta pruning. A heuristic used for determining the score of the chess board after each possible move. The heuristic uses both the positions and values of the pieces on the board.

Running the program

- Bring up the Python 3.7's IDLE (Integrated Development and Learning Environment).
- Then execute chess.py interactively on IDLE.
- Please ensure that the four programs chess.py, ai.py, board.py and pieces.py are in the same directory.
- Also ensure that the sample data files load1.txt, load2.txt and load3.txt are in the same directory

When the chess program is executed, you get the following introductory screen:

```
+=================================================================+
|     Fall 2019 COSC 3P71 Artificial Intelligence: Project        |
|       Students: Karina Bissessar & Shivdeep Khangura            |
+-----------------------------------------------------------------+
|           Implementation of a Chess-Playing Program             |
+=================================================================+
USER SUPPLIED CONTROL PARAMETERS

Enter DEPTH of search - 3 or less
```

You are required to enter the depth of the search of the game tree to determine the AI's moves. It is recommended that values of 3 or less be used as higher values would slow down the response time significantly (unless you have a super-fast computer).

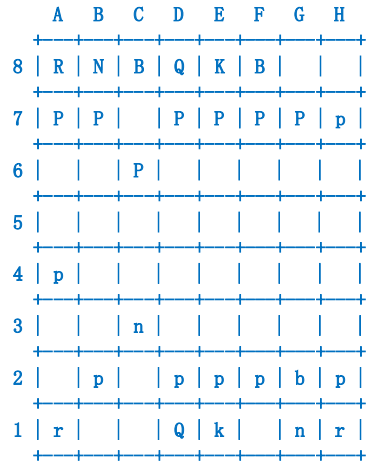
You are then asked to choose your initial board

```
Choose your board
(1) Traditional initial board (2) Specify your own initial board
```

If you choose option 1 you will be provided with a starting chessboard that look like this:

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R | N | B | Q | K | B | N | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P | P | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | p | p | p | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
```

However, if you choose option 2, you will have to specify the text file from which you want to read in your initial chessboard. If for example, you want a starting chessboard like this:



you'll need to create a text file as input which contains the following eight lines of data:

```
RNBQKB
PP PPPPp
  P


p
  n
 p pppbp
r  Qk nr
```

Just make sure and enter a space where there is no piece on the square.

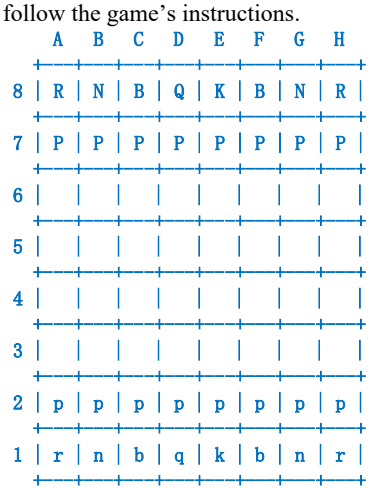
Once you select option 2, you will be asked to enter the file name
**** Please enter the file name:**

There are three text files: load1.txt, load2.txt and load3.txt which have been provided as samples for you to use as initial chessboards. Once you enter a valid filename with valid data, the data will be loaded on a 8 by 8 grid in the application.

These are the instructions that will then be displayed on the screen.

```
Upper case letters are black pieces, lower case letters are white pieces. As the
human, you have control of the white pieces, while the AI has control of the black
pieces. You enter moves using the following format: xfrom, yfrom xto, yto. e.g. to ,
move the knight from G1 to H3, simply enter G1 H3 as the required move
If you wish to quit the game at any time, enter X
```

The game instructions and starting chessboard (for this example we are using the traditional starting chessboard) follow the game's instructions.



```
Your Move:
```

As the human player, you have been assigned white and therefore you have the first move. You enter your move in the format (start co-ordinate) (end co-ordinate). For example, if you want to move the white pawn from coordinate C2 to C4, you simply enter:

C2 C4

After each move is made, you will get a confirmation such as “White pawn has just captured Black Pawn” or “White Pawn moved”

This is what you’ll see on the screen after this particular move is made:

```
Your Move: C2 C4
White Pawn moved

    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R | N | B | Q | K | B | N | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P | P | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | p | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
```

Note that if you entered the move in lower case characters (such as c2 c4), the data will also be accepted.

At this point, the AI will make its move. If you had specified a search depth of 3 or more, there might be a delay before the AI’s move is made. The move made by the AI will be given together with a description of the move.

```
AI is thinking....Please wait
Black Knight moved
AI move: (B8) -> (C6)

    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R |   | B | Q | K | B | N | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P | P | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   | N |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | p | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
```

After each move is made, you will get a confirmation such as “White pawn has just captured Black Pawn” or “White Pawn moved”

After this it's your turn again. You are requested to enter a move. Note that if it is an invalid move, you will be informed and you will be asked to enter another move. For example:

```
Your Move: b1 c2
Invalid move.
Your Move: b1 c3
White Knight moved
```

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R |   | B | Q | K | B | N | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P | P | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   | N |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   | n |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | p | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r |   | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
```

After this, it's the AI's turn again

```
AI is thinking....Please wait
Black Knight moved
AI move: (C6) -> (E5)
```

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R |   | B | Q | K | B | N | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P | P | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   | N |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   | n |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | p | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r |   | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
Your Move:
```

Note that if you attempt to enter a move that will result in your king getting in check or remaining in check, you will be informed. Also after each move by you or the AI you will be provided details of the move (such as White Knight has captured Black Pawn) or En-passant completed or Castling executed). Information will also be provided whenever your king or the AI's king has been checked.

You and the AI will continue taking turns until the game ends with a checkmate or stalemate. However, if at any time you wish to end the game, you can simply enter the letter X as the move.

```
Your Move: X
Do you want to look at a dump of the moves of the game?? (Y)es or (N)o
```

After you quit the game or the game ends in a checkmate or stalemate, you will be asked whether you want to look at a dump of the moves of the game. If you choose Y, then the moves of the game will be displayed as follows:

```
['c2 c4', 'Nb8 c6', 'nb1 c3', 'Nc6 e5']
Another chess game ?? (Y)es or (N)o
```

If you want to play another chess game, you can select Y. Otherwise type N to end the Python session.

SPECIAL MOVES

1 Castling

Castling is the action of making a special move (no more than once in a game by each player) in which the king is transferred from its original square two squares along the back rank towards a rook on its corner square which is then transferred to the square passed over by the king.

Castling on the king side of the chessboard is called kingside castling while castling on the queen side is called queenside castling.

The moves for castling in this application (provided that the conditions stipulated above are met) are as follows:

Color	Queenside Castling	Kingside Castling
White	**E1 C1**	**E1 G1**
Black	E8 C8	E8 G8

The following is an example of kingside castling for the white king (move E1 G1):

```
AI is thinking....Please wait
Black Knight moved
AI move: (G8) -> (F6)

    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R |   | B | Q | K | B |   | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P |   | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   | N |   |   | N |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   | P |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   | p |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   | n |   | b |
  +---+---+---+---+---+---+---+---+
2 | p | p | p | p | p | p |   | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k |   |   | r |
  +---+---+---+---+---+---+---+---+

Your Move: e1 g1
White King moved
Rook moved 2 tiles to the left
White Castling executed King side

    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R |   | B | Q | K | B |   | R |
  +---+---+---+---+---+---+---+---+
7 | P | P | P |   | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   | N |   |   | N |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   | P |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   | p |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   | n |   | b |
  +---+---+---+---+---+---+---+---+
2 | p | p | p | p | p | p |   | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q |   | r | k |   |
  +---+---+---+---+---+---+---+---+
```

2 En-passant

En passant is a special pawn capture that can only occur immediately after a pawn makes a move of two squares from its starting square, and it could have been captured by an enemy pawn had it advanced only one square. The opponent captures the just-moved pawn "as it passes" through the first square. The result is the same as if the pawn had advanced only one square and the enemy pawn had captured it normally. The following is an example of en-passant:

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R | N | B | Q | K | B |   | R |
  +---+---+---+---+---+---+---+---+
7 |   | P | P |   | P | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 | P |   |   | p |   |   |   | N |
  +---+---+---+---+---+---+---+---+
4 | p |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+

AI is thinking....Please wait
Black Pawn moved
AI move: (E7) -> (E5)

    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R | N | B | Q | K | B |   | R |
  +---+---+---+---+---+---+---+---+
7 |   | P | P |   |   | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 | P |   |   | p | P |   |   | N |
  +---+---+---+---+---+---+---+---+
4 | p |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+

Your Move: d5 e6
White Pawn moved
En-passant completed
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 | R | N | B | Q | K | B |   | R |
  +---+---+---+---+---+---+---+---+
7 |   | P | P |   |   | P | P | P |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   | p |   |   |   |
  +---+---+---+---+---+---+---+---+
5 | P |   |   |   |   |   |   | N |
  +---+---+---+---+---+---+---+---+
4 | p |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   | p |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   | p |   | p | p | p | p | p |
  +---+---+---+---+---+---+---+---+
1 | r | n | b | q | k | b | n | r |
  +---+---+---+---+---+---+---+---+
```

Note that you had to enter D5 E6 as the move as the result is the same as if the black pawn had advanced only one square (to E6) and the white pawn at D5 had captured it normally.

3 **PAWN PROMOTION**

Promotion in chess is a rule that requires a pawn that reaches its eighth rank to be immediately replaced by the player's choice of a queen, knight, rook, or bishop of the same color. The new piece replaces the pawn, as part of the same move. The choice of new piece is not limited to pieces previously captured, thus promotion can result in a player owning, for example, two or more queens despite starting the game with one.

For simplicity, in this application, a pawn that reaches its eighth rank is immediately replaced by a queen of the same color. See the example below of a black Pawn promotion to Queen:

```
Your Move: c3 c2
White King moved
```

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 | R |   |   |   |   | K | P | P |
  +---+---+---+---+---+---+---+---+
6 |   | P |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   | P |   | N |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 | P |   | k |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
```

```
AI is thinking....Please wait
Black Pawn moved
Black Pawn was promoted to QUEEN
AI move: (A2) -> (A1)
```

```
    A   B   C   D   E   F   G   H
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 | R |   |   |   |   | K | P | P |
  +---+---+---+---+---+---+---+---+
6 |   | P |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   | P |   | N |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   | k |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
1 | Q |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
```