CHESS CLASSIC

SPARSH KUMAR JHA

19ME10070

Introduction

We have implemented the traditional chess game for both DOUBLE PLAYER and SINGLE PLAYER using mini-max algorithm and alpha-beta pruning.

Overall design of the program

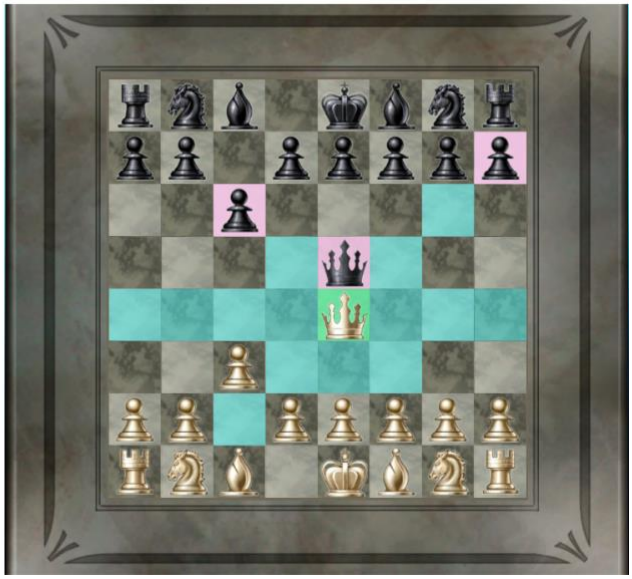
Broadly, the program is divided into three parts :-

1) ***Graphics*** :-We have used 2htdp/image and 2htdp/universe libraries for graphics. It has been abstracted into two parts . All the graphics functions are subset of this:-

a. **to-select** : Once a piece is selected, cells are highlighted into different colours accordingly.

b. **to-move** : In case the user has clicked on a highlighted cell, then the corresponding move is made.

When there is checkmate or stalemate, a message stating that appears on the screen.



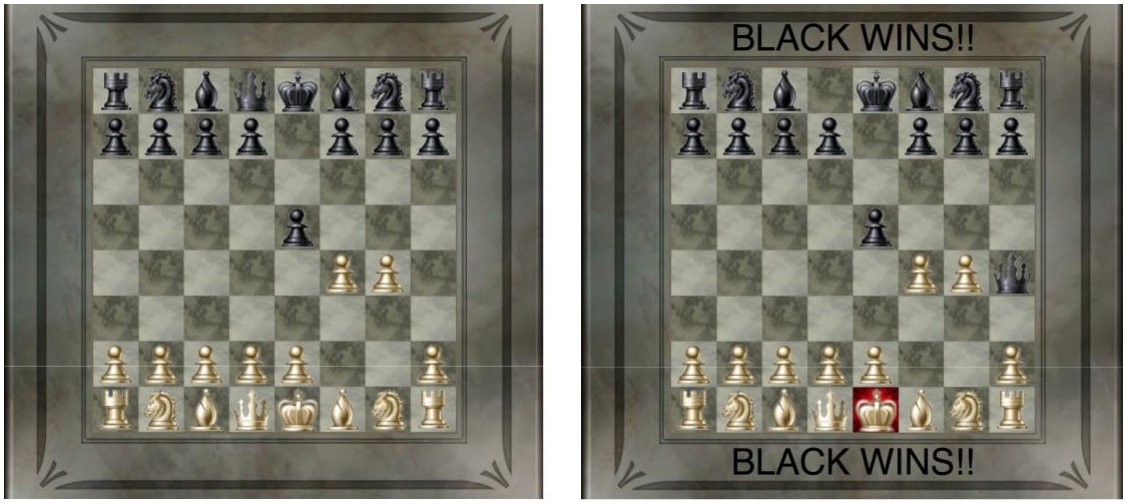
The horse is pinned so it cannot move anywhere.

2) ***Chess setup*** :-This part basically defines valid moves of every piece and all other rules of the game. It consists of the following :-

- **Structures** used for defining pieces, positions and moves. Chessboard has been stored in form of two lists of piece structures.
- **States** and global variables used in cases like castling and draw.
- **valid-moves** function which returns the list of all valid moves of any piece (both capturing and non-capturing) using many filters and higher order list functions like foldr, map.
- Endgame conditions like checkmate and draw

3) ***Artificial Intelligence :-*** This has been implemented using the **mini-max algorithm** and **alpha-beta pruning**. Given a state, our program returns the best possible move at that position (the program thinks upto a certain ply).

The **board evaluator function** evaluates material values of all pieces, positional values of pawn and knight etc.



Checkmate by Computer

Sample input and output

Initially, the choice (one-player, two-player, instructions or credits) to be played should be clicked.

To move a piece, we have to click the moving piece. It highlights all the valid positions of the piece. The clicking one of the blocks moves the piece to that position.

In the one-player format, after moving the piece, it is necessary to press **ENTER** for the computer to play its move.

Any move can be **undone** by pressing **Z**.

To get a **hint**, press ENTER. It automatically executes the best move for the player. If the user feels it to be useful, press ENTER for the computer to play or press Z and make your own move.

Limitations and bugs

1) Pawn can be promoted only to queen.

2) Some evaluator functions like mobility value, rook-queen taxicab distance etc. have been defined but not used as they were taking up too much computation time.

3) Artificial intelligence does not think of castling.

4) We are evaluating only till ply 4 for faster computation (it can be increased by compromising on time).

5) Three-fold repetition draw is not implemented in our program.

6) Due to so much of cases and functions there may be some borderline cases in which our program gives error.