

Chess Titans : A CS 154 Project

Tarun Kathuria
110110028

tarunkathuria@gmail.com

Ranveer Aggarwal
120050020

ranveer@cse.iitb.ac.in

April 21, 2013

1 Introduction

Figure 1: Start Screen



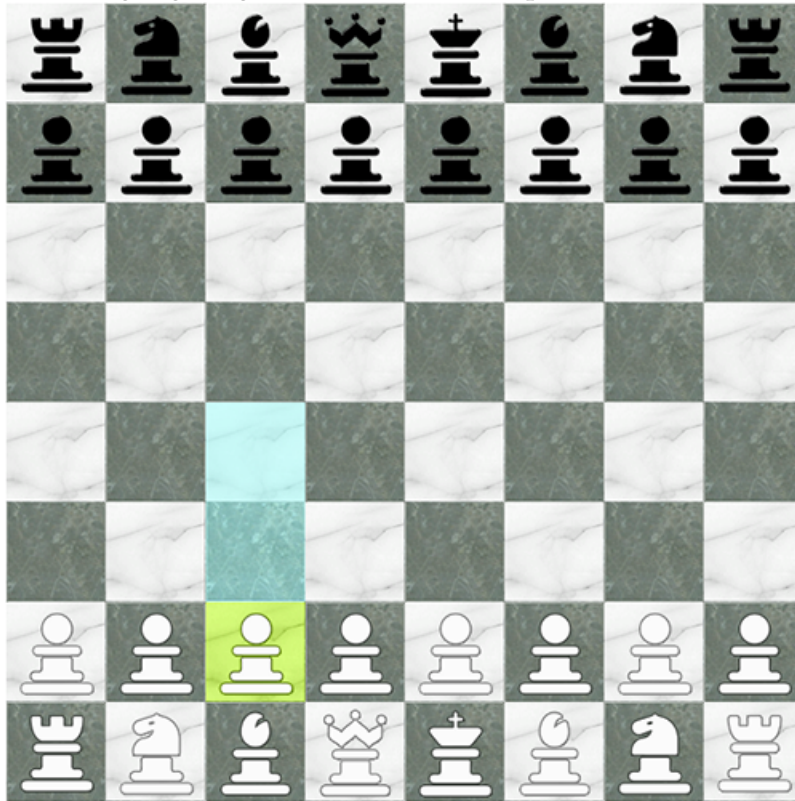
We have implemented the classical game of Chess. The main aim of this project was to learn about higher-order functions and abstractions. This was a very good example of the same since we implemented AI in the form of minimax algorithm with alpha-beta pruning. This served as an intellectually enriching and rewarding experience for us.

2 Design Ideas

2.1 Boards and pieces

We represent the board as a list of lists which are each of size 8. The pieces are represented in terms of their color and piecetype(symbols for various pieces have been used). Various global functions which take the board and the current player(the player who is supposed to make his move right now) and return the needful(e.g. All possible moves the player can make)

Figure 2: Highlighting the valid moves of a pawn on the chess board



2.2 Graphics

We have used the Legacy library for designing the GUI. Buttons have been implemented in the form of image-buttons. The images are displayed and act like buttons when clicked. We spent some time editing various images we found on the Internet to create a visually appealing chess board and chess pieces.

2.3 Artificial Intelligence

We have implemented Minimax algorithm with alpha-beta pruning using a recursive formulation. There are 2 loops inside the main alpha-beta function. One which is used for the max nodes and the other for min nodes.

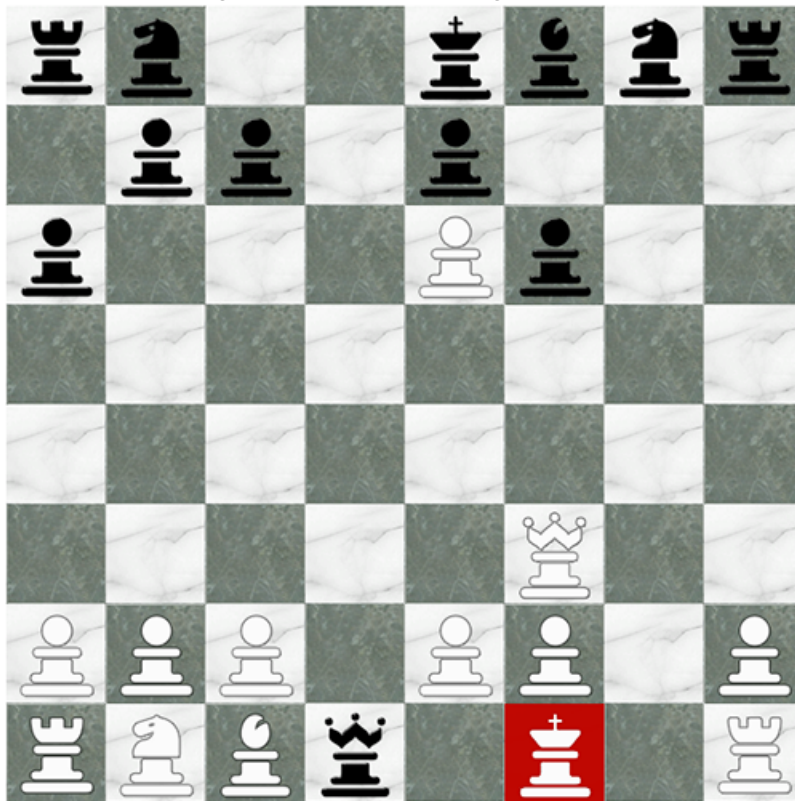
3 Sample Input/Output

The input is given by mouse-clicks. On the first click, if a piece is there and it is the user's turn, all of the valid moves of that piece will be highlighted. (S)he can then choose to click again on whichever square they want that piece to go to. Output is shown on the GUI on the chessboard itself. If a piece can be killed, it's square is highlighted in red indicating the same. A darker red highlights the square of the king if that king is in check.

4 Limitations and bugs

The heuristic board evaluation function is rather simplistic since it's main component is the net material value only. Due to lack of time, we could not implement other heuristics like queen and rook taxicab distance, proximity

Figure 3: The white king in check



of knights to the center of the board, etc.. A slight bug in the program is that the image buttons usually respond only to specific positions on the start screen(usually the upper-left corner)