Introduction to Programming 1
Final Project Report
Shobhit Singh

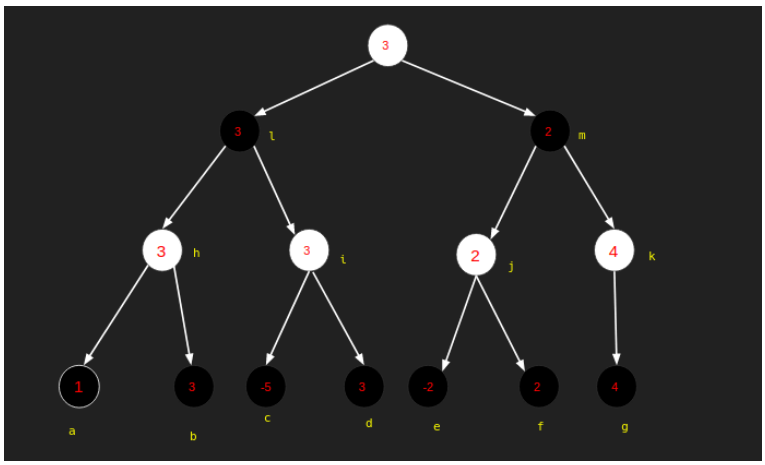
Volkov-Chess

Volkov is a chess engine which tries to calculate the best move in any given position. The only module used in the engine is python-chess .The program is divided into 4 major functionalities.

1. Input : The program takes the FEN and color to play as the input. FEN is standard chess notation to uniquely describe any given position on the board.

2. Material : This function gives the evaluation of the current position. In my program the Queen, Rook, Bishop, Knight and pawn are worth 9,5,3,3 and 1 respectively. And the final evaluation is measured by (Weighted material of white - Weighted material of black). So the black side should try to lower the evaluation as much as possible (which will mean it has more weighted material) and the white side should try to maximise the evaluation.

3. MiniMax : Minimax is a standard algorithm often used in two player zero-sum games. It checks all the possible moves in a position by going down a tree of depth(n). At the depth it runs the material() function on all positions and then chooses and ignores moves based on their evaluation.
   In the example below, a position is given to white and the depth(n=3). For the given position, there are two possible moves for white, and for both the moves there are two possible moves for the black and so on. First the program goes 3 moves down and evaluates the position of moves a,b,c,d,e,f and g. Now as 'h' is white's move, it will try to maximize the evaluation and will chose the move that leads to 3. So the effective evaluation of 'h' is 3. Similarly 'i', 'j' and 'k' will choose the move with more evaluation making their effective evaluation 3,3,2 and 4 respectively. Now it will be black's move, and for black to play the best move, it should minimise the evaluation. So 'l' and 'm' will choose the lesser of possible moves making their effective evaluation 3 and 2 and out of those white will choose the move that leads to 3.



What this means is that the evaluation of white will still be 3 (or greater) even if black plays the best possible plays.

My minmax function takes the position, checks if any side is in checkmate or not (returns -+123456 if they are) and then runs this algorithm on the position to return the final evaluation.
Apart from this I have used alpha beta pruning to speed up min-max. It cuts out the branches in the tree whose evaluation doesn't affect the final outcome.

4. Selecting move : For selecting a move there are three options. I have provided the program with a opening book which contains famous chess openings and counters to it. So if the entered FEN is in that opening book, it will return the right move according to the book. This only works if the game is still under 5-6 moves. In most of the cases though, it will run mixmax() over the list of all possible moves in the position and then save all final evaluations with their corresponding moves in a dictionary. This dictionary is then sorted in ascending order of the evaluations and the first move is returned for black (least evaluation) and the last move is returned for white(maximum possible evaluation).
The third possible case is that if the position is the "Four Knights Opening" (FEN : r1bqkb1r/pppp1ppp/2n2n2/4p3/4P3/2N2N2/PPPP1PPP/R1BQKB1R w KQkq - 4 4) and white has to move. In this case the program explicitly returns 'f3e5'. This is not the strongest move but I think this is pretty interesting so I added this.

How to test :

1. The best way to get chess positions and their FEN's is Lichess. You can make moves from both sides and then generate the FEN and side to play.

2. https://en.alpaso.club/shake/ makes a random legal FEN and its always white to play!