Department of Information Technology
NBA Accredited

A.P. Shah Institute of Technology

G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2021-2022

A Project Presentation on

Chess AI

Submitted in partial fulfillment of the degree of
Bachelor of Engineering(Sem-6)
in
INFORMATION TECHNOLOGY

By
Ayush Jain (18104005)
Anand Morye (19104018)
Neel Dudheliya (19104049)

Under the Guidance of
Prof. Jayshree Jha

1.Project Conception and Initiation

1.1 Objectives

- To build an AI program that can play chess with a rating of at least 1500.
- To build a user-friendly drag and drop UI for easy game interactions.
- To use piece evaluation functions for better position prediction.
- To use and optimize search algorithms like Min-Max & Alpha-Beta Pruning.
- To build an AI program that can follow basic chess rules & regulation including piece movement, legal and illegal moves, castling,etc.
- To build an AI program that can predict move upto the depth of 5.

1.2 Literature Review

In a 1999 paper, Ulrich Schawlbe and Paul Walker discuss Zermelo’s work and contribution to the research area of chess solvability. The main questions he tried to answer were :

When is a player in a “winning” position and can this be defined in a mathematical sense?

If a player is in a winning position, can we determine the number of moves needed to force a win?

1.3 Problem Definition

- Chess is a complex game with abundant amount of possibilities. A computer program must follow the rules of chess while also develop a winning strategy.
- Since the state of the game varies with each move, Keeping a track of states becomes harder.
- Chess is a game with some special moves/rules that a player must follow including :-

    Castling (A King and a Rook manuever).

    En Passant (a special Pawn capture).

    Threefold repitition (for draw).

    Pawn Promotion.

1.4 Scope

- User can train himself in single-player offline mode and test its skill against the AI
- User can make new strategies and test his pre-game strategies against the AI
- User can increase their problem solving skills
- User can improve their concentration by playing chess
- User can increase their cognitive skills
- User can develop ability to think rationally as well as logically

1.5 Technology stack

- Front-end :- HTML, CSS, Javascript
- Back-end :- Flask
- Libraries used :- Python (python-chess)

  Javascript (chessboard.js & chess.js)

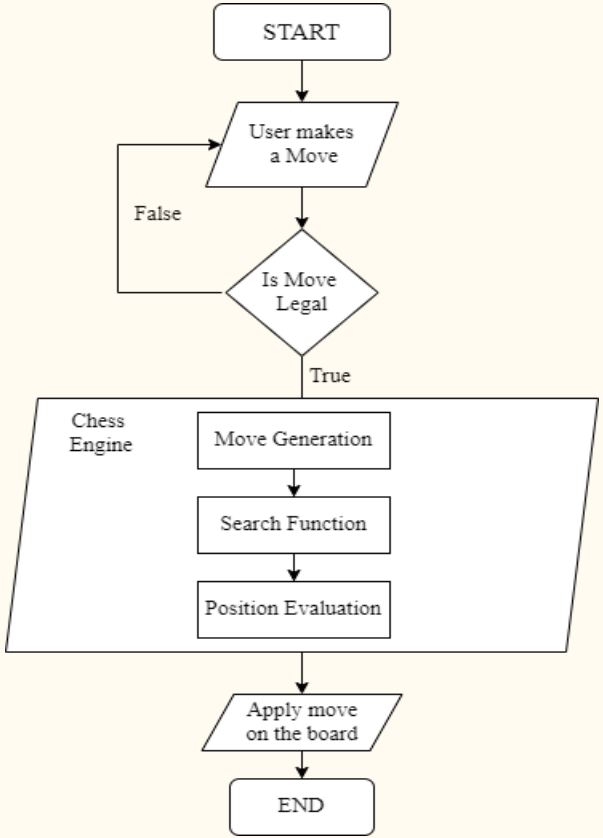
2. Project Design

2.1 Proposed System

- In chess we are using a game tree which is a directed graph whose nodes are positions in the game and edges are moves the player make.
- Game tree will have all required states of a particular chess game.
- For finding the best possible move from the Game tree, Searching algorithms like Min-Max will be used with iterative deepening. To increase the computation speed we will use Alpha-Beta prunning.
- For special rules in chess such as Castling, Pawn promotion, en passant special boolean functions will be use to determine if such rules are applicable or not.

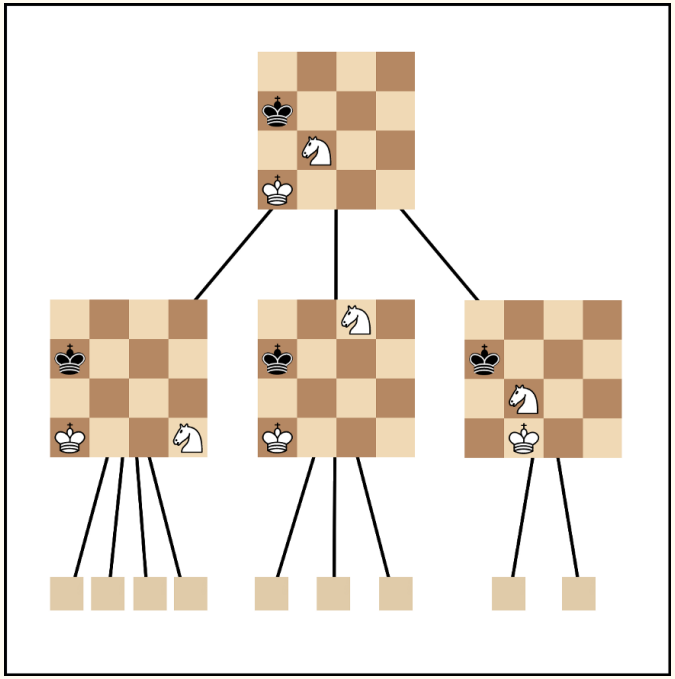
2.2 Design(Flow Of Modules)

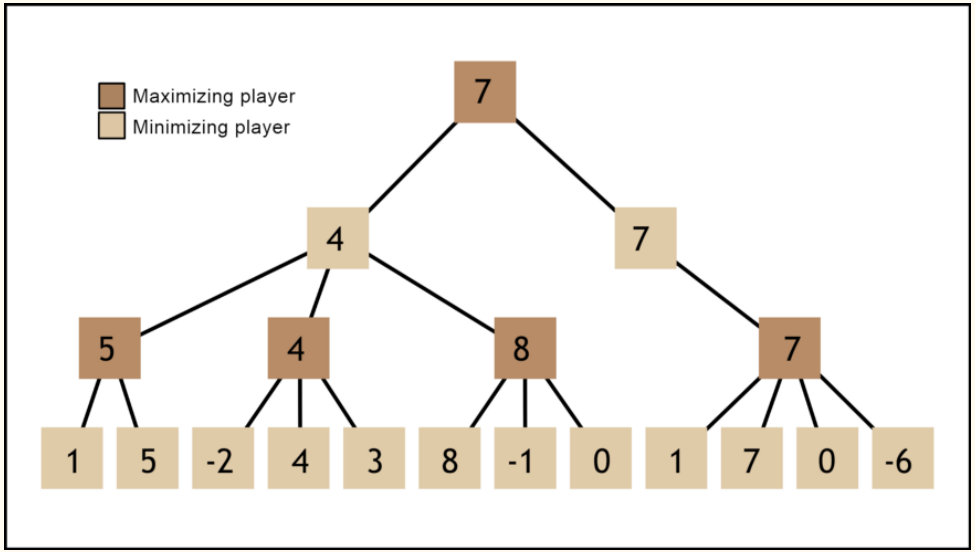
Chess Engine Flow chart:-



2.2 Design(Flow Of Modules)

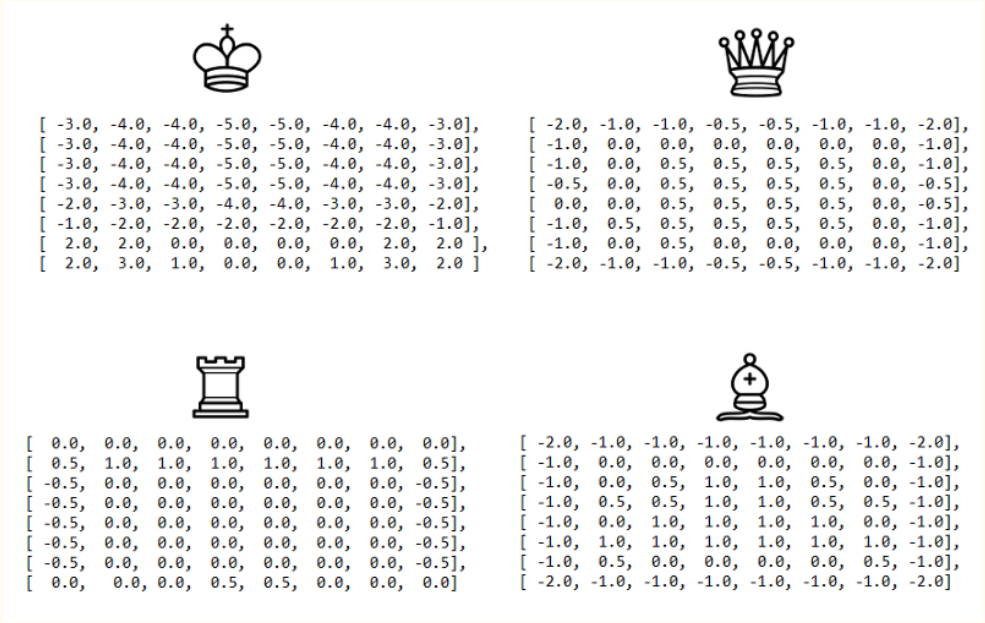
Search function :- Min-Max Algorithms





2.2 Design(Flow Of Modules)

Evaluation function :-



3. Implementation

3.1 GUI (Web App)

White to move

#	White	Black

Take Back | New Game | Depth: 5

4. Result

4.1 Chess Engine (Gameplay)

5. Conclusion and Future Scope

Conclusion & Future Scope

In this project “ Chess AI ” we presented the classic and standard version of chess AI implemented as a GUI. The strength of even a simple chess-playing algorithm is that it doesn’t make stupid mistakes. We've been able to program a chess-playing algorithm that can play chess.

Eventhough we can play chess our algorithm lacks strategic understanding like sacrificing a piece for a future better position. Also restricting the depth of the algorithm would result in “horizon effect”. To overcome this problem we could use Quiescence Searching algorithm to increase the depth we can search for good moves.

References

- Zermelo's Work - https://people.math.harvard.edu/~elkies/FS23j.03/zermelo.pdf
- FreeCodeCamp guide for understanding the workflow of chess engine - https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/
- https://thesmartcoder.dev/build-a-simple-chess-ai-in-javascript/

Thank You