

Chess AI

A cinematic scene featuring five humanoid robots with a steampunk aesthetic, engaged in a chess match. The robots are seated around a large, square wooden chessboard. The robot in the center, facing the viewer, has a white body with a 'V' emblem on its chest and two antennae. It is reaching for a white chess piece. To its left, another robot is partially visible, also reaching for a piece. On the right side of the board, two more robots are positioned, one of which is holding a small, intricate mechanical device. The fifth robot is on the far left, partially cut off. The chessboard is filled with various chess pieces, including pawns, knights, and rooks. The background is dark and industrial, with warm, ambient lighting from above, creating a dramatic and futuristic atmosphere.

By: Michael Atkins, Greg Brinson, Muhammed Owais, and Taeesh Azal Assadi



Problem Statement

We want to create a chess AI that will be powerful enough to not only be fun to play against, but also a good way to learn and improve your chess skills

Use-Case Scenarios

- Gaming and Entertainment
 - Chess playing AI and Chess Training apps
- Health and Cognitive Development
 - Cognitive Training and Therapeutic Purposes
- Social and Community Engagement
 - Online Communities, Chess Clubs and Events



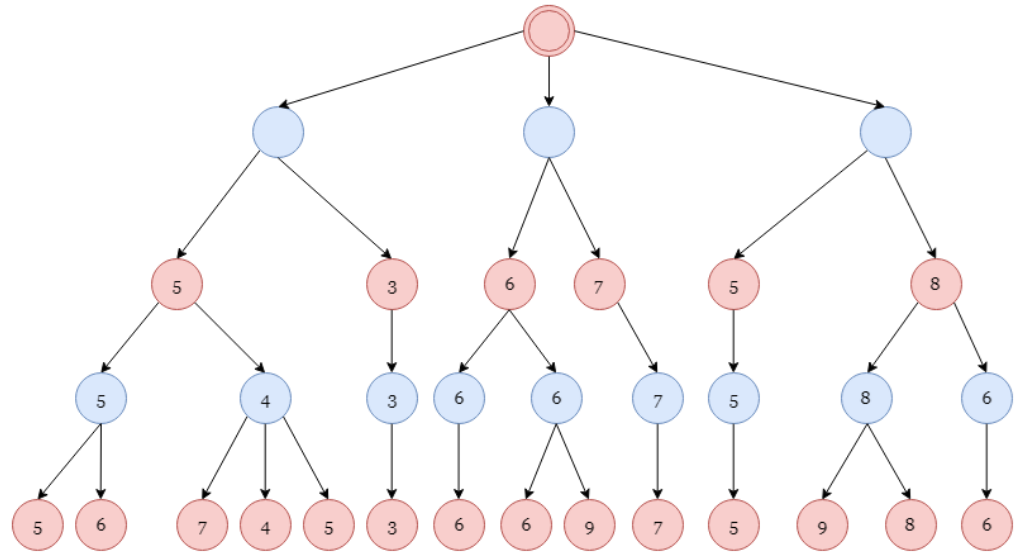


Requirements

- Accurate and powerful
- Fast and performant - people don't like to wait

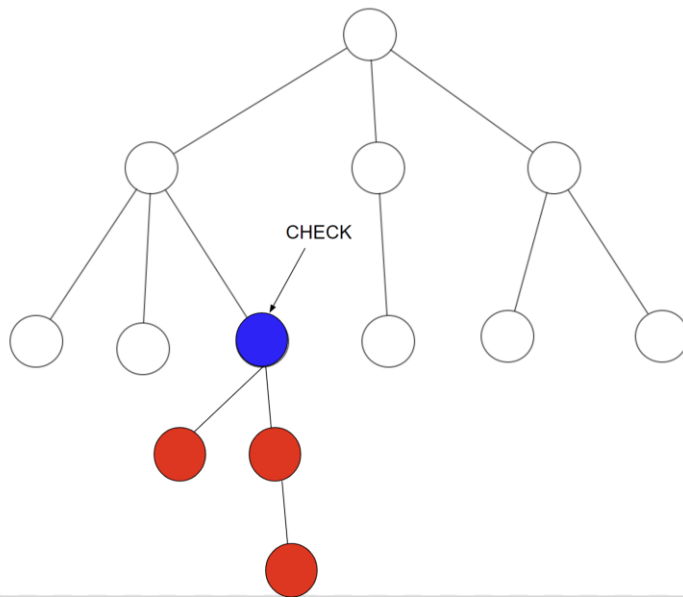
So how do we make it a good chess player?

- Minimax



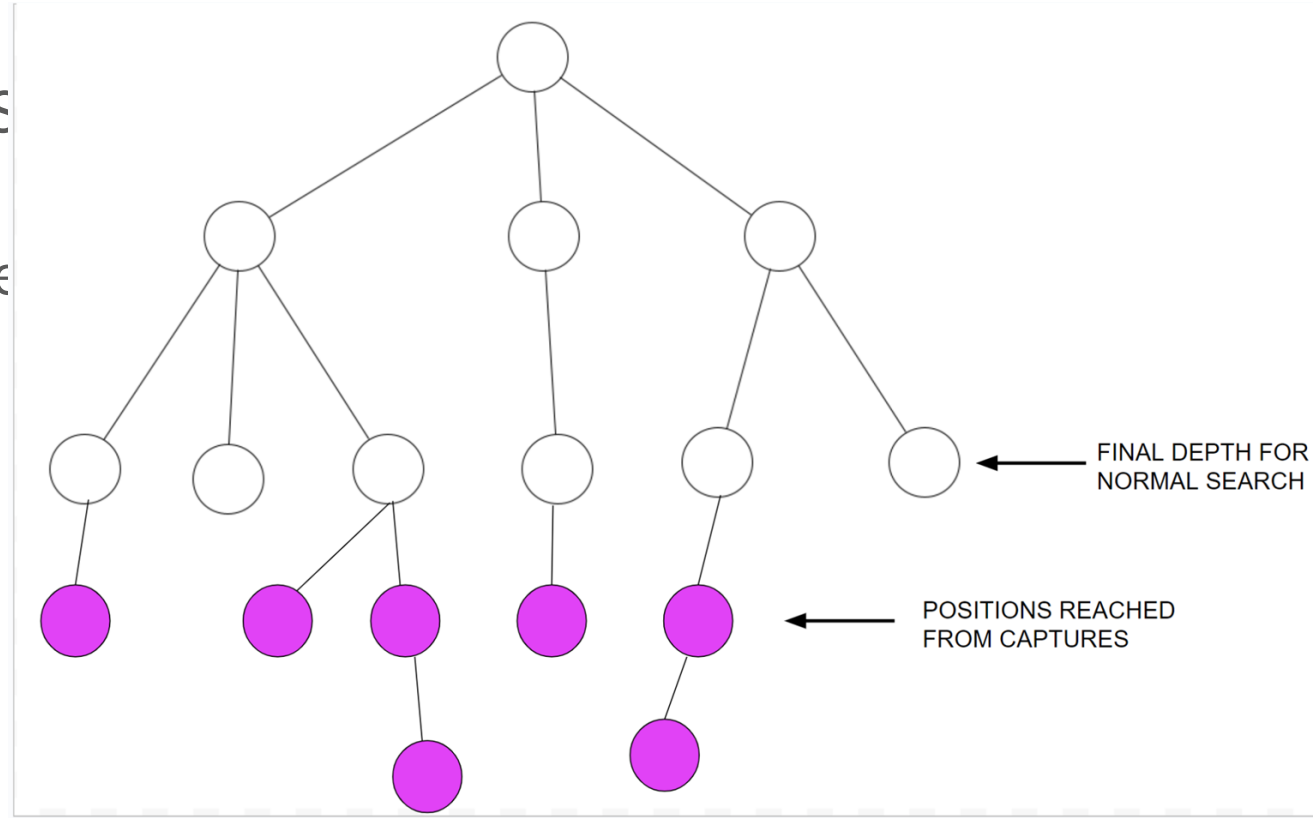
So how do we make it a good chess player?

- Search extensions
 - If we check the opponent, then extend the search depth of that branch



So how do we make it a good chess player?

- Quiescence Search
 - Once we reach a position where we can't make any moves



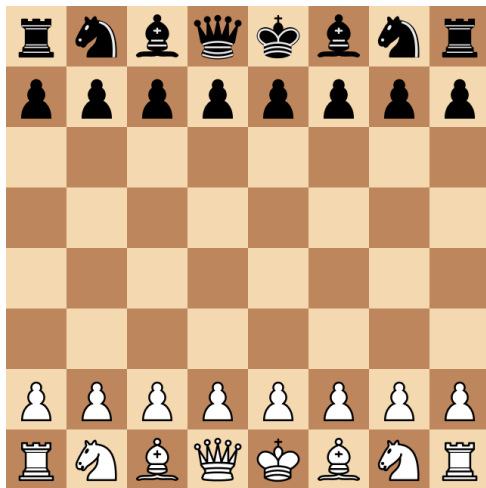
So how do we make it a good chess player?



- Multi-heuristic game state evaluation
 - Material on board
 - King safety
 - Optimal piece locations
 - Castle rights

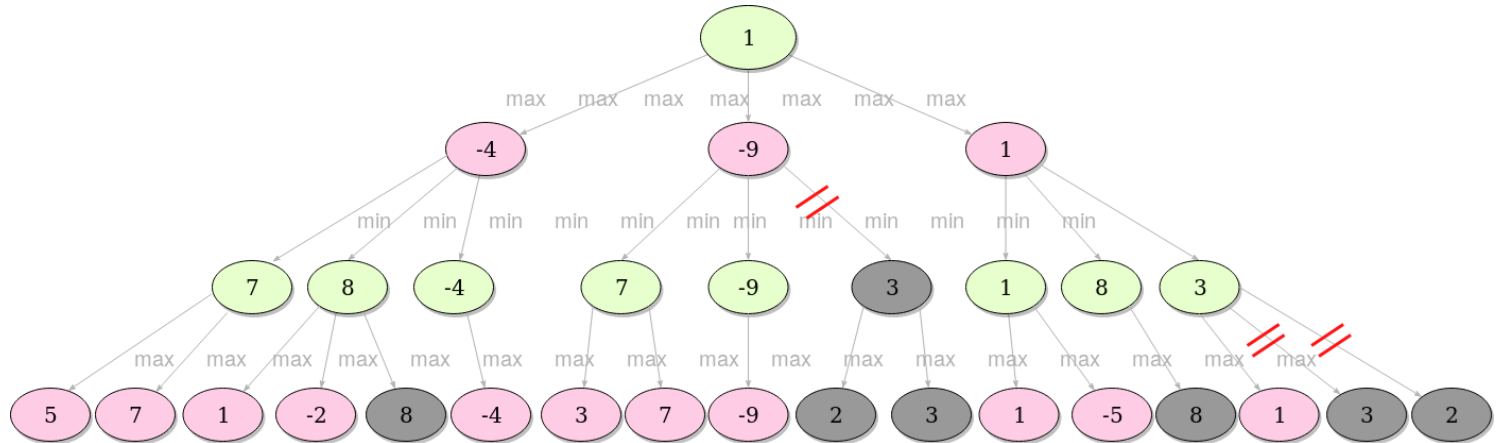
So how do we make it a good chess player?

- Opening book preparation
 - Randomly-weighted opening book knowledge



That's great...but how do we make it fast?

- Alpha-beta pruning



That's great...but how do we make it fast?

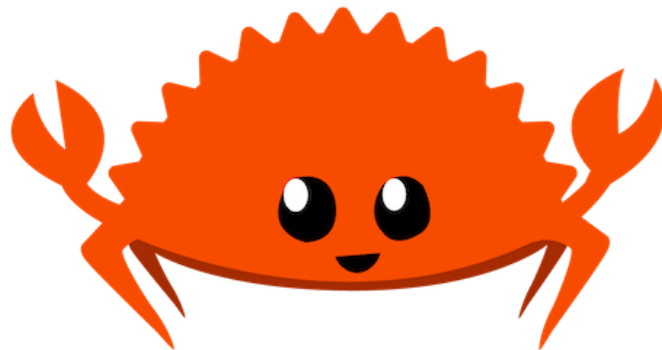
- Move ordering - we prefer:
 - Checks
 - Captures (the better the capture, the better it's ordered)
 - Promotions
 - 'Killer' Moves
 - Early beta cutoff moves are better

That's great...but how do we make it fast?

- Transposition tables
 - Two different sequences of chess moves can result in the same position - so cache the results of each position's evaluation
 - Zobrist hashing
 - Iterative deepening requires we store the depth as well

That's great...but how do we make it fast?

- Rust
 - 'Blazingly fast' language
 - WASM support





Results and Demo

- Rated ~1700 elo
 - This puts it in the 96th percentile of players globally according to chess.com
- Searches to depth of up to 11 ply with search extensions
 - A 'ply' is half a move



Lessons Learned & Future Improvements

- Lots and lots of chess programming resources on the web!
- Move timer
- Improvements to frontend

