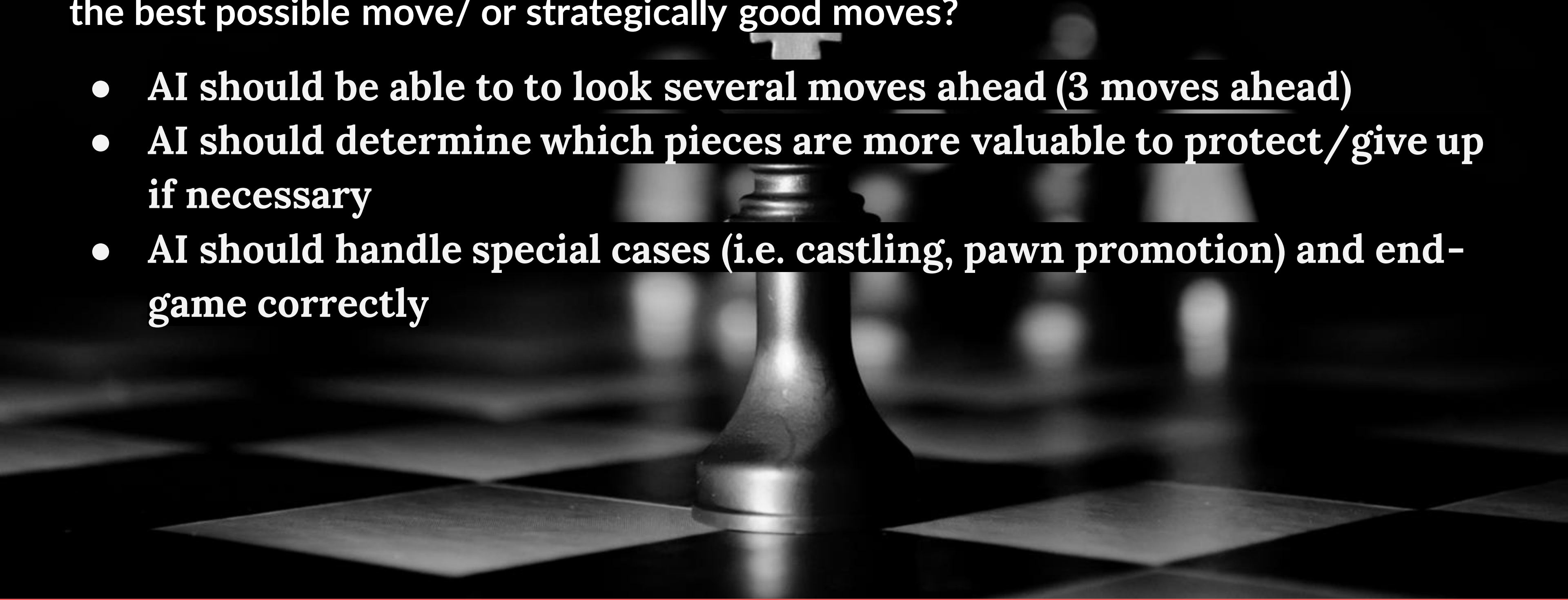
Minimax Chess AI

By: Nicholas Cuttle & Aniket Sonnakul

Problem statement and analysis

Can we implement a chess AI that can effectively interpret user turns and perform the best possible move/ or strategically good moves?

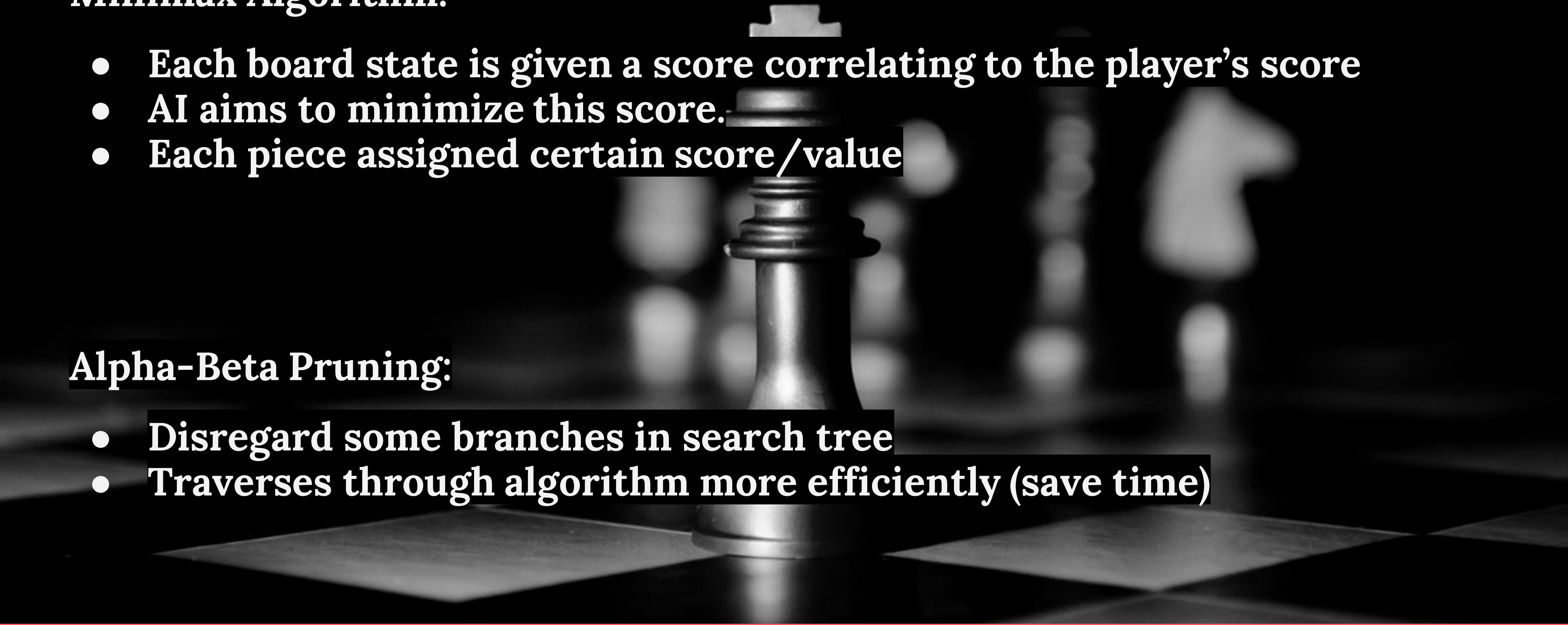
- AI should be able to to look several moves ahead (3 moves ahead)
- AI should determine which pieces are more valuable to protect/give up if necessary
- AI should handle special cases (i.e. castling, pawn promotion) and end-game correctly



Use-Case scenarios

- Help beginner/intermediate chess players improve their moves
- Practice chess without need for an opponent

AI algorithm and model



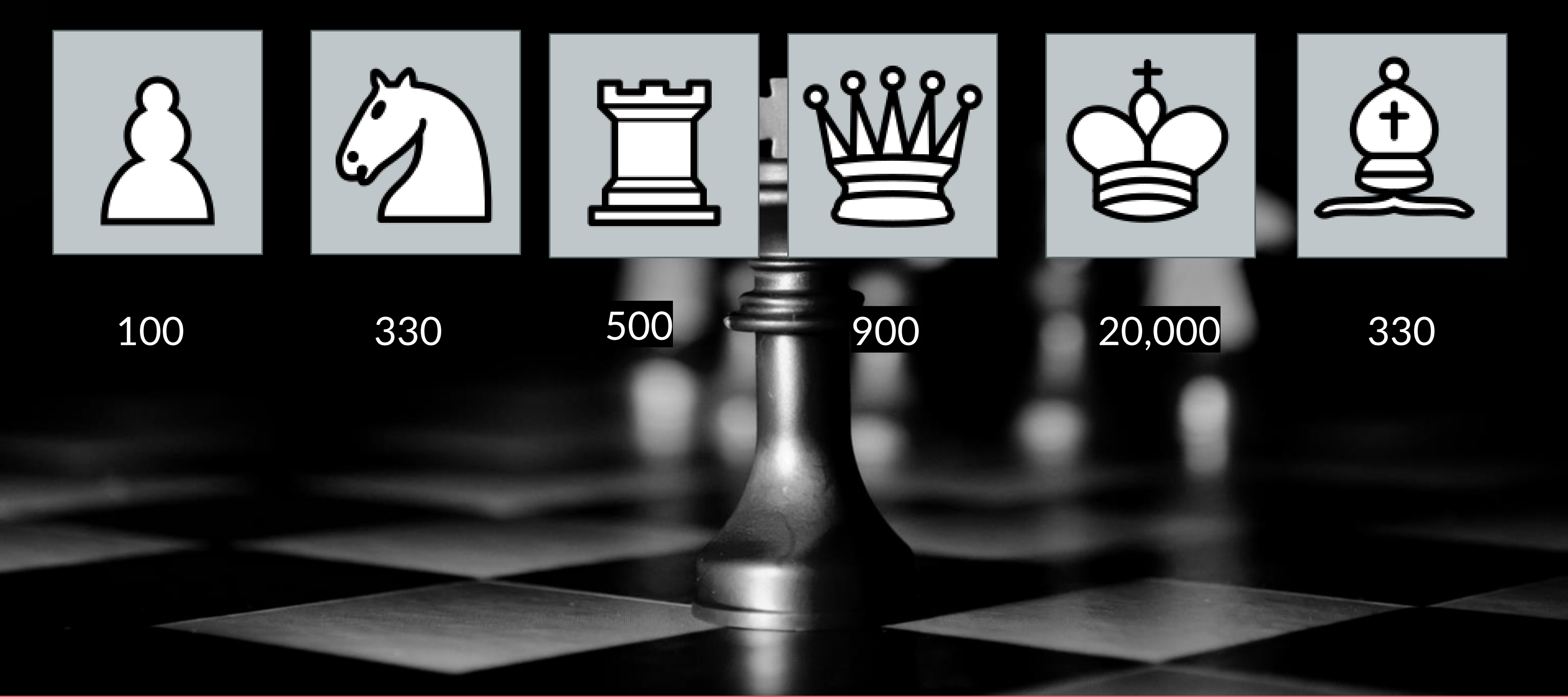
Minimax Algorithm:

- Each board state is given a score correlating to the player’s score
- AI aims to minimize this score.
- Each piece assigned certain score/value

Alpha-Beta Pruning:

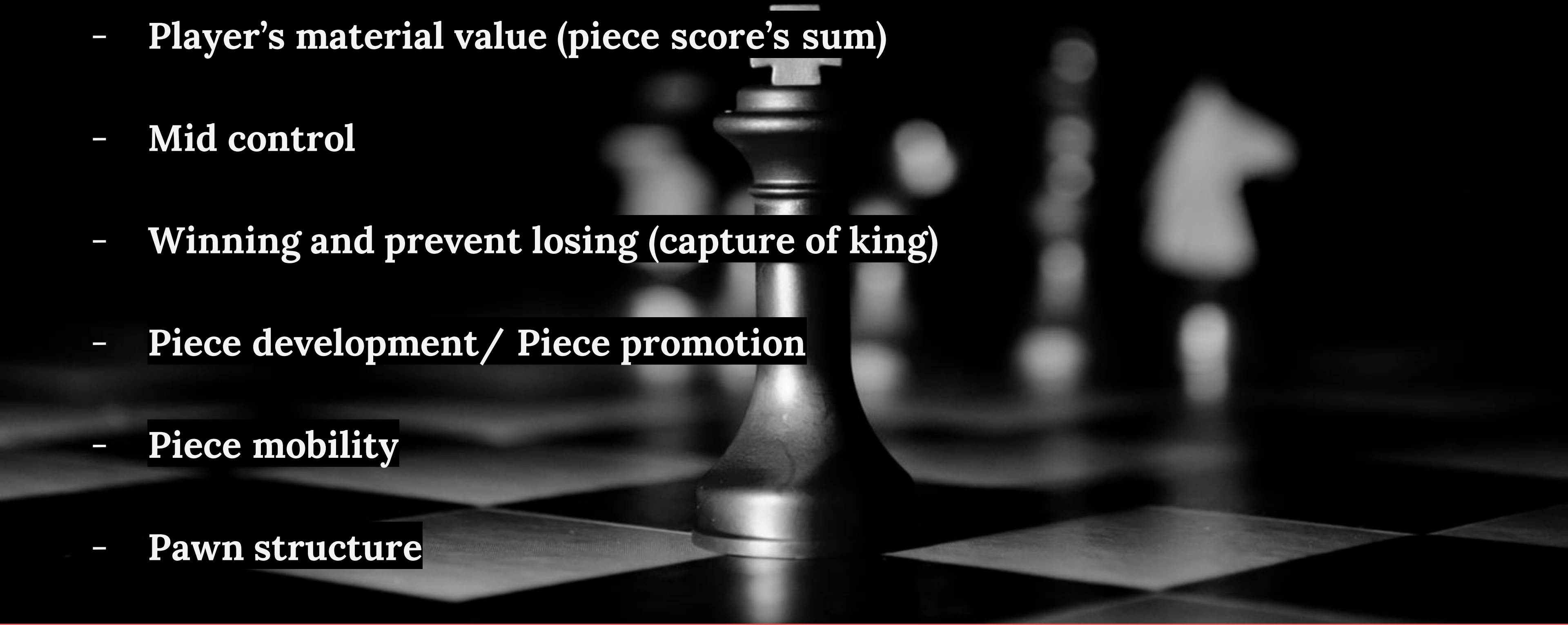
- Disregard some branches in search tree
- Traverses through algorithm more efficiently (save time)

AI algorithm and model: Piece values



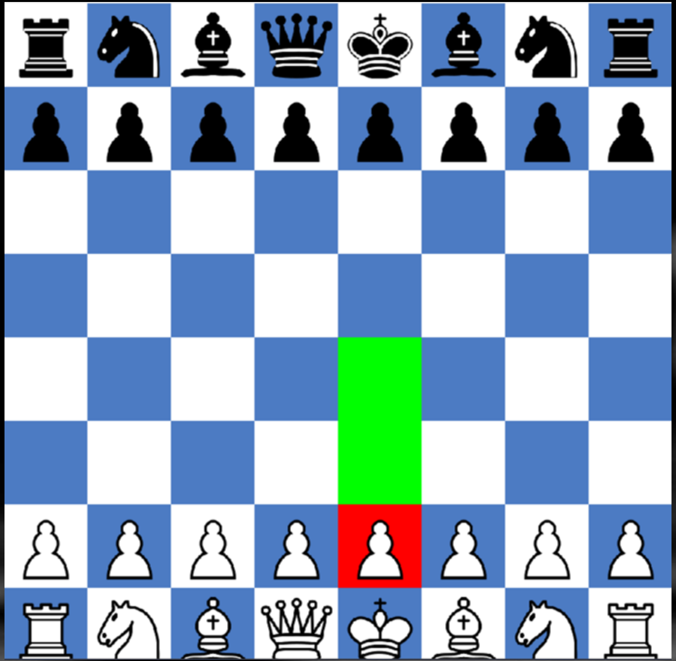
♙	♘	♖	♕	♔	♗
100	330	500	900	20,000	330

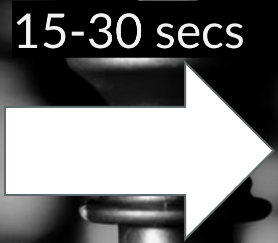
AI algorithm and model: Criteria

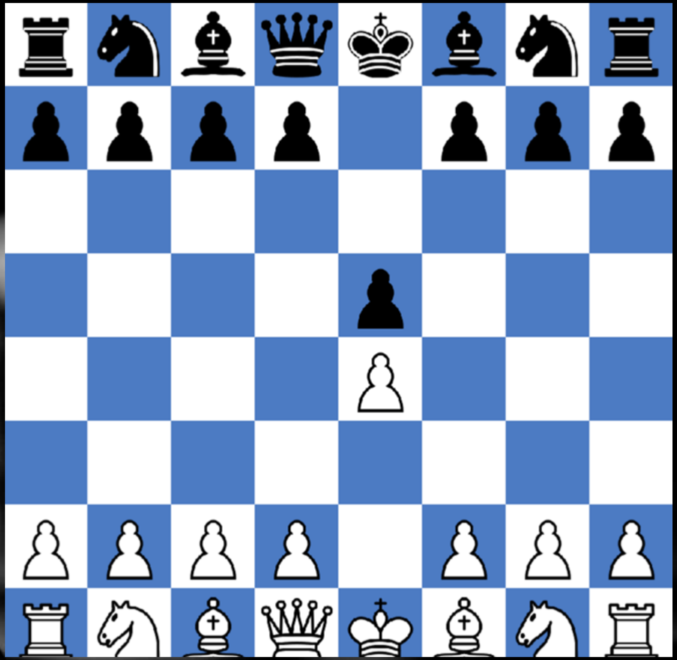


- Player's material value (piece score's sum)
- Mid control
- Winning and prevent losing (capture of king)
- Piece development/ Piece promotion
- Piece mobility
- Pawn structure

Results and demonstration

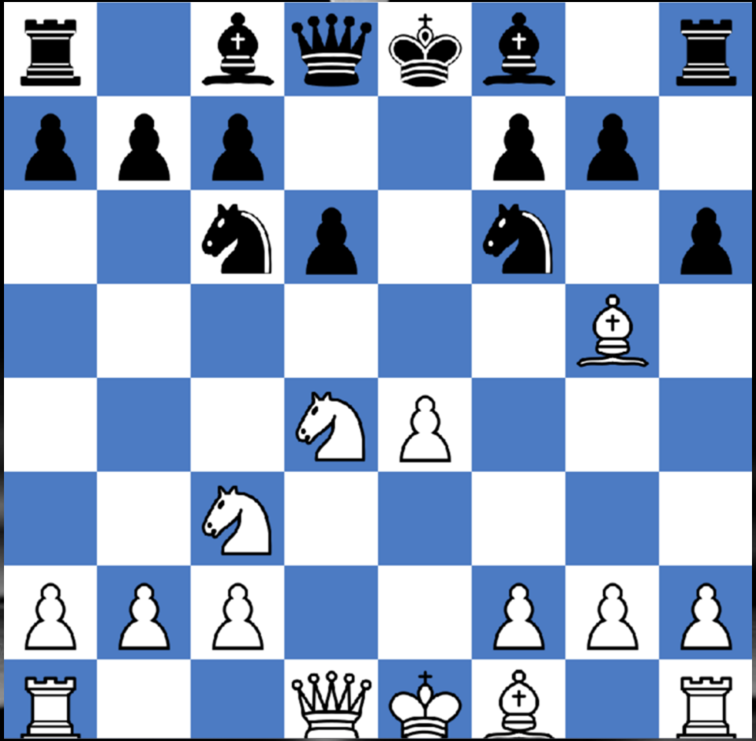






Results and demonstration

Many moves later



Lesson learned

- How to apply algorithms we learned in class to real life applications
- Importance of tweaking weights and criteria to get desired results
- Troubleshooting bad AI moves and modify accordingly.

Q&A

Any Questions?