

Chess Intelligence 2

29 October 2021

11:14

A) How to stock a position ?

- $\alpha 1) \rightarrow$ All pieces still in game White } In order, K, 8p, 2N, 2B, 2R, Q
Black }
- $\alpha 2) \rightarrow [0,7] \times [0,7]$ array for board
- $\alpha 3) \rightarrow 2 \times 2$ array for castles 1 still possible
0 not possible anymore
2 done
- $\alpha 4) \rightarrow$ 1 stock for semi-turn (Who's to play)
- $\alpha 5) \rightarrow$ 1 stock for the "en passant" square

B) How to evaluate a position ?

(Very basic first and optimize in further versions)

- \rightarrow Check ? Ideas , (pawns that gain value when they go further on the board)
- \rightarrow No move ? (castle that is favorised)

Yes w/ check $\rightarrow \pm 1'000$
Yes w/o check $\rightarrow 0$
No (Later verify here repetition and 50 moves rule)

Step 2

- $\rightarrow Q = 9$
- $\rightarrow R = 5$
- $\rightarrow B = 3$
- $\rightarrow N = 3$
- $\rightarrow p = 1$
- $\rightarrow 0-0=0.01$
- $\rightarrow 0-0-0=0.01$ } If they are keep or
0.03 if one is done

C) How does the program work ?

- \rightarrow Set initial parameters A
- \rightarrow Set color of computer and deepness
- \rightarrow While not checkmate
 - \rightarrow Save position in $\alpha 1-4$ in a stock
 - \rightarrow If not computer to play :
 - \rightarrow read the move
 - \rightarrow is the move legal
 - Yes

• Actualise A and visual
• Add halfturn

No

• restart loop not computer
 - \rightarrow If computer to play

↳ • restart loop not computer

→ If computer to play

Is actual position in opening basis

Yes → play associated move.

No → Look all legal moves until deepness
and define best move with the
best-answer-algorithm. then play it

D) Look all legal moves:

→ castles

→ for piece in your-color-pieces

→ try all possible moves for the piece
and look which one is legal

→ between all possibility at a
certain stage, black minimize and
white minimize, then go back until
the move you have to play and play it

F) Additionnal features:

- Terminal board, then visual board
- Card. of moves, then real names of moves