

# CENG461 Homework I

November 4th 2019

Deadline: 18/11/2019 23:55

## 1 Problem

In this homework, you should code a program that plays chess. The UI is already coded for you, you are asked to code the program which takes a scenario (or starting board) and calculates the best possible move. The program should play against the user, first AI makes the move then user and it goes like this.

## 2 Approach

The chess game is based on making decision based on the current board at each step. So you should create a tree based on legal moves at each step, and expand the leaves based on legal moves at each leaf as well. Since this problem is exponentially complex to the number of possible moves, you are asked to search for until depth 5 (in other words, AI only thinks 5 move ahead like "AI-user-AI-user-AI"). There is an algorithm called minimax, which you can use in your program, for search in turn-based decision systems. Also you need a value function to evaluate each moves and what parameters your function depends on is up to you.

## 3 Grading

Your program is expected to solve 2 types of problems. One is finding the checkmate or preventing from it; in this scenario the game is played until some point, then your AI should make a move which is a check-mate (so it must find the checkmate move) or make a move to prevent from getting checkmated. The other one is to making checkmate in 3 moves. So AI plays, then user plays, then AI finds the checkmate. The example situations for 1 and 3 look-ahead scenarios are given and your program will be tested with more of them. However, your program should be able to play from start of the game until the end but you will be graded based on these 2 scenarios. Additionally, there are other grading factors as well, such as Clean code, Proper commenting, Quality of value function, Cheating, Any extra effort to make the program more efficient or accurate.

## 4 Example Scenarios

There are several notations to represent chess boards. In this homework, we use Forsyth-Edwards Notation (FEN). The links are given below to clarify what the characters of notation stands for:

[https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards\\_Notation](https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards_Notation)

<https://www.chessgames.com/fenhelp.html>

### 4.1 Example scenarios

The 1 move ahead scenarios:

Checkmating:

8/5K1k/8/8/8/8/6Q1 w - - 0 1

8/8/8/8/k7/8/1R3K2/1R6 w - - 0 1

1Q6/8/8/2R3K1/8/8/k7 w - - 0 1

Preventing from checkmate:

6k1/r4Qpp/8/8/8/4K3/8/8 b - - 0 1

2k2R2/1b1p4/3B2r1/8/4q3/6p1/ppp5/1k1r4 b - - 0 1

The 3 move ahead of checkmating scenarios:

3Q4/8/8/2R3K1/8/8/k7/8 w - - 0 1

8/8/8/8/7k/4KR2/6R1 w - - 0 1

3K2R1/n3r3/5P1P/8/n7/pp6/kp6/3Q4 b - - 0 1

## 5 Additional Notes

Any code copied from internet is prohibited. You can of course use algorithms from internet or learn techniques and code them, but copy-paste code is prohibited. We are aware of websites that has chess code, so the cheat will be found immediately and we will take an action against any cheater.

There is a library called python-chess, you can use any functionality of it other than its chess engine, its value function, or tree-operation like creating tree over it.

In the UI code given to you, there is ai\_play function which returns the ai's move on its turn. It is a string in the structure of "b1b2" which means the piece on b1 moved on b2. This kind of string should be returned by that function to use UI properly and see AI's moves on board. If any problem occurs because of UI or given code and you can't solve, you can reach me via email and only return the string, don't use the UI.

To use UI as user, If you plan to do castling you should move king to the position that it will be at after castling, rook will be placed automatically. Pawn of user is only promoted to queen, however AI can promote to anything.