Computer Science 5400  
Artificial Intelligence

Spring 2024

Game Assignment Set : Chess

version 24.04.22

****

Chess in a Kilobyte  
[ <https://vole.wtf/kilobytes-gambit> ]

# Introduccion

The game this semester is Chess. The rules and objective for Chess are, except for the here noted **exception**, as contained on Wikipedia’s following page: [<http://en.wikipedia.org/wiki/Rules_of_chess>] The one exception we will use is that instead of the official three board state repetition draw rule, the following simplified logic is used: *If for the last eight moves no capture, promotions, or pawn movement has happened and moves 0,1,2, and 3 are identical to moves 4, 5, 6, and 7 respectively, then a draw has occurred*. Two moves are identical if the starting position (rank and file) and ending position (rank and file) of the moves are identical.

To complete the game project assignments, you need to fill in the make\_move() function of the AI class in the provided **AI framework** template to implement an AI player for Chess in either **C++**, **C#**, **Java**, or **Python3**. Upon each call of the make\_move() function, your AI player shall return a legal move it can find using the search algorithm specified in the assignment. **Never** try to modify the member variables of any of the provided AI framework classes, they should be considered read only. **You need to represent the state of the board using YOUR OWN data-structures**.

Further **documentation** regarding the AI-framework is available at the following link, [<https://github.com/siggame>]

* C++:

[<http://siggame.github.io/Joueur.cpp/namespacecpp__client_1_1chess.html>]

* C# : [<http://siggame.github.io/Joueur.cs/games/Joueur.cs.Games.Chess.html>]
* Python : [<http://siggame.github.io/Joueur.py/chess/index.html>]
* Java : [<http://siggame.github.io/Joueur.java/#>] (Packages > games.chess)

Last year the TAs lovingly prepared a video tutorial on how to approach the framework and documentation: ( *Thanks Mark and Henry!* )  
[<https://drive.google.com/open?id=1GdPD_h3r8iOFCXvAzpbp8s79Ye9itpCT>].   
Do not take this video as a complete or comprehensive guide on the use of the framework; instead, think of it as an initial jumping off point for your further exploration of the framework and accompanying documentation. The process for setting up the local server is also demonstrated.

# Input and Output:

Your program will need to support arbitrary **initial states** in Forsyth-Edwards Notation [<https://en.wikipedia.org/wiki/Forsyth-Edwards_Notation>], so do not assume that when you start a game it will necessarily be in the default initial chess state.

Required output will change according to the assignment but in general you will use **Universal Chess Interface notation** [<https://en.wikipedia.org/wiki/Universal_Chess_Interface>)] to return/output the move chosen by your AI.