GEORGE MASON UNIVERSITY
COMPUTATIONAL & DATA SCIENCES

Project # 5

Chess

Due Date: 12/7/2021

1 A DIFFERENT KIND OF PROJECT

This project is different than the others. There is no

- Q&A.
- Video presentation

There is only a competition. Your grade will be determined by your group's performance which is measured in how well your program can play chess against other teams.

2 EXPLANATION

You will have been given a chess program in class that doesn't perform well. However, it does play a legal game of chess. Your task is to improve the performance of this program.

- You program must be stand-alone. That means it does not access other resources during play. You can not fetch your moves from web sources. You can not import other functions designed to play chess that are written by people outside your team.
- The current program does not castle, promote, or use en passant. You can add these.
- The instructor will write a Master program that manages a game between two teams. Thus, you will need to have a common interface. The master program will call one

function from your program (such as Team1()). The inputs to this function are the current board and 'w' or 'b' signifying your color during the game. The output of your function is your move which are the two coordinate locations (the *from* location and the *to* location. The master program will call **Move** and move your piece.

- Your program must be able to play either Black or White.
- The instructor will set up a tournament schedule. Your program will play other teams. The games will be considered as a Win, Loss, or Draw. You will get 2 points for each win, and 1 point for each Draw.
- The team with the most points gets the A+ (100 points). We have 8 teams. The current plan is that the grades for performance will be 100, 98, 96 ... 86. However, this can be adjusted. A team will get lower grades if their program fails to function, makes illegal moves, or takes too long to make decisions.
- You can have many trials before the tournament - to make sure your program is working with the master program, and to check the performance of your program.

3 INTERFACE

Your program will need to interface with the Master. So you will need a couple of functions with the given names, inputs and outputs.

- PieceValues(): This is the same as the original function, but you can adjust the values of the pieces. Your values will be applied to both sides. The Master will run two boards simultaneously so the teams can play with different values.
- pick = SelectedMove(board, color): The inputs are the playing board and the color of your team: either 'w' or 'b'. The output is same tuple as used in the original program. There are four items in it: the name of the piece, the old location, the new location, and the score. The Master will use only the old location and the new location. The output is a single tuple as you are providing just one move. It is not a list of tuples - just one tuple with four items.
- Option. A third *optional* argument has been added. The variable *pastplays* is the list of plays that have been made. It is an optional argument to the function that you don't have to use. pick = SelectedMove(board, color, listOfPlays):