Othello Lab 4 – 100 Games against Random
AI - Fall 2020

Now that you have an Othello infrastructure in place, you are ready to do something useful with it. In particular, you can imagine your script playing a game against another script by way of a moderator script whose job it is to query each of the two scripts in turn asking them: given this board and a token (corresponding to their side), what move should be made? At this point, the queried script should respond with a preferred move.

For this to be practical, time has to be taken into account. In particular, it is often the case that when you look at a situation (any situation), you might amend your initial impression with a subsequent view. It is this way with game playing, too, in that if your script has more time, it might be able to come up with a better course of action (move to play). Therefore, what constitutes a move declaration is the final integer that a script prints before it either terminates voluntarily or gets cut off.

Othello Lab 4: Play 100 games against Random, coming away with the largest percent of tokens.

In particular, the moderator (the Othello 4 grader) will pit your script against Random in 100 games, where your script will play alternating games as x and o, with the grader accumulating the following two numbers at the end of each game: total number of tokens on your script's side and the total number of tokens on the board. You are looking to maximize this ratio.

It is the grader's responsibility to make each indicated move and it is the grader's responsibility to determine if one side or another must pass and also when the game is over. Your script will always receive a board and a token where a move is possible, and the responsibility of your script is to print a valid move.

Therefore, your script should do the following. It should first determine all possible moves and print them out. That is to say, your Othello 3 scripts should be acceptable to the Othello 4 grader. Of course, that will just establish a baseline for you, where you expect to get to around 50% of all the tokens.

Therefore, after (AND DO NOT LEAVE THIS OUT) your script has printed all possible moves, it will take an informed choice out of the possibilities. It should then do:

```
print(f"My preferred move is: {mvPref}").
```

This informed choice should be based upon a NON-RECURSIVE examination of the board. It turns out that analyzing an Othello board is a non-trivial task, and there is no uniform rule of thumb. Strategies such as the obvious 'make the move which gets you the most number of tokens' may not actually work in your favor. The next page describes some possible avenues. Your script will have .5 seconds per move, not that it should ever get timed out. You may anticipate that a submission takes around four minutes to run. Do not make additional submissions while one is running.

Othello strategies

You may wish to consider implementing some or all of the strategies below as your preferred move, and you may wish to research others.

1) If you can play to a corner, then play to a corner

2) If you can avoid playing next to an unfilled corner, then don't play there

3) If you can play to a safe edge position (one that can't be converted), then do so.

Places where you may read about other strategy ideas include:

http://radagast.se/othello/Help/strategy.html

https://bonaludo.com/2017/01/04/how-to-win-at-othello-part-1-strategy-basics-stable-discs-and-mobility/

https://www.ultraboardgames.com/othello/tips.php

and

http://samsoft.org.uk/reversi/strategy.htm

Because there is a large standard deviation on this lab, your net score will be the average of your last n submissions, where n is a number from 3 to 5. If you have fewer than n submissions, however, it will be the average of your submissions. If you only have a single run, your maximum grade is limited to 90%.