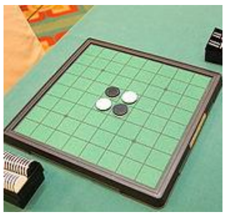
Introduction to AI Spring 2020 Group Game Project Due 6/16/2020



The objective of this assignment is for you to design a game-playing agent. The game to be played is a modified version of Othello/Reversi on an 8x8 square board. If you are not familiar with Othello/Reversi, go check on the rules on the Internet, or play with an App.

However, since the game is very well studied, we will have the following rule changes:

- The board is initially empty.
- ***Black*** moves first.
- The four corners are excluded.
- The legal moves are:
  - Placement of a piece that causes color flipping of some pieces. This is the same as the original Reversi.
  - Placement of a piece in any empty square within the central 6x6 of the board. (As a result, the opening moves will be very different from the standard version.)
- A player is skipped if there is no legal move. However, if legal moves still exist, the player has to take one.
- The game ends when neither player has any legal move.
- At the end, the score of a player is the number of pieces in his/her color minus the number of pieces in the opponent's color. It is possible for the game to end in a tie.
- There is a time limit for each move, so your program needs to be able to keep track the time used while "thinking".

In the tournament, each pair of players will play two games, with each player being ***Black*** once. The "tournament points" are awarded as: two points for the winner, zero point for the loser, and one point each if the game ends in a tie. The overall ranking is based on the total tournament points of the players. The tie-breakers, in case multiple players have the same tournament points, are

(1) Number of games actually won.

(2) Total score differential (a player's total scores minus the opponents' total scores).

You have a lot of flexibility in designing your game agent. It can be as simple as a set of rules. You can try the classical method of minimax search, possibly with alpha-beta pruning. You can also try to implement MCTS, or to train your agent using reinforcement learning. The requirement is that you need to implement the algorithms yourself; you can not use modules/libraries developed by others for game playing. When the TAs run the tournament, the game server and both player programs will run on the same computer. There will be no outside connectivity and no GPU support.

Regarding implementation:

- A team ID will be assigned to each group. You need to include this ID in both your filename and within the code.
- You can only implement the program in C++ or Python 3. The environments will be posted by the TAs.
- The communication between the game server and your program, running as a client, is via TCP. The TAs will provide instructions and sample codes on including the communication capabilities in your program.
- The TAs will provide a sample server program for you to use during the development of your program.

The submission is to be through E3. No late submission is accepted for this project. You should submit your program source code (all in a single file if possible) and a report file separately. The report (maximum 5 pages single-spaced) should describe how your game AI works, and your reasons. The TAs will announce later whether you need to submit executables. Only one submission required for each group. Remember to list the group name, ID, and members in both the source code and the report.