Othello Game Interface Specification

November 11, 2009

As you know, we will be holding a bracket style tournament of AIs sometime towards the end of the semester at a date TBA. It’s important to follow these specifications as failure to do so may eliminate your AI from the competition.

You need to provide a module with a class called OthelloPlayer that has:

1. A constructor OthelloPlayer() which takes no arguments.

2. A method called getName() that returns the name of your program. We will use that name to refer to your program.

3. A method makeMove(board, color).

   The argument board is the current game state. The board is represented as a list of 64 integers. 0 indicates an empty square, 1 a white game piece and 2 a black game piece. For example the starting board configuration is represented by the following list:

   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

   The argument color is the color of the piece your program needs to play, 1 for white, 2 for black.

   The makeMove method should return an integer which is the index of the square where your AI wants to place a game piece (and yes, we will be checking if the move is valid).

   You may store any other data required by your AI in the OthelloPlayer class which will be instantiated once at the beginning of the match. Any persistent variables/data structures your AI requires should be stored as members of the OthelloPlayer class.

   To help you make sure your code works with our testing framework, we will provide the code we will use to pair off your AIs in the tournament. This code will be made available on the wiki (based on code by Matt — thanks!).