CSC 242 Project 1 Reversi by Xinyu Cai and Ningyuan Xiong

Board

We deliberately add "*" to our board in order to let our user see all next available moves. "*" represents for all available moves for next player.

1. An agent that plays randomly

We have a random function in Minimax.java and call it in minimaxDecision where option=1.

2. An agent that uses Minimax

Our Minimax algorithm is mainly implemented in Minimax class. We followed the textbook's pseudocode for Minimax algorithm. We first called minimaxDecision function in PlayWhite/PlayBlack function, and minimaxDecision called minimaxValue. The minimaxValue function call itself recursively until it reaches the leaf node. We give different ending conditions for recursion based on the opponent that our user chose. If option=2, we will return utility function for our leaf node.

3. An agent that uses MINIMAX with alpha-beta pruning

Similar with the option 2, the leaf node evaluation also returns utility function. For finding value for each MIN or MAX level, we use alpha-beta pruning in minimaxValue function.

4. An agent that uses Heuristic Minimax with a fixed depth cutoff and alpha beta pruning

We created an 2D array that represents the value of each tile. Our heuristic function calculated my score minus opponent score and add this particular value which we thought this is the best heuristic function we could thought of. Since we give different ending conditions for recursion based on the opponent that our user chose, if our user chose option 4, then it will use alpha-beta pruning to find best value in MinimaxValue function. We made a comment beside this. Besides, we set our depth to 10. If our user chose option 4, we will return heuristic function for the end of recursion.