Raymond Lee

Reversi AI Project

ECE 469: Artificial Intelligence

Prof Carl Sable

Starting the Game:

On the command-line (terminal) you must type:

javac Othello.java // Compile

java Othello //Run Game

rm \*.class //Clean .class files

Implementation:

The program is divided into 5 classes. The bulk of the program is inside Othello.java. There lies the main method of the program and the Othello class that drives the game. Inside the Othello class, I implemented the driver for the GUI using Swing (interface) and even listeners to detect clicks. The Game class object holds the game board itself. In addition the Game class methods legalMove and pointMove checks if the move is legal and assigns a heuristic value respectively. Each move a player makes becomes a Move object derived from the Move class. Implementing a move this way becomes very convenient for searching for future moves because each Move object holds variables: legal <bool> (check if said move is legal), points <int> (heuristic value of the move), x, y <int> (board position). When implementing my alpha-beta search, I would need to build a tree of nodes. Each node is made by the Node object from the Node class. Each Node contains lastMove <Move> (last move made before making node), state <Game> (state of game at that point in time), turn <int> (turn of player who made/will make said move).

As for the AI, I implemented Mini-Max Search with Alpha-Beta-Pruning. Using this, the program plays very well. It does not only give up corners randomly, but actually pushes you into a position to give it a corner. Unfortunately, I was not able to successfully implement the Iterative-Deepening-Search. I was able to get it working but it would not obey the time limit. I was not able to suppress the program from doing one extra search when the elapsed search time. In addition, the program without Iterative-Deepening-Search played much better than the program with IDS. For the sake of performance, I decided to go with the program that didn’t implemented it. Ali Rahman (EE ’20) contributed in the process of debugging my IDS program but there was no such luck. Although you will not be able to set the search time, you may set the max depth search limit for the program to play with. All in all, I was satisfied in my program’s performance and wish you luck against it!!

Notes:

* If you would like to load in a pre-made board, you must adhere to the sample board inside sampleboard.txt.
* Max search depth limit (when prompted) should not exceed more than 10, my computer could not play the game at all. \*\*Take note when doing this in the pre-made board.
* When the computer runs out of available moves, the game moves onto the next turn and will prompt you for your move.
* When you run out of available moves, click anywhere on the board to move onto the computer’s turn.
* When playing computer v. computer, press start to begin game and stop to stop.