Cmpt 310 Project: Reversi with Monte-Carlo Tree Search

Jia Ming Wu 301278354

Zihao Zhao 301297708

The project we chose is Reversi with MCT search and we will be implementing this program in C++. Our program will use two heuristic algorithms to play against each other to compare their differences. The black pieces which are represented as “X” will use the Monte-Carlo Tree search and the white pieces which are represented as “O” will use a simple heuristic.

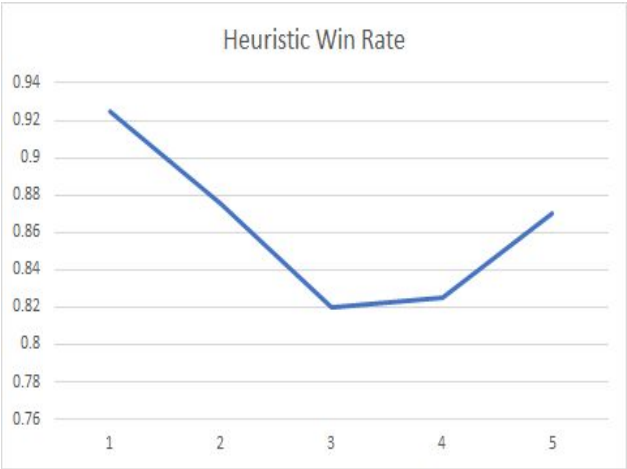
The Monte-Carlo Tree Search in our implementation will select all possible moves and perform a random sampling of game states. Each game state will play out until the end and record a score if they win. After all simulations of the possible moves, the function will choose the highest score’s move as the actual move. The simple heuristic in our implementation will use the current board states and check which move will flip the most opposing pieces, this heuristic will take less memory and it is much faster in comparison to the MCTS.

After running the game, We will ask the user to input how many rounds they would like for the bots to play. After each bot placed their move, a new graph will be shown.

Here are some statistics from the game results:

(W: heuristic Wins ; L: heuristic Lose ; D: Draw)

	10(rounds)	20(rounds)	30(rounds)	40(rounds)	50(rounds)
Attempt 1	W:8 L:2 D:0	W:20 L:0 D:0	W:28 L:2 D:0	W:33 L:5 D:2	W:41 L:6 D:3
Attempt 2	W:10 L:0 D:0	W:17 L:2 D:1	W:23 L:3 D:4	W:35 L:5 D:0	W:44 L:5 D:1
Attempt 3	W:9 L:1 D:0	W:15 L:5 D:0	W:22 L:8 D:0	W:31 L:9 D:0	W:47 L:3 D:0
Attempt 4	W:10 L:0 D:0	W:18 L:2 D:0	W:26 L:3 D:1	W:33 L:3 D:4	W:42 L:6 D:2
Average heuristic wins rate:	92%	87.5%	82.5%	82%	87%
Average Monte-Carlo wins rate:	7.5%	11.25%	13%	13.75%	10%



According to the statistics shown above, it is easy to see that the simple heuristics have a win rate around 86% and MCT Search have a win rate around 11%. In conclusion, the heuristic performs much better than MCTS in memory, speed and win rate.