**Heuristic analysis**

Adversarial game playing Agent for isolation

**Introduction**

This assignment aims to try and improve heuristic values for playing the game of isolation.

Game of isolation is a two player abstract strategy board game. It is played on a 7x7 board which is initially filled with squares, except at the starting positions of the pieces. Both players have one piece; it is in the middle position of the row closest to his/her side of the board. This assignment uses the version of isolation where the player moves in an L shape move like the knight from chess game.

**Evaluating heuristics**

The tournament opponents are listed below:

* Random: An agent that randomly chooses a move each turn.
* MM\_Null: CustomPlayer agent using fixed-depth minimax search and the null\_score heuristic
* MM\_Open: CustomPlayer agent using fixed-depth minimax search and the open\_move\_score heuristic
* MM\_Improved: CustomPlayer agent using fixed-depth minimax search and the improved\_score heuristic
* AB\_Null: CustomPlayer agent using fixed-depth alpha-beta search and the null\_score heuristic
* AB\_Open: CustomPlayer agent using fixed-depth alpha-beta search and the open\_move\_score heuristic
* AB\_Improved: CustomPlayer agent using fixed-depth alpha-beta search and the improved\_score heuristic
* Athar: customPlayer agent using alpha-beta search iterative and my\_heuristic function

def my\_heuristic(game, player):

if game.is\_loser(player):

return float("-inf")

if game.is\_winner(player):

return float("inf")

my\_moves = len(game.get\_legal\_moves(player))

opponent\_moves = len(game.get\_legal\_moves(game.get\_opponent(player)))

return my\_moves \* my\_moves - 1.5 \* opponent\_moves

Using the above heuristic function I got the highest score as 70.71%. Of course different trials different scoring but this is the highest result I get using 5 matches against each opponent.

My heuristic function combine between maximizing the player moves and weighted minimization of opponent moves at the same time.

By changing the number of matches to 5, due to slow execution in our devices, I’ve got the following result comparing with built in agents and my group members

**Results**

|  |  |  |
| --- | --- | --- |
| **Agent** | **Performance** | **Rank** |
| **ID\_Improved** | 62.86% | 7 |
| **Student1** | 63.57% | 6 |
| **Student2** | 58.57% | 9 |
| **Student3** | 65.00% | 3 |
| **Student4** | 61.43% | 8 |
| **Student5** | 58.57% | 9 |
| **Student6** | 64.29% | 4 |
| **Student7** | 65.00% | 3 |
| **Athar** | 70.71% | 1 |
| **Kah Hui** | 63.57% | 6 |
| **Amrita** | 68.57% | 2 |
| **Dong** | 64.00% | 5 |

**Appendix**







