OCT.2017

Udacity AIND

Raquel Redó

# INTRODUCTION

In this project, we are asked to develop an adversarial search agent to play de game "Isolation". Isolation is a deterministic, 2-player game of perfect information in which the players alternate turns moving a single piece from one cell to another on a board. Rules are:

- Board is 7x7 grid
- Whenever either player occupies a cell, that cell becomes blocked for the remainder of the game
- This first player with no remaining legal moves loses, and the opponent is declared the winner.

Also, Udacity adds the following rules

- Agents will have a fixed time each turn to search for the best move and respond. If the time limits expire during a player's turn, that player forfeits the match, and the opponent wins.
- The agents can move to any open cell on the board that is 2-rows and 1-column or 2-columns and 1-row away from their current position on the board.

# HEURISTIC ANALYSIS

The goal is to develop a heuristic such that the coded agent (named Student) outperforms a base_line agent called *ID improved,* that uses Iterative Deepening and the improved_score heuristic defined in sample_players.py provided.

The tournament.py script is used to evaluate the effectiveness of heuristic.

All the heuristics have been coded considering the relation between the possible legal moves for the two players (my player and the opponent players). Assuming that my player has to maximise the possible moves vs the opponents.

## HEURISTIC 1

The first heuristic is going to be squared difference between the number of the possible moves for each player.

$$Heuristic\ 1 = len(my\ possible\ moves)^2 - len(opponent's\ possible\ moves)^2$$

## HEURISTIC 2

The second heuristic is going to be the difference between my movements and the opponents. However, I am going to weight the opponent's to 3 giving much more value to movements which are more distant. The weight has been set iterating some values.

$$Heuristic\ 2 = \text{len}(my\ moves) - 3 * \text{len}(opponent's\ possible\ moves)$$

## HEURISTIC 3

The third heuristic is going to be the Manhattan distance to the distance to the centre of the game.

$$Heuristic\ 3 = \sum_{i=1}^{n} Absolute(my\ distance\ to\ centre - opponent's\ distance\ to\ centre_i)$$

## EVALUATION OF HEURISTICS

The script provided tournament.py is going to be used to measure the effectiveness of my custom heuristics. The script measures the relative performance of my *Student* agent a round-robin tournament against several other pre-defined agents. The *Student* agent uses time-limited Iterative Deepening along with my customs heuristics.

The tournament opponents are listed below

- **Random**: An agent that randomly chooses a move each turn.

- **MM_Open**: MinimaxPlayer agent using the open_move_score heuristic with search depth 3

- **MM_Center**: MinimaxPlayer agent using the center_score heuristic with search depth 3

- **MM_Improved**: MinimaxPlayer agent using the improved_score heuristic with search depth 3

- **AB_Open**: AlphaBetaPlayer using iterative deepening alpha-beta search and the open_move_score heuristic

- **AB_Center**: AlphaBetaPlayer using iterative deepening alpha-beta search and the center_score heuristic

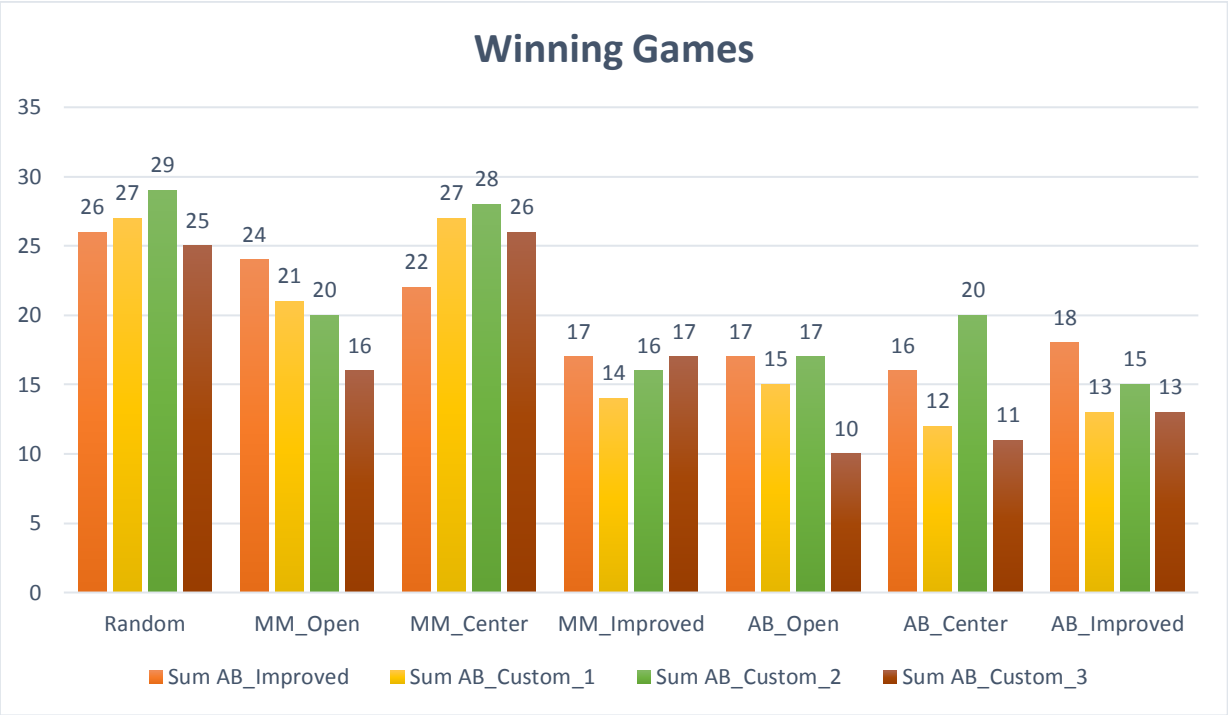
- **AB_Improved**: AlphaBetaPlayer using iterative deepening alpha-beta search and the improved_score heuristic

## RESULTS

I did run the heuristics in 3 tournaments and summarised the game results in the following table:

| Opponent | Sum AB_Improved | | Sum AB_Custom_1 | | Sum AB_Custom_2 | | Sum AB_Custom_3 | |
|---|---|---|---|---|---|---|---|---|
| | Won | Lost | Won | Lost | Won | Lost | Won | Lost |
| Random | 26 | 4 | 27 | 3 | 29 | 1 | 25 | 5 |
| MM_Open | 24 | 6 | 21 | 9 | 20 | 10 | 16 | 14 |
| MM_Center | 22 | 8 | 27 | 3 | 28 | 2 | 26 | 4 |
| MM_Improved | 17 | 13 | 14 | 16 | 16 | 14 | 17 | 13 |
| AB_Open | 17 | 13 | 15 | 15 | 17 | 13 | 10 | 20 |
| AB_Center | 16 | 14 | 12 | 18 | 20 | 10 | 11 | 19 |
| AB_Improved | 18 | 12 | 13 | 17 | 15 | 15 | 13 | 17 |
| Total general | 140 | 70 | 129 | 81 | 145 | 65 | 118 | 92 |
| %(win rate) | 66.67% | | 61.43% | | 69.05% | | 56.19% | |

And with plot:




My second heuristic (weighted difference on the number of movements) got better results on winning rate. Winning rate is 69% beating the AB_improved (66,7%) in a bit more than two percentual points. Moreover, it hits the other heuristics in nearly all kind of heuristic opponent (4 out of 7).

The second best performance is for heuristic AB_Improved which ranks better also in 4 out of 7 opponents but with less winning games.

## CONCLUSION

The heuristic chosen will be heuristic 2:

- It is a simple calculation and its definition it is plain and easy to interpret.
- It beats the majority of opponents (4 out of 7).
- It does not need to use another in-memory object. Therefore, the computation time is small.

## APPENDIX

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      7   |  3     9   |  1     9   |  1     10  |  0
   2       MM_Open     6   |  4     7   |  3     6   |  4     6   |  4
   3      MM_Center    7   |  3     10  |  0     9   |  1     8   |  2
   4     MM_Improved   6   |  4     6   |  4     6   |  4     7   |  3
   5       AB_Open     7   |  3     5   |  5     5   |  5     4   |  6
   6      AB_Center    5   |  5     5   |  5     6   |  4     4   |  6
   7     AB_Improved   7   |  3     6   |  4     4   |  6     6   |  4
-----------------------------------------------------------------------
          Win Rate:      64.3%        68.6%        64.3%        64.3%
```

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      10  |  0     8   |  2     10  |  0     8   |  2
   2       MM_Open     9   |  1     8   |  2     6   |  4     5   |  5
   3      MM_Center    9   |  1     10  |  0     10  |  0     9   |  1
   4     MM_Improved   6   |  4     7   |  3     6   |  4     6   |  4
   5       AB_Open     5   |  5     5   |  5     6   |  4     2   |  8
   6      AB_Center    5   |  5     2   |  8     8   |  2     3   |  7
   7     AB_Improved   5   |  5     3   |  7     6   |  4     4   |  6
-----------------------------------------------------------------------
          Win Rate:      70.0%        61.4%        74.3%        52.9%
```

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      9   |  1     10  |  0     10  |  0     7   |  3
   2       MM_Open     9   |  1     6   |  4     8   |  2     5   |  5
   3      MM_Center    6   |  4     7   |  3     9   |  1     9   |  1
   4     MM_Improved   5   |  5     1   |  9     4   |  6     4   |  6
   5       AB_Open     5   |  5     5   |  5     6   |  4     4   |  6
   6      AB_Center    6   |  4     5   |  5     6   |  4     4   |  6
   7     AB_Improved   6   |  4     3   |  7     5   |  5     3   |  7
-----------------------------------------------------------------------
          Win Rate:      65.7%        52.9%        68.6%        51.4%
```