Activity 3: Heuristic Analysis (Individual Project)
Group 2
Student Name: Ho Kim Chuan (17068452/1)
Group mate: Muhammad Alvin Pramadrian Aria (17141350/1)

Introduction

A heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

Isolation is a deterministic, two-player game of perfect information in which the players alternate turns moving a single piece from one cell to another on a board.

In the assignment we were asked to create a 'custom_agent' with a method of custom heuristic that will be used by the minimax/alpha-beta tree search algorithm.
The revamped code is written in 'game_agent.py' under the function 'custom_score'.
Afterwards, the created custom score function is called in the tournament.py. To execute the program, please run the 'tournament.py.'

Heuristic Agent Evaluation

The heuristic function created is tested repeatedly and gotten the average score listed as below. My teammate and I was taking 20 match as standard to conduct the evaluation to lessen the run time of the program. The custom heuristic function created will be compared with other heuristic function and the evaluation result (tested win rate) of respective heuristic function is listed as below:

Heuristic Function	Average win rate for 20 matches.(%)
Improved Score	67.14
Aggressive heuristic	66.43
Defensive heuristic	62.86
Maximizing win chances heuristic	61.43
Minimizing losing chances heuristic	64.29
Chances heuristic	62.86
Weighted chances heuristics	65.71
Weigted chances heuristic_2	69.29
Custom Heuristic (Self)	70.40
Custom (Group Mate)	70.24

Discussion

The heuristic function designed under the concept of instructing the agent to emphasized on a more Aggressive move in the front part of the game, in this example I take 50% progress of a match as a threshold, once the progress of a match is larger than 50%, the agent is instructed to emphasized on a more Defensive move. This approach successfully increases the win rate of the player while minimizing the win rate of the opponent. The simplified mathematically formula is stated as below:

$$H(t) = \begin{cases} p - 1.5 * (o), & \text{if match\_progress} <= 0.5 \\ 1.5 * p - (o), & \text{if match\_progress} > 0.5 \end{cases}$$

Let p = player's move, o = opponent's move

Conclusion

The heuristics that present dynamic weights, this is that their value changes depending on the state of the game promote a more regular behaviour. The heuristic function that able to split the timing of using the offensive and aggressive heuristic is able to reduce the lose rate of the player. The function performs slightly better than the heuristic function of my friend by chance, I am taking the average anyway.

The use of heuristics with changes of strategy in different moments of the game or with aggressive behaviour are more efficient.

Alpha-beta pruning with iterative deepening has a better performance than those that do not have it since its percentage of victories for all cases is above 50%, therefore the Opponents has turned out to be an extremely complicated adversary.