Heuristic Analysis

In the implementation of game agent, implemented agent is tested by using three different heuristic score calculation functions. The aim is to choose best function which gives best results against other agents. In this manner, I have chosen the heuristic function given in the first section. I recommend the first one based on experiments I have made with tests applied in “tournaments.py”. By using this function “Student” agent has managed to beat the “ID_Improved” agent by being successful in about %87 of tournament games. Below, experimental results of played games and success rates are given for three different score calculation functions, mainly the ultimate one is given.

1 - Ultimate Evaluation Function ⇒

#my_moves - 2 * #min_opponent_moves_one_level_deeper

The ultimate heuristic function is difference between number of current player’s moves and two times minimum number of opponent’s moves after each player move is applied on board. In the experiments, this function gave the best results among all.

The main reasons that I have chosen the first heuristic are;

- It achieved best score among three different heuristic functions and beat the “ID_Improved” agent in tournaments.
- It provides a more complex heuristic than other functions so that it takes the effect of each possible move of current player into account by checking opponent’s legal moves after its move and calculating difference between the number of current player moves and minimum number of opponent moves times two. Multiplying the opponent’s legal moves by two is beneficial in terms of blocking opponents moves more aggressively.
- Even if simpler heuristic functions yield the alphabeta function to go deeper, complex functions like this one could provide a good move sooner than simple ones as it does not need to go deeper in game tree.

```
************************
  Evaluating: Student
************************

Playing Matches:
----------
 Match 1:  Student  vs  Random     Result: 19 to 1
 Match 2:  Student  vs  MM_Null    Result: 20 to 0
 Match 3:  Student  vs  MM_Open    Result: 20 to 0
 Match 4:  Student  vs MM_Improved Result: 20 to 0
 Match 5:  Student  vs  AB_Null    Result: 18 to 2
 Match 6:  Student  vs  AB_Open    Result: 15 to 5
 Match 7:  Student  vs AB_Improved Result: 11 to 9

Results:
```

Student **87.86%**

2 - Second Evaluation Function ⇒

#my_moves - #min_opponent_moves_one_level_deeper

Second type of heuristic function is similar to ultimate one so that it calculates difference between number of current player's moves and two times minimum number of opponent's moves after each player move is applied on board. This function provides lower distinction than ultimate one. Test results for this type are given below.

```
*************************
  Evaluating: Student
*************************

Playing Matches:
----------
 Match 1:  Student  vs  Random     Result: 19 to 1
 Match 2:  Student  vs  MM_Null    Result: 19 to 1
 Match 3:  Student  vs  MM_Open    Result: 20 to 0
 Match 4:  Student  vs MM_Improved Result: 20 to 0
 Match 5:  Student  vs  AB_Null    Result: 15 to 5
 Match 6:  Student  vs  AB_Open    Result: 8 to 12
 Match 7:  Student  vs AB_Improved Result: 14 to 6


Results:
----------
Student         82.14%
```

3 - Third Evaluation Function ⇒ **#my_moves - 2 * #opponent_moves**

Third type of heuristic function is similar to ultimate one so that it calculates difference between number of current player's moves and two times number of opponent's moves. This function provides lower distinction than ultimate one. Test results for this type are given below.

```
*************************
  Evaluating: Student
*************************

Playing Matches:
----------
 Match 1:  Student  vs  Random     Result: 19 to 1
 Match 2:  Student  vs  MM_Null    Result: 19 to 1
```

```
Match 3:  Student  vs  MM_Open        Result: 20 to 0
Match 4:  Student  vs MM_Improved     Result: 20 to 0
Match 5:  Student  vs  AB_Null    Result: 15 to 5
Match 6:  Student  vs  AB_Open        Result: 8 to 12
Match 7:  Student  vs AB_Improved     Result: 14 to 6


Results:
----------
Student         82.14%
```