Heuristic Analysis

Heuristic 1: Scaling by time

We implement the heuristic in function **h_diffmove_over_numblank**. The function is constructed by taking the ratio of (#own_move - #opp_move) to (the number of blank spaces left + 1). A smaller value of the number of blanks indicates that the game is closer to its end. As the game approaches towards its end, if a player can maintain a higher value of (#own_move - #opp_move), the opponent has smaller chance of reversing the situation, and vice versa. Hence, a larger weighting is given to the move-difference when the number of blank spaces left is small.

The simulation result is demonstrated as follows:

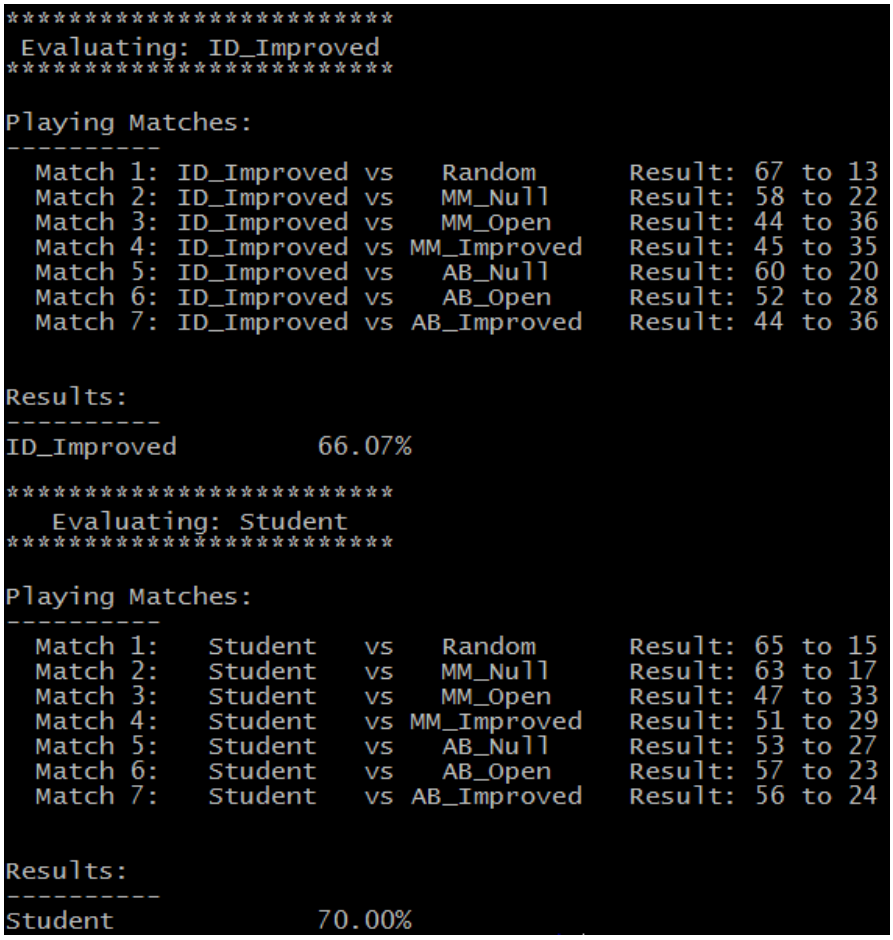
```
**************************
 Evaluating: ID_Improved
**************************

Playing Matches:
----------
  Match 1: ID_Improved vs    Random      Result: 49 to 31
  Match 2: ID_Improved vs    MM_Null     Result: 53 to 27
  Match 3: ID_Improved vs    MM_Open     Result: 53 to 27
  Match 4: ID_Improved vs MM_Improved    Result: 45 to 35
  Match 5: ID_Improved vs    AB_Null     Result: 54 to 26
  Match 6: ID_Improved vs    AB_Open     Result: 49 to 31
  Match 7: ID_Improved vs AB_Improved    Result: 49 to 31


Results:
----------
ID_Improved         62.86%

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

Playing Matches:
----------
  Match 1:   Student    vs    Random      Result: 64 to 16
  Match 2:   Student    vs    MM_Null     Result: 59 to 21
  Match 3:   Student    vs    MM_Open     Result: 47 to 33
  Match 4:   Student    vs MM_Improved    Result: 52 to 28
  Match 5:   Student    vs    AB_Null     Result: 63 to 17
  Match 6:   Student    vs    AB_Open     Result: 54 to 26
  Match 7:   Student    vs AB_Improved    Result: 46 to 34


Results:
----------
Student             68.75%
```

It is observed that the Student player beats ID_Improved by a winning rate 5.89%. Compared with ID_Improved, the Student performs better when playing with Random, MM_Null, MM_Improved, AB_Null and AB_Open.

Heuristic 2: Adding distance to the center

We implement the heuristic in function **h_diffmove_plus_diff2center**. We suspect that a smaller distance between the current position and the center of the board could indicate a higher number of potential moves in the future. The function is constructed by adding (#own_move - #opp_move) and (#opp_distance_to_center - #own_distance_to_center).

The simulation result is demonstrated as follows:



It is observed that the Student player beats ID_Improved by a winning rate 3.93%. Compared with the ID_Improved player, the Student player performs better when playing with MM_Null, MM_Open, MM_Improved, AB_open, and AB_Improved. Specifically, the Student player gets 15% higher winning rate than the ID_Improved player when playing with AB_Improved

Heuristic 3: Less aggressive when trapped at the boundary

We implement the heuristic in function **h_corner_less_aggressive**. For most cases, we return the utility (#own_move – 2 * #opp_move) to make the player more aggressive in trapping the opponent. However, when the player is located at the boundary or the corner of the board, it becomes less aggressive since it should first seek for positions open for movements. In this case, we return the utility used by the improved_score function: (#own_move – #opp_move).

The simulation result is demonstrated as follows:

```
**************************
 Evaluating: ID_Improved
**************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 70 to 10
  Match 2: ID_Improved vs   MM_Null     Result: 60 to 20
  Match 3: ID_Improved vs   MM_Open     Result: 49 to 31
  Match 4: ID_Improved vs MM_Improved   Result: 41 to 39
  Match 5: ID_Improved vs   AB_Null     Result: 60 to 20
  Match 6: ID_Improved vs   AB_Open     Result: 50 to 30
  Match 7: ID_Improved vs AB_Improved   Result: 48 to 32


Results:
----------
ID_Improved         67.50%

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

Playing Matches:
----------
  Match 1:   Student   vs   Random      Result: 69 to 11
  Match 2:   Student   vs   MM_Null     Result: 67 to 13
  Match 3:   Student   vs   MM_Open     Result: 49 to 31
  Match 4:   Student   vs MM_Improved   Result: 52 to 28
  Match 5:   Student   vs   AB_Null     Result: 60 to 20
  Match 6:   Student   vs   AB_Open     Result: 49 to 31
  Match 7:   Student   vs AB_Improved   Result: 47 to 33


Results:
----------
Student             70.18%
```

It is observed that the Student player beats ID_Improved by a winning rate 2.68%. Compared with the ID_Improved player, the Student player performs better when playing with MM_Null and MM_Improved. Specifically, the Student player gets 13.75% higher winning rate than the ID_Improved player when playing with MM_Improved.

Overall speaking, the **Heuristic 1** with move count difference weighted by the stage of the game gets the highest overall winning rate. However, Heuristic 2 and Heuristic 3 also show significant improvements when competing with smarter opponents such as AB_Improved and MM_Improved, respectively.