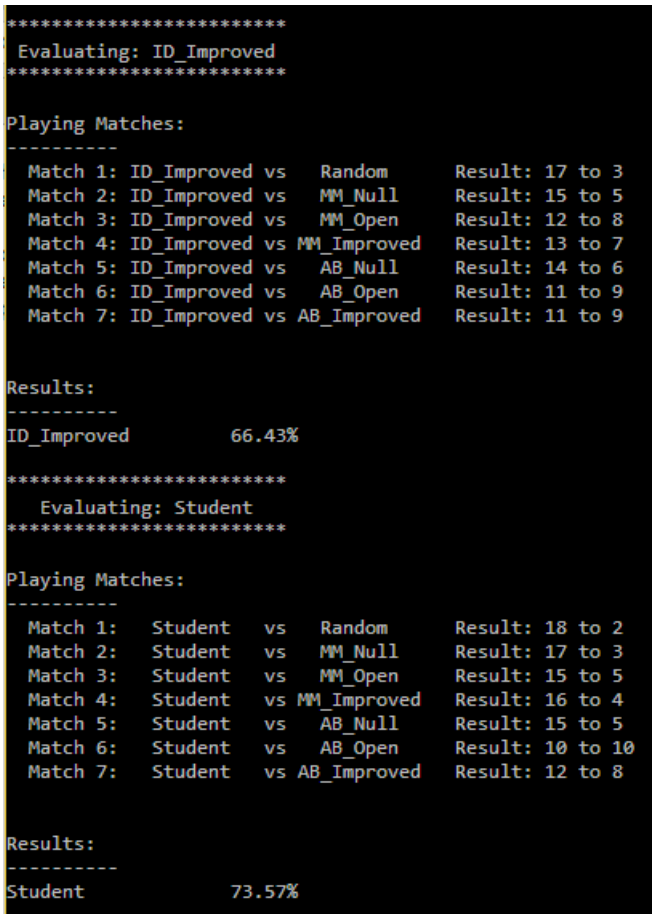
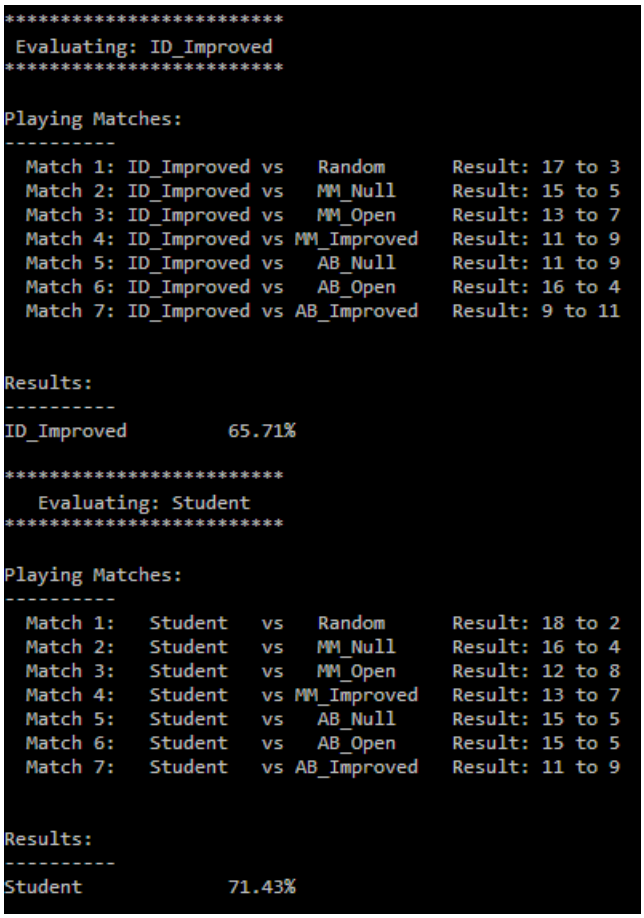
Analysis of Custom Heuristic Functions

1st Heuristic: Difference in moves between players but with bias against opponent moves.

- **Name**: pro_self_score
- **Formula**: own_moves - (opp_moves * .9)
- **Description**: The idea behind this heuristic is to emphasize the number of player moves over the number of opponent moves. This is more evident during evaluations that return the same values. For example the values [7 - 4], [8 - 5], [6 - 3] would all return 3 but with the above calculation [3.4, 3.5, 3.3], the second option is chosen.
- **Results**:





This heuristic function performed better against ID_Improved most of the time. The test performed better 7 out of 10 times.

ID_Improved	Student
66.43	73.57
65.71	71.43
63.57	69.29

2nd Heuristic: Difference in moves between players but with bias against player moves

- **Name:** anti_opponent_score
- **Formula**: (own_moves*.9) - opp_moves
- **Description**: The idea behind this heuristic is to emphasize the number of opponent moves over the number of player moves. This is the counterpart of the 1st heuristic. As with the above, this is more evident during evaluations that return the same values. With the same example values [7 - 4], [8 - 5], [6 - 3] would all return 3 but with the above calculation [2.3, 2.2, 2.4], the third option is chosen.
- **Results**:

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

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 17 to 3
  Match 2: ID_Improved vs   MM_Null     Result: 15 to 5
  Match 3: ID_Improved vs   MM_Open     Result: 13 to 7
  Match 4: ID_Improved vs MM_Improved   Result: 14 to 6
  Match 5: ID_Improved vs   AB_Null     Result: 15 to 5
  Match 6: ID_Improved vs   AB_Open     Result: 11 to 9
  Match 7: ID_Improved vs AB_Improved   Result: 10 to 10


Results:
----------
ID_Improved         67.86%

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

Playing Matches:
----------
  Match 1:   Student   vs   Random      Result: 16 to 4
  Match 2:   Student   vs   MM_Null     Result: 15 to 5
  Match 3:   Student   vs   MM_Open     Result: 15 to 5
  Match 4:   Student   vs MM_Improved   Result: 15 to 5
  Match 5:   Student   vs   AB_Null     Result: 13 to 7
  Match 6:   Student   vs   AB_Open     Result: 15 to 5
  Match 7:   Student   vs AB_Improved   Result: 12 to 8


Results:
----------
Student             72.14%
```

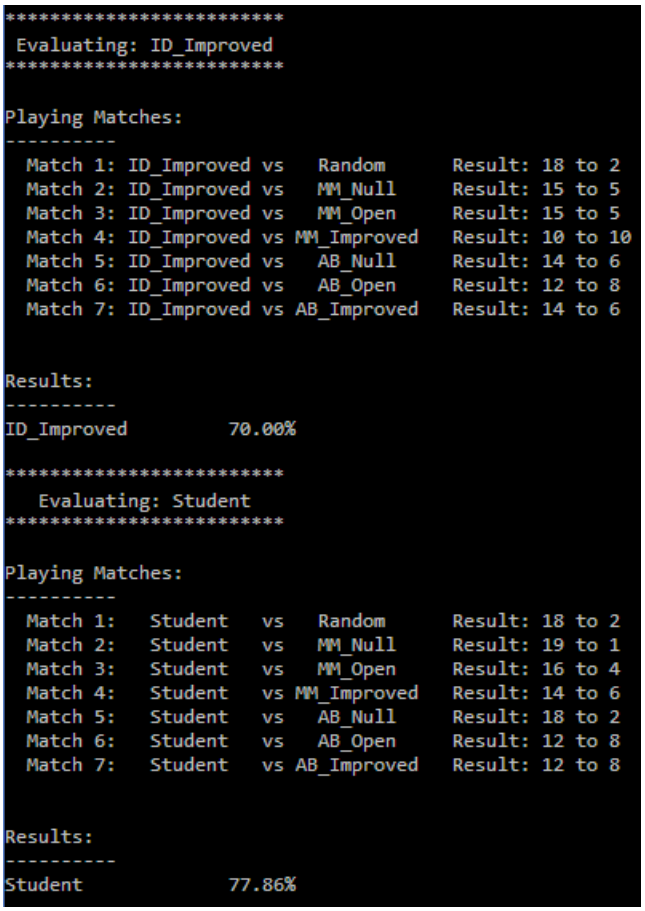
```
************************
 Evaluating: ID_Improved
************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 16 to 4
  Match 2: ID_Improved vs   MM_Null     Result: 17 to 3
  Match 3: ID_Improved vs   MM_Open     Result: 14 to 6
  Match 4: ID_Improved vs MM_Improved   Result: 7 to 13
  Match 5: ID_Improved vs   AB_Null     Result: 12 to 8
  Match 6: ID_Improved vs   AB_Open     Result: 11 to 9
  Match 7: ID_Improved vs AB_Improved   Result: 12 to 8


Results:
----------
ID_Improved         63.57%

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

Playing Matches:
----------
  Match 1:   Student   vs   Random      Result: 18 to 2
  Match 2:   Student   vs   MM_Null     Result: 15 to 5
  Match 3:   Student   vs   MM_Open     Result: 11 to 9
  Match 4:   Student   vs MM_Improved   Result: 12 to 8
  Match 5:   Student   vs   AB_Null     Result: 16 to 4
  Match 6:   Student   vs   AB_Open     Result: 14 to 6
  Match 7:   Student   vs AB_Improved   Result: 11 to 9


Results:
----------
Student             69.29%
```

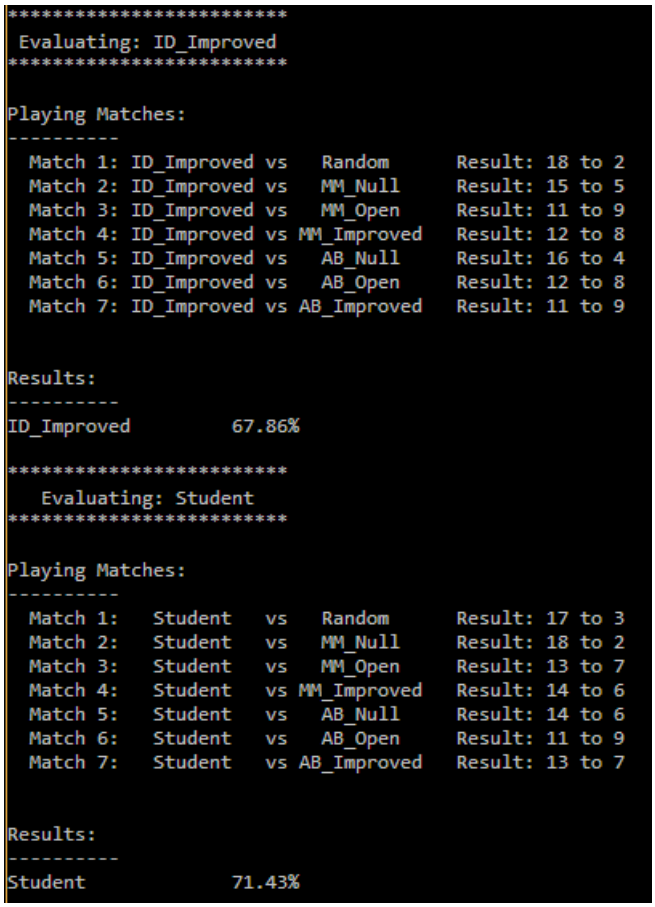
This heuristic function performed better against ID_Improved most of the time but a bit worse than the first function. The test performed better 6 out of 10 times.

ID_Improved	Student
67.86	72.14
63.57	69.29
69.29	70.00

3rd Heuristic: Difference in moves between players but with bias determined by free moves

- **Name:** free_score
- **Formula**: (own_moves*.9) - opp_moves or own_moves - (opp_moves * .9)
- **Description**: This heuristic combines the above by taking into account the number of free spaces and comparing it to the total number of moves. The idea is that if the player and the opponent share a lot of moves and there are limited free spaces left, the heuristic would be more partial to reducing the opponent's moves. Conversely, if there are a lot of free spaces, the heuristic would be more inclined towards player moves.
- **Results**:





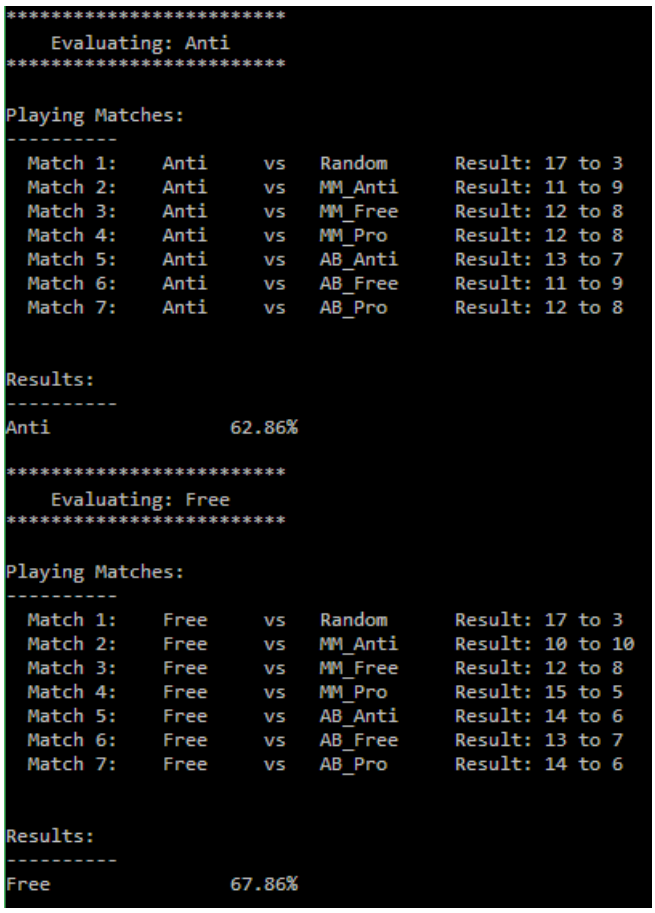
This heuristic function performed better against ID_Improved and the two previous functions. It also consistently produces the highest score out of the three. I ran the test ten times and it got a higher result for nine of them and with consistently higher scores (>70%).

ID_Improved	**Student**
70.00	77.86
67.86	71.43
65.71	77.14
66.43	76.43
75.00	77.14
72.86	65.71
69.29	70.00
68.57	72.14

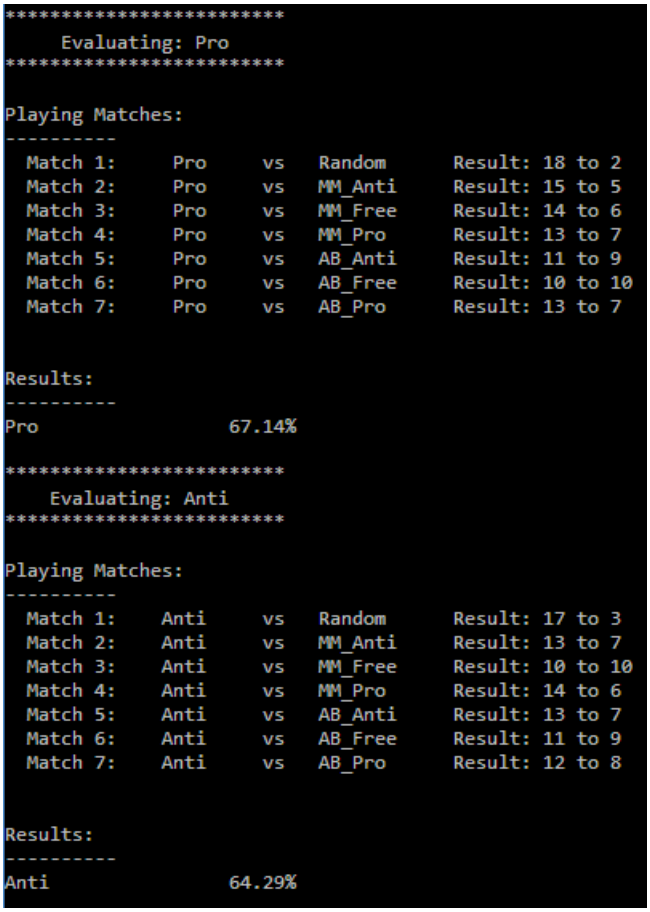
Conclusion:

Out of all three custom heuristic functions, the third one is the best choice. The reasons are as follows:

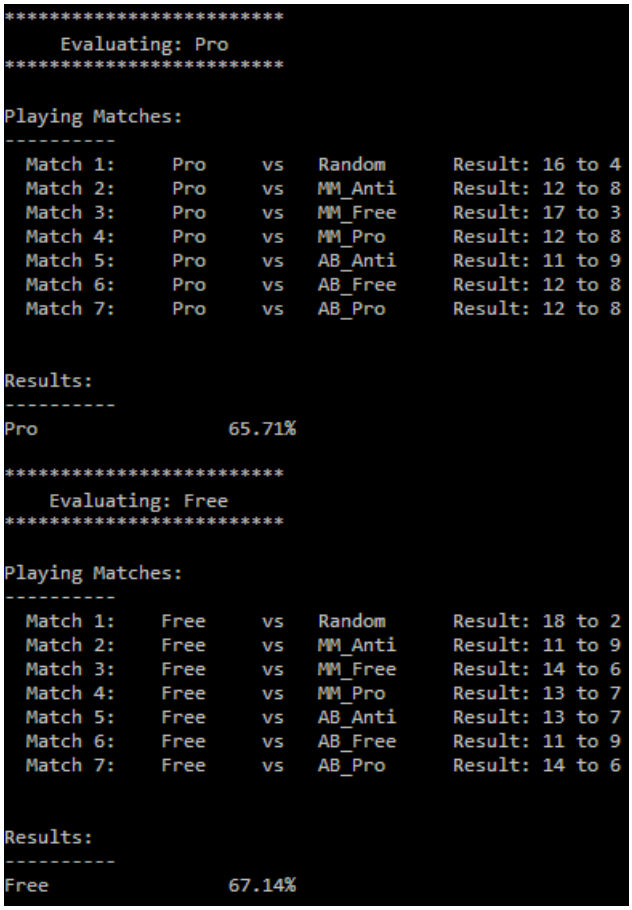
- Out of the three, the *free_score* function is the most consistent in defeating ID_Improved. It scores higher about nine out of ten times. Of course, this is only based on the limited runs I have performed so it is far from definitive.
- The *free_score* function produced the highest score among all the functions tested including ID_Improved in all of the runs which is 77.86%. It also frequently produces scores above 70%.
- It won against the other two functions after modifying tournament.py to make them compete against each other. (See results in the images below)



anti_opponent_score vs free_score



pro_self_score vs anti_opponent_score



pro_self_score vs free_score