Artificial Intelligence Nanodegree Program
Mitesh Valechha

Heuristics Analysis

AB_Custom:-
This function evaluates the difference between square of my_moves and 2 times the square of opp_moves

```
    my_moves = len(game.get_legal_moves(player))
    opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float((my_moves**2) - (2*(opp_moves**2)))
```

AB_Custom_2:-
This evaluation causes the player to chase behind the opponent as opp_moves is multiplied with 2 then subtracted from my_moves.

```
    my_moves = len(game.get_legal_moves(player))
    opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float(my_moves - opp_moves * 2)
```

AB_Custom_3:-
This function evaluates the quotient of my_moves by opp_moves + 1, 1 is added to ensure divided by zero error doesn't occur.

```
    my_moves = len(game.get_legal_moves(player))
    opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float(my_moves/(opp_moves+1))
```

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

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random      29 |   1     28 |   2     30 |   0     30 |   0
    2       MM_Open     21 |   9     20 |  10     23 |   7     19 |  11
    3      MM_Center    24 |   6     25 |   5     27 |   3     28 |   2
    4     MM_Improved   21 |   9     25 |   5     23 |   7     20 |  10
    5       AB_Open     17 |  13     19 |  11     18 |  12     12 |  18
    6      AB_Center    16 |  14     18 |  12     19 |  11     18 |  12
    7     AB_Improved   15 |  15     18 |  12     15 |  15     18 |  12
--------------------------------------------------------------------------
           Win Rate:      68.1%        72.9%        73.8%        69.0%
```

Based on above results, the evaluation function that I would recommend is AB_Custom_2 as it has the best winrate of all the Heuristics. That’s because the algorithm causes the player to chase after the opponent which is an offensive strategy that tries minimize opponent moves. Also the function is quite simple and not complex which allows it to go deeper in the search tree. We can also see that the function performs very good against random opponents too.