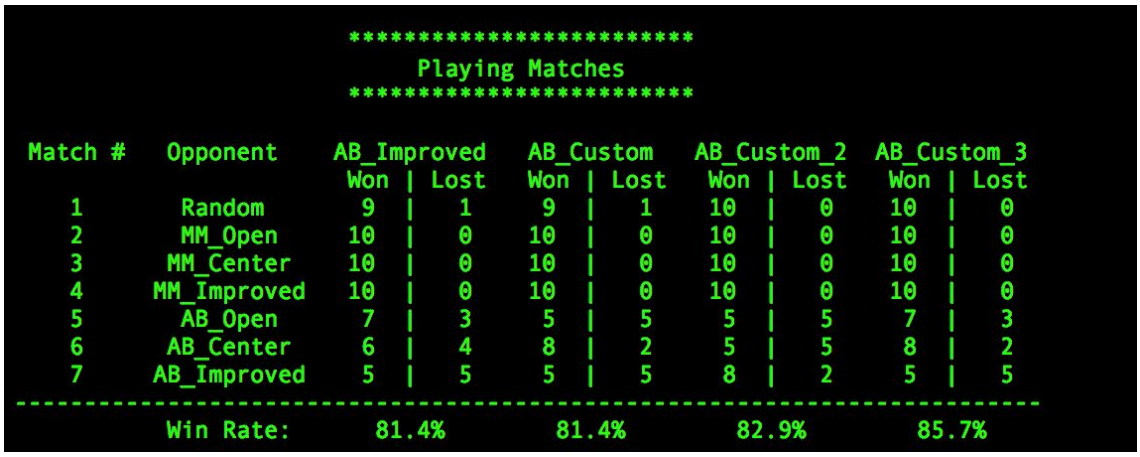
Heuristic analysis

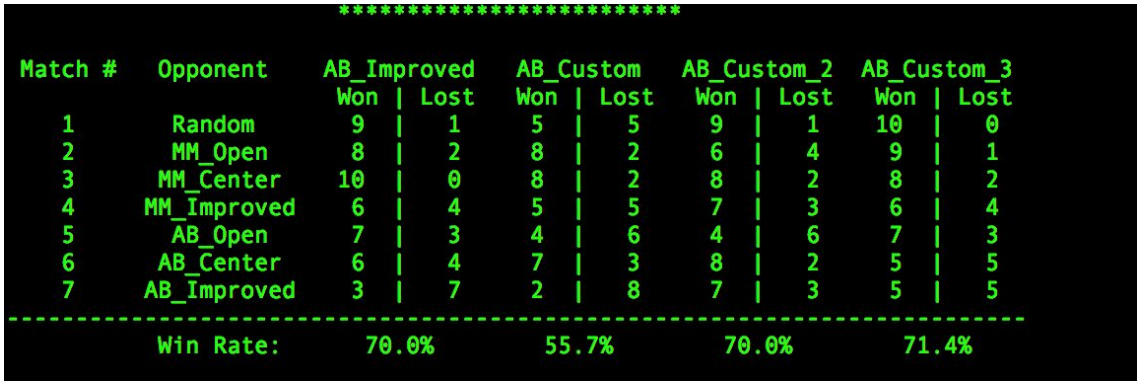
I implemented 3 custom scores as a part of the game. Its is very difficult to pick one good heuristic amongst the 3. In general a deeper search and a lower time limit gives me results above 80% for AB_Improved, AB_Customscore, AB_Customscore1 and AB_Customscore2. Screenshot below.
Time_limit = 15 millisec
Depth = 7



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

 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2   AB_Custom_3
                         Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1        Random       9  |   1     9  |   1    10  |   0    10  |   0
    2        MM_Open     10  |   0    10  |   0    10  |   0    10  |   0
    3       MM_Center    10  |   0    10  |   0    10  |   0    10  |   0
    4     MM_Improved    10  |   0    10  |   0    10  |   0    10  |   0
    5        AB_Open      7  |   3     5  |   5     5  |   5     7  |   3
    6       AB_Center     6  |   4     8  |   2     5  |   5     8  |   2
    7     AB_Improved     5  |   5     5  |   5     8  |   2     5  |   5
-----------------------------------------------------------------------------
           Win Rate:      81.4%        81.4%        82.9%        85.7%
```

Now reducing the depth to 3 and increasing the time limit to 25 - I get the following results. If you see below AB_Custom has reduced from 81.4% to 55% and there is a significant drop in other heuristics as well



```
                      ************************

 Match #    Opponent    AB_Improved    AB_Custom    AB_Custom_2   AB_Custom_3
                         Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1        Random       9  |   1     5  |   5     9  |   1    10  |   0
    2        MM_Open      8  |   2     8  |   2     6  |   4     9  |   1
    3       MM_Center    10  |   0     8  |   2     8  |   2     8  |   2
    4     MM_Improved     6  |   4     5  |   5     7  |   3     6  |   4
    5        AB_Open      7  |   3     4  |   6     4  |   6     7  |   3
    6       AB_Center     6  |   4     7  |   3     8  |   2     5  |   5
    7     AB_Improved     3  |   7     2  |   8     7  |   3     5  |   5
-----------------------------------------------------------------------------
           Win Rate:      70.0%        55.7%        70.0%        71.4%
```

Now keeping the time limit to 25 and increasing the depth to 8 - there is an increase in winning percentage across all the heuristics

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

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random       7  |   3     9  |   1     7  |   3     8  |   2
   2       MM_Open     10  |   0    10  |   0    10  |   0    10  |   0
   3      MM_Center    10  |   0    10  |   0    10  |   0    10  |   0
   4     MM_Improved   10  |   0    10  |   0    10  |   0    10  |   0
   5       AB_Open      5  |   5     7  |   3     7  |   3     5  |   5
   6      AB_Center     6  |   4     6  |   4     6  |   4     7  |   3
   7     AB_Improved    3  |   7     6  |   4     5  |   5     6  |   4
-----------------------------------------------------------------------
          Win Rate:      72.9%        82.9%        78.6%        80.0%
```

Increasing the depth to 11

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

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random       8  |   2     9  |   1     7  |   3    10  |   0
   2       MM_Open     10  |   0    10  |   0    10  |   0    10  |   0
   3      MM_Center    10  |   0    10  |   0    10  |   0    10  |   0
   4     MM_Improved   10  |   0    10  |   0    10  |   0    10  |   0
   5       AB_Open      5  |   5     8  |   2     6  |   4     5  |   5
   6      AB_Center     5  |   5     9  |   1     6  |   4     9  |   1
   7     AB_Improved    4  |   6     3  |   7     6  |   4     5  |   5
-----------------------------------------------------------------------
          Win Rate:      74.3%        84.3%        78.6%        84.3%
```

Increasing the depth to 13

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

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      10  |   0     9  |   1     9  |   1     7  |   3
   2       MM_Open     10  |   0    10  |   0    10  |   0    10  |   0
   3      MM_Center    10  |   0    10  |   0    10  |   0    10  |   0
   4     MM_Improved   10  |   0    10  |   0    10  |   0    10  |   0
   5       AB_Open      5  |   5     7  |   3     6  |   4     5  |   5
   6      AB_Center     8  |   2     8  |   2     9  |   1     6  |   4
   7     AB_Improved    4  |   6     4  |   6     8  |   2     4  |   6
-----------------------------------------------------------------------
          Win Rate:      81.4%        82.9%        88.6%        74.3%
```

Again this does not mean that by increasing the depth linearly will give you a better result, quiesence will be reached.

Here is a brief description of all the heuristics

1. Custom_score - reward your player if your player has more moves than the opponent's move

```
if game.is_loser(player):
        return float("-inf")

    if game.is_winner(player):
        return float("inf")

    own_moves = len(game.get_legal_moves(player))
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float(own_moves - opponent_moves)
```

2. Customer_Score2 - play more aggressively and reward your player if your player has more legal moves to make than the opponents

```
if game.is_loser(player):
        return float("-inf")

    if game.is_winner(player):
        return float("inf")

    #return float(len(game.get_legal_moves(player)))
    own_moves = len(game.get_legal_moves(player))
    #print(len(own_moves))
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    #print(len(opponent_moves))
    return float(own_moves - (2*opponent_moves))
```

3. Customer_Score2 - play more aggressively if the number of blank spaces are above 45 and less aggressively towards the end of the game

```
if game.is_loser(player):
        return float("-inf")

    if game.is_winner(player):
        return float("inf")

    blank_spaces = game.get_blank_spaces()
    #return float(len(game.get_legal_moves(player)))
    own_moves = len(game.get_legal_moves(player))
    #print(len(own_moves))
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    if len(blank_spaces) >= 45:
        return float(own_moves - 3*opponent_moves)
    else:
        return float(3*own_moves - opponent_moves)
```

Conclusion: Based on the consistent performance overall between Custom_score2 and Customer_score3 - I will either chose customer_score2 or customer_score3. And again if I have to chose between custom_score2 and Customer_score3, I will chose customer_score3 as it has more scores in the 80's than custom_score2.