Heuristic 1:

This is an adaptation of the heuristic which checks for the difference between the legal moves of the two players. Further fine tuning is done to accommodate the following:

1. Maximum number of squares or spaces at start – game width * game height (max)
2. Legal moves available at the given time for player and opponent (a, b)
3. Blank spaces available at the given time for both players (m)

Formula:
Available spaces for Player = max – m – a
Available spaces for Opponent = m – b
Difference = (max-m-a) – (m-b)

The idea is penalize the player at the earlier stages when max – is high so that the player is defensive and selects the right moves.

```
                    *************************
                         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     8  |  2     10 |  0     8  |  2
    2       MM_Open       6  |  4     5  |  5      5 |  5     7  |  3
    3      MM_Center      7  |  3     7  |  3      8 |  2     7  |  3
    4     MM_Improved     5  |  5     5  |  5      7 |  3     8  |  2
    5       AB_Open       5  |  5     6  |  4      4 |  6     7  |  3
    6      AB_Center      6  |  4     3  |  7      6 |  4     7  |  3
    7     AB_Improved     6  |  4     4  |  6      3 |  7     5  |  5
--------------------------------------------------------------------------
          Win Rate:      60.0%        54.3%        61.4%        70.0%
```

This heuristic performs more or less closer to the AB_Improved.

Heuristic 2:

This is an adaptation of the heuristic which checks for the difference between the legal moves of the two players. Further fine tuning is done to accommodate the following:

1. Number of moves consumed during the game – game.move_count() (a)
2. Legal moves available at the given time for player and opponent (b,c)

Formula:
Available spaces for Player = b - a
Available spaces for Opponent = c

The idea is incentivize the player at the earlier stages when b - a is high so that the player is aggressive and selects the right moves. As the game progress, b – a becomes low and the player adopts a more defensive strategy.

```
                    *************************
                        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     10  |  0     7   |  3
   2       MM_Open     6   |  4     6   |  4     9   |  1     7   |  3
   3      MM_Center    7   |  3     8   |  2     6   |  4     8   |  2
   4     MM_Improved   6   |  4     6   |  4     7   |  3     6   |  4
   5       AB_Open     4   |  6     3   |  7     5   |  5     6   |  4
   6      AB_Center    6   |  4     6   |  4     7   |  3     6   |  4
   7     AB_Improved   5   |  5     1   |  9     6   |  4     5   |  5
------------------------------------------------------------------------
          Win Rate:      58.6%        55.7%        71.4%        64.3%
```

This heuristic performs most times better than AB_Improved.

Heuristic 3:

This is an adaptation of the null score heuristic in sample_players.py. Here we also assume no knowledge of terminal states and return a random number in the range (-10, 10).

Formula:
Return a random number in the range(-10,10)

```
                    *************************
                        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     8   |  2     7   |  3
   2       MM_Open     6   |  4     4   |  6     7   |  3     5   |  5
   3      MM_Center    9   |  1     8   |  2     7   |  3     9   |  1
   4     MM_Improved   8   |  2     6   |  4     6   |  4     7   |  3
   5       AB_Open     5   |  5     5   |  5     6   |  4     3   |  7
   6      AB_Center    3   |  7     5   |  5     5   |  5     5   |  5
   7     AB_Improved   5   |  5     5   |  5     5   |  5     4   |  6
------------------------------------------------------------------------
          Win Rate:      62.9%        60.0%        62.9%        57.1%
```

This heuristic performs most times better than AB_Improved.