Analysis

Since there is a huge randomness, I have followed the first advice from the re-view and increased the number of matches to 50

```
                              ************************
                                   Playing Matches
                              ************************
Match # Opponent AB_Improved     AB_Custom     AB_Custom_2 AB_Custom_3
                   Won| Lost     Won | Lost    Won | Lost    Won | Lost
1 Random           45 | 5        46  | 4       45 | 5        47 | 3
2 MM_Open          39 | 11       31 | 19       34 | 16       33 | 17
3 MM_Center        46 | 4        48  | 2       44 | 6        44 | 6
4 MM_Improved      38 | 12       34 | 16       28 | 22       29 | 21
5 AB_Open          30 | 20       26 | 24       24 | 26       25 | 25
6 AB_Center        31 | 19       26 | 24       27 | 23       26 | 24
7 AB_Improved      28 | 22       25 | 25       24 | 26       19 | 31
--------------------------------------------------------------------------
Win Rate:          73.4%        66.7%        64.6%        63.3%
```

Custome score:

Here I used the same logic as the improved score function in sample_player.py, but also subtracted the distance from the center of the board to keep my moves away from the edges, so that I can have more moves in the future

Logic is to get the nearest move to the center that limits my opponent's future moves and increases my future moves

Equation: number of my available moves – number of opponent's moves – distance to the center of the board "using Pythagoras theorem"

Results:

```
Agent            Won | Lost
Random           46 | 4
MM_Open          31 | 19
MM_Center        48 | 2
MM_Improved      34 | 16
AB_Open          26 | 24
AB_Center        26 | 24
AB_Improved      25 | 25
```

In this heuristic, the winning rate is higher than the other heuristics as it has two conditions: difference between the player's and his opponent's moves and the distance from the center, but the winning rate isn't bigger than the other heuristics which indicate that the distance from the center isn't actually significant when playing the game, this is appearing clearly when compared to the agents using alpha-beta pruning where the difference between the matches won and lost isn't big. In summary, considering the distance from the center isn't decisive, it depends on the state of the game.

Custome_score_2:

Using the heuristic in the nano-degree to subtract the opponent's available moves from mine, I added it here to compare it to Custome_score and make sure that subtracting the distance from the center did improve this heuristic.

Results:

Agent	Won | Lost
Random	45 | 5
MM_Open	34 | 16
MM_Center	44 | 6
MM_Improved	28 | 22
AB_Open	24 | 26
AB_Center	27 | 23
AB_Improved	24 | 26

In this heuristic we aim to go for the move that increase the number of our next moves while decreasing our opponents moves, as we see it didn't give good results with alpha-beta agents as alpha-beta agents can visit more nodes than mini-max agents so they can go deeper and perform better.

Custome_score_3:

Here I used the open_move heuristic where only the move that produces more future moves for our player is chosen.

Results:

Agent	Won | Lost
Random	47 | 3
MM_Open	33 | 17
MM_Center	44 | 6
MM_Improved	29 | 21
AB_Open	25 | 25
AB_Center	26 | 24
AB_Improved	19 | 31

In this heuristic we aim to go for the move that increase the number of our next moves only. Since there is a wide variety of moves that can be chosen and we choose the moves that come in the for loop first, so the effect is actually random here as we choose the node depending on its order in the loop so sometimes the move is good and sometimes it's bad, this heuristic is the least from performance point of view, but that is expected specially with alpha-beta agents that go deeper and scan much more nodes in the given time threshold.

My recommendation is to use Custome-score2 because:

1. it's not as complex as custom_score_1 as we don't calculate distances or use Pythagoras theorem
2. Despite it is simple it gives results that are near to the results of heuristic taking into account the distance from the center of the board, so its gain is high compared to its complexity.
3. It doesn't consume much time due to its simplicity allowing more time for going deeper into the tree.