The goal of this project was to design an AI agent for the adversarial game ‘Isolation’. AI agent applies minimax algorithm which conceptually based on the Game theory principles. In Game Theory each player makes a move in response to the best move of the opponent.

To estimate the value of value of legal move, the AI agent depends upon utility function. The utility functions provides a numerical based on the state of the game which makes comparison among different moves of the game feasible.

In the first pass of the development, I used the custom scores provided in the sample_agent.py. These 3 custom scores functions provide below listed heuristics:

1. # of legal moves for the player
2. # of own_moves - # of opponent_moves: The heuristic implies that that player with more legal moves has the advantage in the game
3. Euclidean distance between the physical center of the board and the current position of the player. The idea is that a player situated in the center of the board will have an upper hand in the game as it can maneuver more number of moves.

As the AI agent just replicates the strategy of opponent, the win ratio obtained by initially replicating the above heuristics from the sample_agent provided below doesn’t capture the relative importance of the heuristics.

HEURISTIC	**WIN RATIO**
OWN MOVES – OPPONENT MOVES	64.3%
OPEN MOVES (# OF LEGAL MOVES)	67.1%
EUCLIDEAN DISTANCE FROM CENTER	55.7%

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

 Match #    Opponent     AB_Improved    AB_Custom    AB_Custom_2   AB_Custom_3
                          Won  | Lost    Won  | Lost    Won  | Lost    Won  | Lost
    1       Random        8   |   2      8   |   2      9   |   1      9   |   1
    2       MM_Open       4   |   6      7   |   3      6   |   4      4   |   6
    3      MM_Center      7   |   3      9   |   1      8   |   2      7   |   3
    4     MM_Improved     7   |   3      6   |   4      6   |   4      6   |   4
    5       AB_Open       5   |   5      6   |   4      4   |   6      4   |   6
    6      AB_Center      8   |   2      7   |   3      4   |   6      4   |   6
    7     AB_Improved     6   |   4      4   |   6      8   |   2      5   |   5
--------------------------------------------------------------------------------
          Win Rate:      64.3%        67.1%        64.3%        55.7%
```

I implemented additional heuristic functions to explore the AI agent behavior.

Manhattan Distance: This heuristic returned the manhattan distance between the agent and its opponent. This heuristics didn't yield any improvements in terms of the win ratio shown as the highlighted column below.

```
                    *************************
                         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     8 |  2     9 |  1     5 |  5
    2       MM_Open     6 |  4     7 |  3     8 |  2     6 |  4
    3      MM_Center    9 |  1     9 |  1     7 |  3     7 |  3
    4     MM_Improved   3 |  7     7 |  3     6 |  4     4 |  6
    5       AB_Open     6 |  4     6 |  4     6 |  4     7 |  3
    6      AB_Center    4 |  6     3 |  7     7 |  3     5 |  5
    7     AB_Improved   4 |  6     7 |  3     5 |  5     5 |  5
--------------------------------------------------------------------------
          Win Rate:      57.1%        67.1%        68.6%        55.7%
```

Recommendation:

The heuristic that evaluates the board position by comparing the number of legal moves of the player and its opponent is certainly better. This heuristic results in better win ratio. The heuristic provides a utility value that is easier and faster to compute. Intuitively the heuristics makes sense as a player who can navigate more positions on the board has an advantage.