Types of heuristics used for isolation project:

1) **AB_Custom**

The ‘quality’ of legal moves available to a player was evaluated by considering each legal move of a player separately. ‘Quality’ was computed by finding the distance from the edge of each legal move. If a move is located close to the boundary, a lower score is assigned to that move while a higher score is assigned otherwise. The quality of move available to each player was multiplied with the corresponding number of legal moves available for the player and difference between the score of player and opponent was computed by taking a linear combination of the two.

The snapshot below shows the method used to compute quality of each legal move.

```
def quality_legal_moves(game, legal_move_play):
    y_legal_move, x_legal_move = legal_move_play
    w, h = game.width / 2., game.height / 2.
    dist_from_center_x,dist_from_center_y = abs(x_legal_move - h), abs(y_legal_move - w)
    if (dist_from_center_x <= h-1 and dist_from_center_y <= w-1):
        score_for_this_legal_move = 4.
    elif (dist_from_center_x > h-1 and dist_from_center_y <= w-1) or (dist_from_center_x <= h-1 and dist_from_center_y > w-1):
        score_for_this_legal_move = 3.
    elif (dist_from_center_x > h-1 and dist_from_center_y > w-1):
        score_for_this_legal_move = 2.
    else:
        score_for_this_legal_move = 1.
    return score_for_this_legal_move
```

*Score = Player_score*number_player_legal_moves - 1.2*Opponent_score*number_opponent_legal_moves*

2) **AB_Custom_2**

For this case, a linear combination of center score (distance from center), difference between legal moves for each player and number of moves available to the player was used.

*Score = 0.6*score_center_player + 0.3*difference_open_moves + 0.1*player_open_moves*

3) **AB_Custom_3**

Manhattan distance between the players linearly combined with difference between open moves for each player and center score was used to compute score for this heuristic.

*Score = 0.4*score_center_player + 0.2*difference_open_moves+ 0.4*manhattan_distance_player_opposition*

Best heuristic

Based on the simulations conducted, **AB_Custom** is the best amongst all the heuristics. Reasons which explain why it could be best are:

- Uses linear combination of quality of legal moves weighted by number of legal moves available to each player
- The snapshots below shows that with this heuristic, the win rate about 6% higher than the AB_Improved scoring function
- Weighs the importance in score of player and opponent through a penalization factor in computing linear combination of scores. The player will have to be much better than the opponent by a factor in order to get a positive score considering other parameters to be constant, in this case the player has to be better by a factor of 1.3
- Looks at just one depth deeper in the tree (by using iterative deepining) thus avoiding computationally intensive heuristics

An alternate approach to calculate even better heuristics could be to use Machine learning techniques such as deep q learning based reinforcement learning that could potentially come up with even better strategy for the game

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

Match #   Opponent    AB_Improved   AB_Custom    AB_Custom_2   AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost    Won | Lost
   1       Random      15  |   5    18  |   2    18  |   2     19  |   1
   2       MM_Open     15  |   5    16  |   4    12  |   8     15  |   5
   3      MM_Center    18  |   2    18  |   2    16  |   4     17  |   3
   4     MM_Improved   13  |   7    19  |   1    14  |   6     17  |   3
   5       AB_Open     11  |   9    12  |   8     8  |  12     12  |   8
   6      AB_Center    12  |   8    11  |   9    12  |   8     10  |  10
   7     AB_Improved   10  |  10     9  |  11     6  |  14     11  |   9
--------------------------------------------------------------------------
          Win Rate:      67.1%        73.6%        61.4%         72.1%
```

The plot below shows a game between: **AB Improved** (Alpha beta player with improved score heuristics) and **AB Custom Score** (Alpha beta player with best heuristics). It can be seen that for this case the winner is AB player with custom heuristics (**AB_Custom**).After the end of game, AB Custom heuristics still has at least two legal moves left.

Player1:
AB Improved
Player2:
AB Custom
Move History:
[[0, 2], [1, 3], [1, 4], [3, 2], [3, 5], [2, 0], [4, 3], [4, 1], [6, 2], [5, 3], [5, 4], [3, 4], [6, 6], [4, 2], [4, 5], [6, 3], [2, 4], [4, 4], [1, 6], [5, 2], [0, 4], [3, 1], [2, 5], [1, 0], [0, 6], [2, 2]]
Run Game

AB Improved vs AB Custom

(0,2)	(1,3)
(1,4)	(3,2)
(3,5)	(2,0)
(4,3)	(4,1)
(6,2)	(5,3)
(5,4)	(3,4)
(6,6)	(4,2)
(4,5)	(6,3)
(2,4)	(4,4)
(1,6)	(5,2)
(0,4)	(3,1)
(2,5)	(1,0)
(0,6)	(2,2)