Heuristics review

Custom_score:
This evaluation function uses the player and opponent's location to calculate the absolute distance
by elevating the difference in their positions to the power of 2.
As a result shorter distances between them get rewarded.
If our player begins the game being one move ahead and close to the second player increases the chance
of isolating the opponent.
This function has consistently performed better than AB_improved.

Custom_score2:
Following Malcolm advise in the lecture, this evaluation is based on the player's ability to reflect the opponent's moves.
This is helpful at the begining of the game, when the board is fairly open.
The function uses the opponent's possible moves to calculate how many of those is the player capable of reflecting
by searching his own possible moves in a 180-degree rotated version of the game.
The reflected move count is used as reward coefficient to player's moves count.
This function is winning 50% of the time other players. I think it could perform better if tournament.py let you choose
where to play the first move, so you can take the center.

Custom_score3:
This evaluation functions calculates how many of the possible moves would leave the player in a bad position.
It assumes a bad position being places where mobility is limited (i.e. corners).
Those 'bad moves' are entered in the calculation as a detriment to the respective player's moves count.
This function works better towards the end of the game where moves are limited and placing yourself in a poor spot is critical.
This function is winning 45% of the time which is low compare to the rest. Creating a game spent coefficient could help.