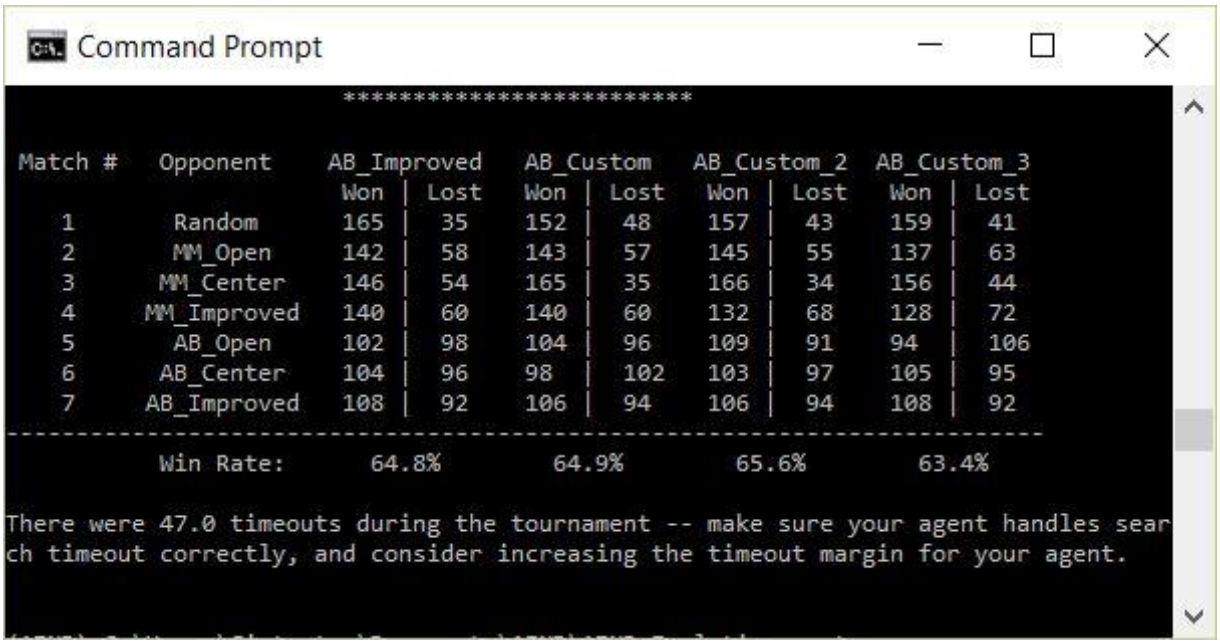
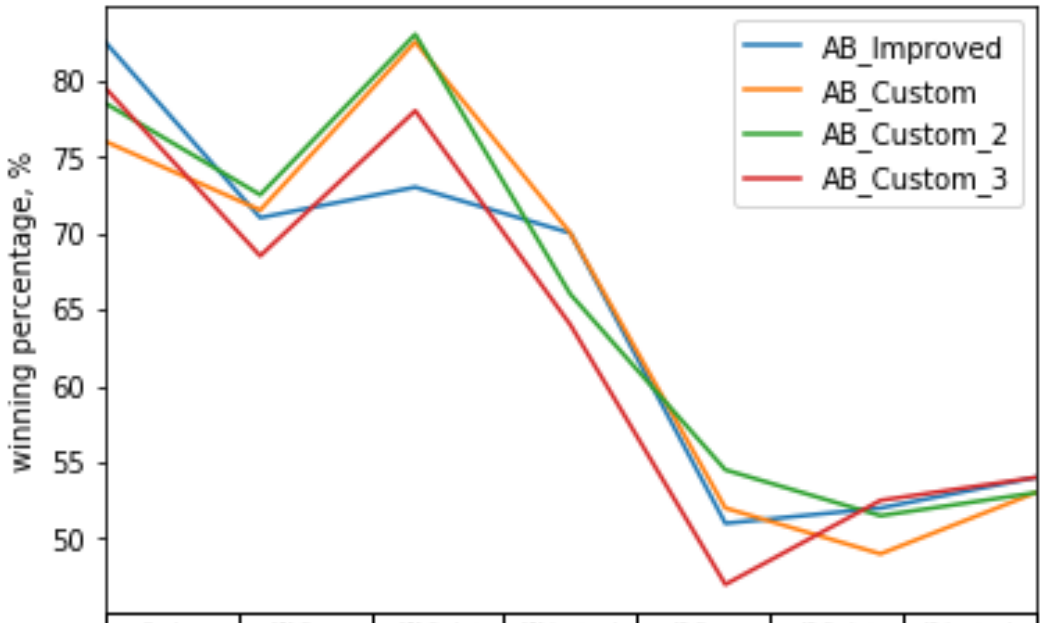
Adversarial search utilizing minimax method with alpha-beta pruning for isolation game.

Game_agent.py file contains coded minimax() and alphabeta() functions along with their helper min_value() and max_value() functions. Note, that each helper function checks for len(legal_moves), if that equals zero with a call to a score function and terminates recursion.

Custom score functions: custom_score(), custom_score_2() and custom_score_3() have the common basis from improved_score(), meaning len(legal_moves)-len(opponent_legal_moves). The difference are: Custom_score() uses multiplication factor of2 for the number of opponent moves. Custom_score_2() is similar to Custom_score() but has an additional booster. The booster checks if there are any moves for both players that land them on the same cell on the game board. A score for such moves is boosted up significantly. The check is only performed after half of the board is filled. Custom_score_3() increases the score for moves that land in cells that are heavily surrounded by blocked cells. The increase only happens at the beginning of the game: first 18 moves for 7*7 size board. I came up with this heuristic from playing that game on the physical board. During the game I noticed that when the board is semi-full there are left empty cells in the center of the board and they all connected to each other by a horse move.

Tournament was run with 100 games (200 total against a pair of players). Raw results presented below:





	Random	MM_Open	MM_Center	MM_Improved	AB_Open	AB_Center	AB_Improved
AB_Improved	82.5	71.0	73.0	70.0	51.0	52.0	54.0
AB_Custom	76.0	71.5	82.5	70.0	52.0	48.0	55.0
AB_Custom_2	78.5	72.5	68.0	66.0	54.5	51.5	53.0
AB_Custom_3	79.5	68.5	78.0	64.0	47.0	52.5	54.0

Figure shows the last run pf tournament.py. Y-axis is wins/totalgames. Findings were:

1. Alpha-beta pruning algorithm dominated over the minimax.
2. MM_Center game agent was not particularly good as it lost about ¾ of games.
3. Let's explore in more detail Custom_score_2 and Custom_score_3() playing against AB_Open (109 and 94 wins respectively). Is that a significant difference to claim that Custom_score_2 is better compared to Custom_score_3? P-value being about 0.04, indicates that yes, Custom_score_2 is better than Custom_score_3 playing against AB_Open.
4. Same question as in 4 but against AB_Improved(): 106 and 108 wins. P-value ~ 0.8. Statistically cannot confirm that Custom_score_3 is better than Custom_score_2 while playing against AB_Improved.
5. Recommend using Custom_score_2 evaluation function as in some cases it proved to be better.