Heuristic analysis

The following table summarizes the comparison of the improved heuristic versus student heuristics. The number of rounds played was increased to 50. The table shows percentage of total wins and wins/losses ratio per heuristic.

Opponent	ID_Improved	Improved_Optimized	Opposite	Location
Total	75.71%	75.36%	76.71%	79.50%
Random	190/10	191/9	186/14	187/13
MM_Null	163/37	171/29	176/24	173/27
MM_Open	148/52	145/55	145/55	144/56
MM_Improved	141/59	132/68	141/59	152/48
AB_Null	157/43	161/39	161/39	166/34
AB_Open	135/65	131/69	130/70	144/56
AB_Improved	126/74	124/76	135/65	147/53

The first heuristic I tried is just an optimization on the improved heuristic. I noticed the redundant call to `get_legal_move` function which can be done only once in the heuristic function to speed up the search and increase the depth. The results show no improvement.

The second try was to do the opposite of what the improved heuristic does - in the beginning of the game prefer boards with minimum number of moves for your agent. The idea is to cover the board on the outside in the beginning of the game and to move to the center in the end when iterative deepening is able to go all the way to the terminal nodes.

```
if game.move_count < (game.width * game length) / 3:
  return (opp_moves - my_moves)
else:
  return (my_moves - opp_moves)
```

The results are in the Opposite column and not much different from ID_Improved.

The last heuristic takes the same idea to focus in a specific part of the board in the beginning of the game by adding the product of location coordinates to the score according to this formula:

```
(my_moves - opp_moves) + (locx * locy) / move_count
```

I divide it by `move_count` to decrease the importance of location feature as game goes on. This heuristic is called Location in the table and that's the one I decided to submit as my custom_score.

To summarize I recommend to use the Location heuristic because it showed slightly better performance than others.