**Build a Game-Playing Agent**

**Project – Part 1: Heuristic Analysis**

Student: Bruno G. Eilliar

On my agent I tested the following heuristics:

1) An Aggressive Chaser: similar to what we had in lectures, but the score function allow to weight the number of opponent move: CustomPlayerMoves – **w**\*OpponentMoves;

2) A second heuristic (couldn’t come up with a name yet) that evaluate the number of moves over percentual occupation of the board: CustomPlayerMoves/BoardOccupation;

3 ) A third heuristic (couldn’t come up with a name yet) that evaluate the number of available moves for both players, the taxicab distance between players and the percentual occupation of the board: CustomPlayerMoves + Distance – BoardOccupation – OpponentMoves;

From the grah below, it’s possible to notice that the 3rd heuristic performs better than the other two, winning 73,57% of the matches, followed by the 2nd heuristic, winning 66,43% of the matchess. The aggressive chaser with a **w** = 2 performs poorly and on some cases (as bellow) even worse than the ID\_Improved agent, winning only 58,57% of the matches against 61,43% of the former.

As a result, even though I couldn’t come up with my goal of reaching > 85%, the third heuristic seems to be prety reasonable, combining the features described it seems to perform a strategy to keep distance from the opponent and look for states in which the board is less occupied but also trying to make the opponent loose some ground. I’m still trying to use weights on those features from the 3rd heuristic, but until now the best results came from the described combination with weights ecqual to 1.

The next table shows the complete results.

