

Report on Adversarial Game Playing Agent

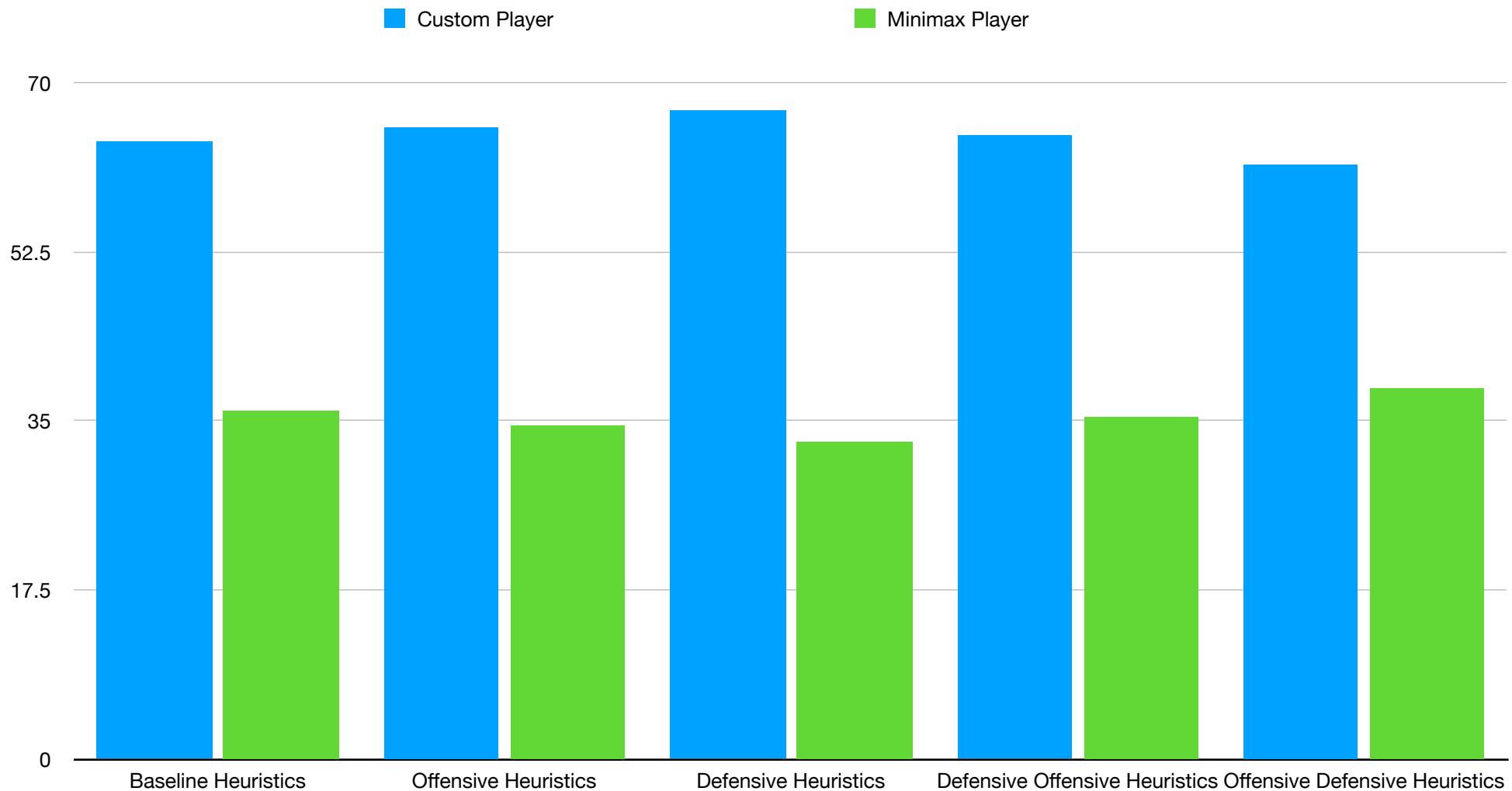
**Prepared By
Rajiv Krishnan R**

Experiment Results

The Custom Game Playing agent has been programmed with following strategies

- 1) Minimax Algorithm for decision making to select optimal move for the player
- 2) Alpha-Beta pruning to decrease the number of nodes to be evaluated by the Minimax Algorithm
- 3) Iterative deepening strategy for Time and Space efficiency
- 4) Multiple heuristic score calculations to find the performant function

The results of the Game play against a Minimax agent is presented below. The results are of Custom player with different heuristic score calculation



What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

I have used the depth component of iterative deepening while computing the heuristic score for the move selection.

I have experimented with different heuristic style like Defensive Offence, where the player will have more move options initially and during later stages tries to minimise and block opponent movements.

In Offensive Defence the player will be more aggressive in the initial moves trying to block and minimise opponent movements and goes into defensive mode.

In both the above experiments I have used the depth value used for Iterative deepening to control offensive and defensive movements.

Analyse the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

The maximum search depth achieved by my agent is level 4. The search speed has impact on the accuracy of heuristic performance, as more importance to search speed can cause the agent to make poor move decisions since the depth knowledge gets limited and hence the feedback received from the child states won't be the reality.

Similarly the more deep the agent can experiment with, more accurate will be the decision making.

But there needs to be a balance, where more importance to speed can result in poor decision making and importance to exhaustive depth search can cause the game to timeout, so the best possible solution would be to decide on a realistic and limited depth search using techniques like iterative deepening depth search which can provide the playing agent a near valid options to make a optimal move.