

## MP2 Report

In this report, we implemented the minimax algorithm for the game: dots and boxes. Minimax itself can become costly when going deep into the search space, and so we also implemented a flavor of minimax, one with alpha beta pruning. Alpha beta pruning utilizes a heuristic based upon the nodes that it had already visited and helps determine whether it's even worth to traverse further down the path.

The implementation of minimax was pretty straightforward. Depending on the level, get the min or max of the level below. I tested around with different depths, and found that the model seemed to perform best when the search space was shallow as opposed to deeper which could imply that a more greedy strategy does give some good results. One interesting thing that I did deploy was a random move picker that picks a move at random when there aren't any particular best options (ie. beginning of the game). As the search goes on, it replaces the initial move with a more optimal move if the search finds one. At the end, I chose to leave my minimax level to be 1 because it performs best, but it will consistently perform well up to around 99 depth. It seemed to start overseeing locally optimal solutions of easy points when the tree got too deep.

Moving forward into alpha beta pruning minimax, I found that there was not much pruning action when the tree was shallow, and would become more and more prevalent as the tree grew to above 5 levels. As the levels got deeper, more pruning happened — which makes sense. Other than that, the performance of this algorithm paralleled to that of minimax, but lost sometimes against the brute force minimax. That being said, the time to compute was significantly less for alpha beta minimax compared to that of minimax with the same depth search.

The evaluation function seems to be a fun part of the game. Attempting to define our own ideas of how well a move is to be picked can be either simple or complex. I chose to tackle a relatively simple evaluation function of taking the depth of the node and multiplied it with the score of the move.

As a fun activity, I replaced the simple Player 1 in file1.GUI.py with my minimax algorithm and put it up against my alpha beta pruning. After playing a couple games, they seem to perform comparable. Minimax would sometimes win by a point or 2.