CSE3521 SP21
John Choi.1655
Austin Schall.37

1c. It plays the way that I expected and the cpu player is able to look at each empty position and determine the best move. After several tries, we realized that the cpu player is unbeatable and results in a draw or a loss every attempt.

Evaluated 59705 states, expanded 31973 states
Evaluated 927 states, expanded 486 states
Evaluated 61 states, expanded 37 states
Evaluated 5 states, expanded 3 states

We noticed that after every turn, the states being evaluated/expanded were becoming significantly less because the cpu has less spots to look at.

1e. This also plays the way we expected and it's still impossible to beat.

Evaluated 10557 states, expanded 6239 states
Evaluated 157 states, expanded 91 states
Evaluated 53 states, expanded 33 states
Evaluated 5 states, expanded 3 states

The states evaluated/expected vs the original minimax was significantly less, meaning that using alpha-beta pruning in minimax is a lot more efficient.

1f.
Minimax Expanded Nodes: 294778
Minimax-Pruning Alpha-Beta Expanded Nodes: 73647

Standard minimax algorithm is expected to expand on more nodes compared to the Alpha-Beta pruning minimax algorithm because unlike the Minimax-Pruning algorithm, the standard minimax algorithm goes through all nodes before making calculations. Minimax-Pruning algorithm is able to omit some nodes that are guaranteed to have a score lower than the root.

---- After changing move_expand_order -----

Minimax Expanded Nodes: 294778
Minimax-Pruning Alpha-Beta Expanded Nodes: 53627

The original move_expand_order will have no impact on the standard minimax algorithm as expected because the standard minimax algorithm goes through each and every node no matter what the ordering is. On the other hand, Minimax-Pruning Alpha-Beta's node expansion

value is changed because the ordering matters when the pruning happens (It may be able to find the node with the highest projected score before scanning others).