

CSCI 3202: Project Intermediate Report

Students: Jason Hunter, Tristan Barnes

Date: 11/14/25

Basic Minimax AI Player vs Random at varying depths:

We built an AI player that uses basic minimax with no alpha-beta pruning. Running 100 simulations of this AI against our random player resulted in the following stats:

Depth (plies)	AI Win %	Random Win %	Avg. Moves to Win
5	97%	3%	29.9

Minimax vs Random with a depth of 5 plies:

Then we played 100 games with the random player against the minimax AI player at a depth of 5 plies.

Metric	Value
Number of games	100
Avg. Time per Game	3.502 seconds
AI Win %	97%
Random Win %	3%
Avg. Moves to Win (AI)	30.5

As you can clearly see, the AI player is significantly better than the random player. It practically never loses. On average, the AI wins the game in 30.5 moves. This makes sense because the random player is literally just choosing any available pit to play while the minimax AI is algorithmically building out the best possible solutions at each turn up to 5 moves out. Effectively, the AI can almost always outplay the random player. The times when the random player wins can almost entirely be attributed to chance.

Minimax with Alpha-Beta AI Player with 5-ply depth:

Here we use a version of the AI player that utilized minimax with alpha-beta pruning and ran it for 100 simulations at a depth of 5 plies against the random player. The results were as follows:

Metric	Value
Number of games	100
Avg. Time per Game	0.586 seconds
AI Win %	99%
Random Win %	0%
Tie %	1%
Avg. Moves to Win (AI)	30.3

The AI player that uses Alpha-Beta pruning manages to win 99% of the time. It tied in the one game it did not win. The average time per game to run to completion was 0.586 seconds. Each move played by the AI (using 5 ply-depth) took on average 0.038 seconds. On average it takes 30.3 moves to win. The results for this are honestly similar to the basic minimax AI player. In this case, the alpha-beta pruning player one more game and never technically lost, only tying a single game. The difference is that the run time at similar ply depths is significantly faster in this implementation. This makes sense because both are using similar algorithms, but naturally, alpha-beta pruning is an extension on the simple minimax that is optimized further to reduce the number of branches that need to be visited.

Minimax Alpha-Beta AI Player with 10-ply depth:

At 10-ply depth, which took forever (nearly two hours) to run to completion for 100 simulations, these were the results when paired up against a random player:

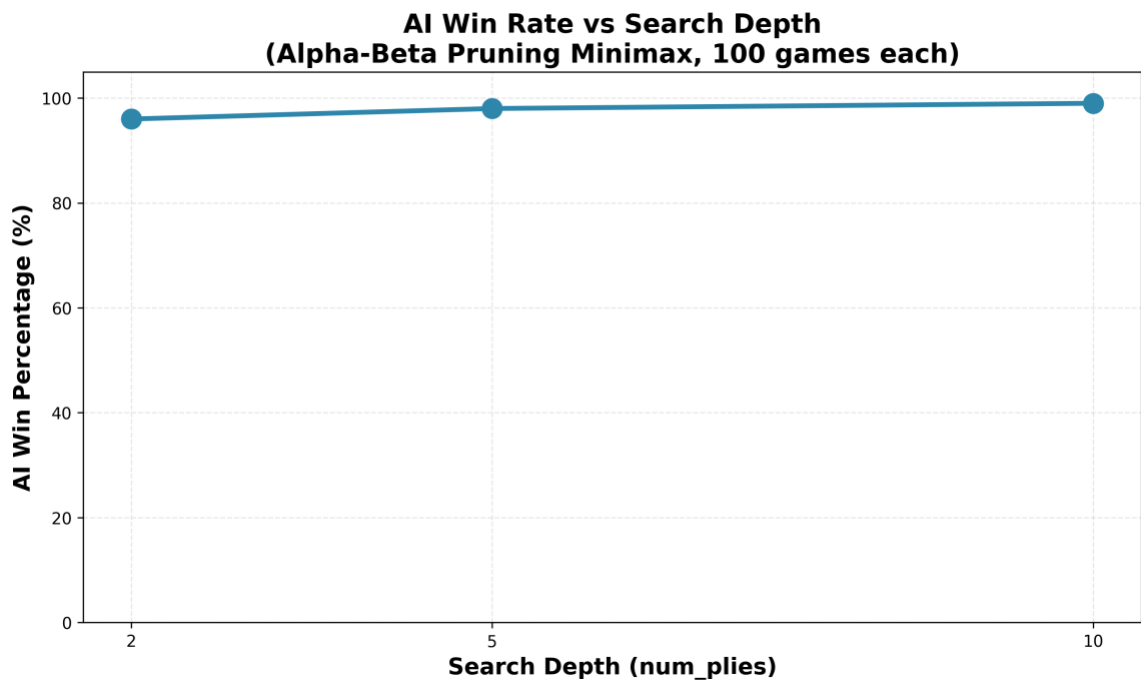
Metric	Value
Number of games	100
Avg. Time per Game	66.735 seconds
AI Win %	100%
Random Win %	0%
Tie %	0%
Avg. Moves to Win (AI)	28.5

When comparing the runtimes for the minimax alpha-beta player with that of the basic minimax player both using a depth of 5 plies, the runtime for the optimized alpha-beta player is significantly faster and more efficient. The basic AI player's average game length ran for 3.502 seconds while the alpha-beta pruning AI player's average game length was just 0.586 seconds. This is nearly six times faster.

Considering the way that these algorithms seem to scale, and the fact that at 10-ply depth, the alpha-beta pruning minimax AI, which is the faster of the two, took 111 minutes to run 100 games to completion, I can imagine that the basic minimax AI would take an unfeasible amount of time to run 100 games to completion.

Win Percentage vs Search Depth:

Here is a curve that plots these win percentages vs ply depth:



Here you can see that as the number of plies is increased, the AI win percentage also increases. They seem to be correlated, but further testing is necessary for confirmation. This makes sense because as the number of plies is increased, the complexity with which AI attempts to find an optimal move increases. The AI will search deeper into future game states and as a result, is more likely to make the best move, ensuring that it wins as many games as possible.

Depth (plies)	AI Win %
2	96%
5	98%
10	99%