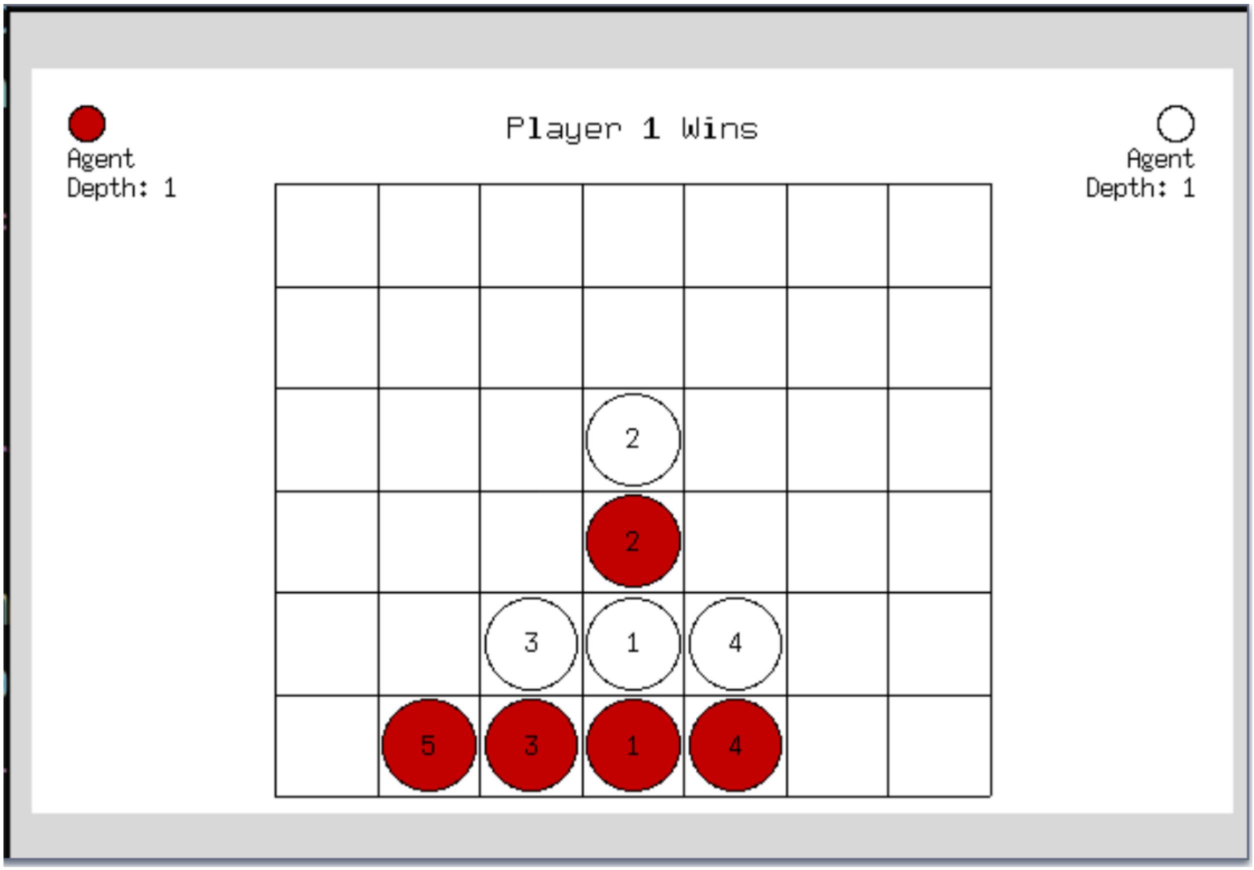
CS170 Project 3

Alex Totah
atota002
862368232

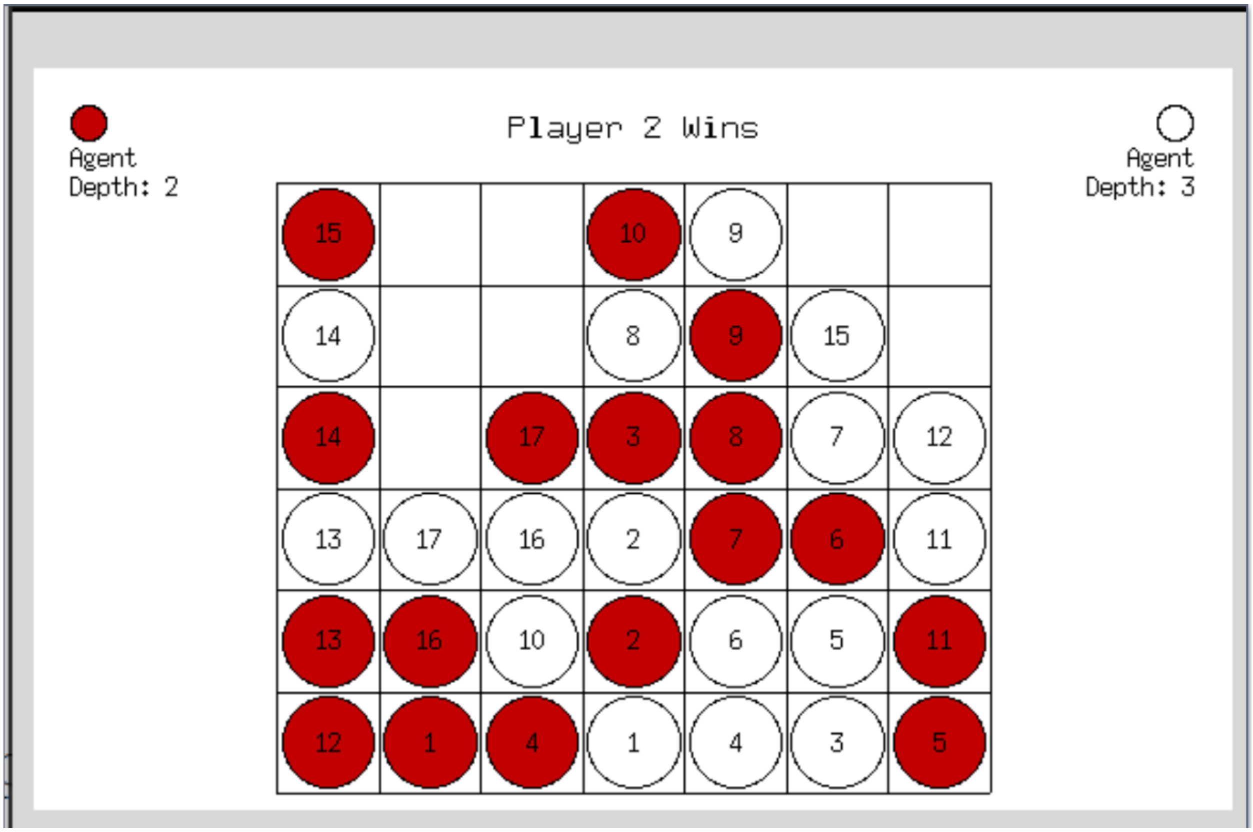
Minimax Algorithm

1v1



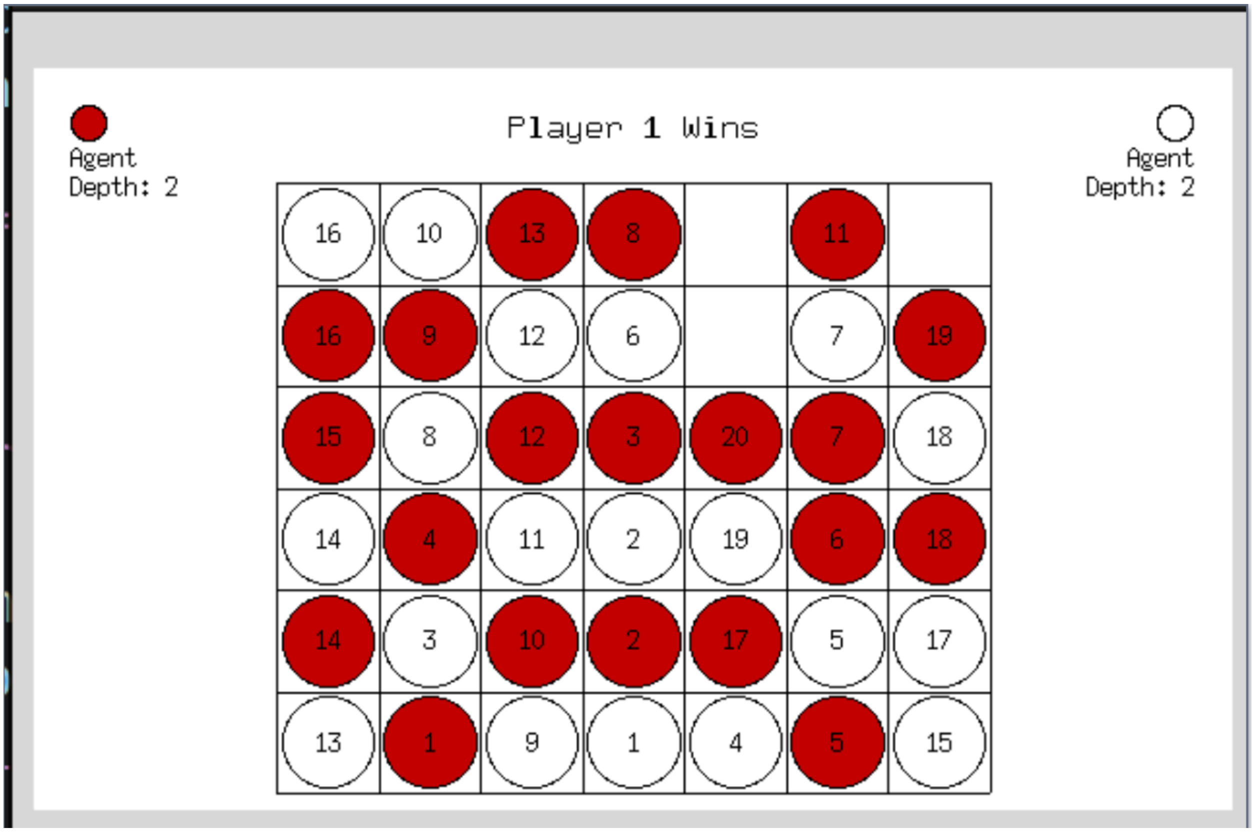
The algorithm seemed to only play where the opponent had played. This resulted in a win for player one.

2v3

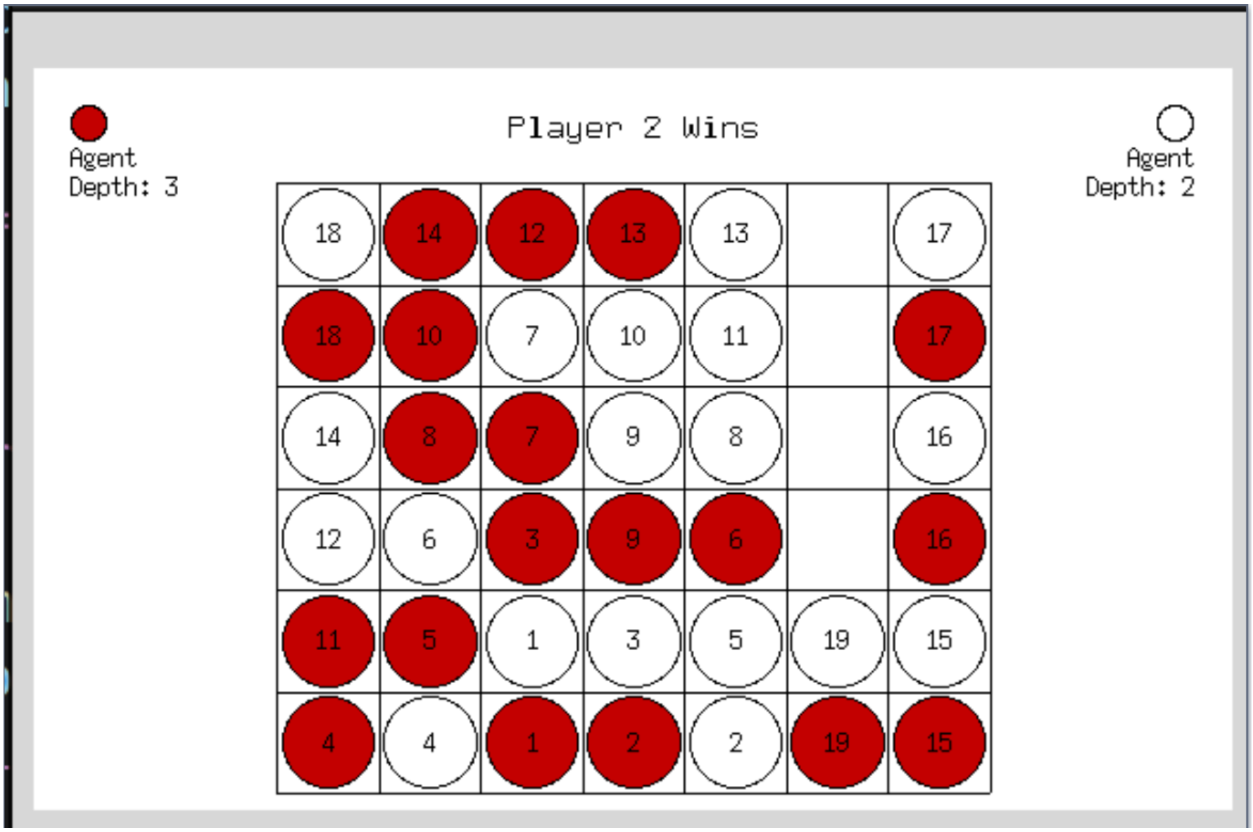


This example shows that depth only controls the search and assumes that a player is choosing from that same tree. However, when two players of different depths play against each other, they choose different paths, so the result is not necessarily optimal.

2v2


Agent
Depth: 2
Player 1 Wins
Agent
Depth: 2
16
10
13
8
11
16
9
12
6
7
19
15
8
12
3
20
7
18
14
4
11
2
19
6
18
14
3
10
2
17
5
17
13
1
9
1
4
5
15

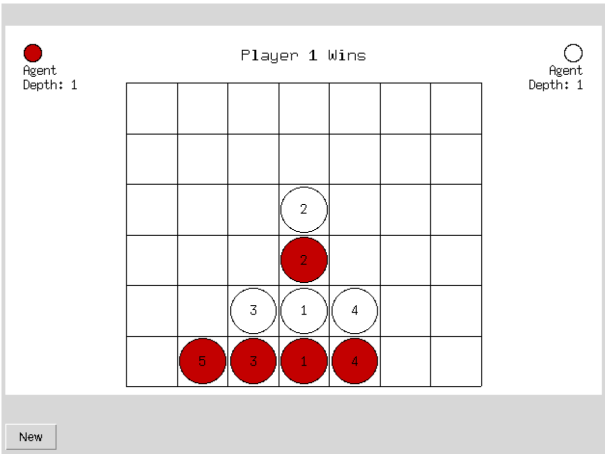
3v2



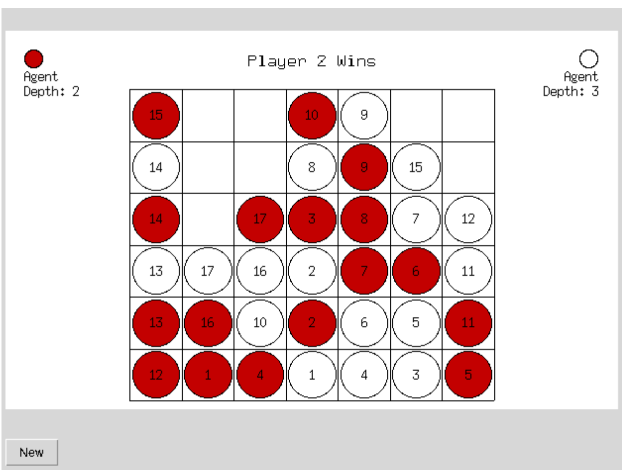
Notes on minimax: Minimax took quite some time at the higher depths. This is because of the exponential complexity of minimax without pruning. Since all paths are being explored, this algorithm alone is not efficient to compute the optimal path quickly.

Alpha Beta Pruning

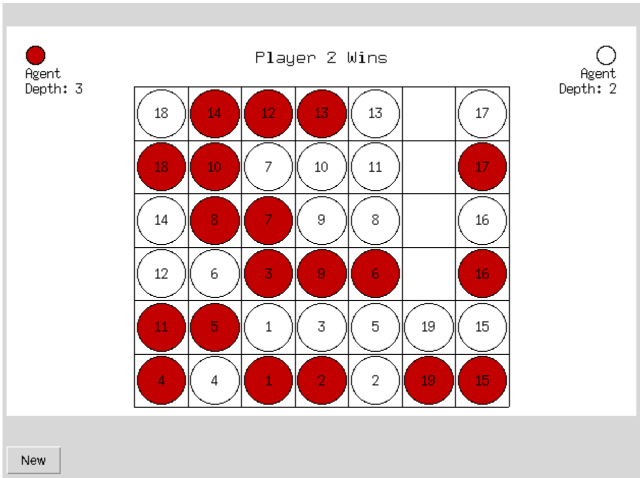
1v1



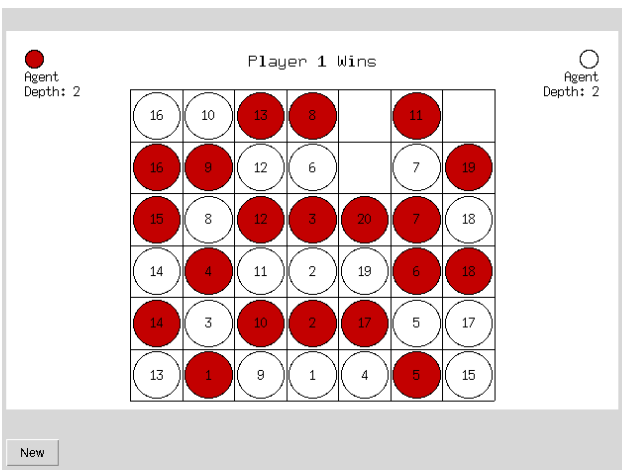
2v3



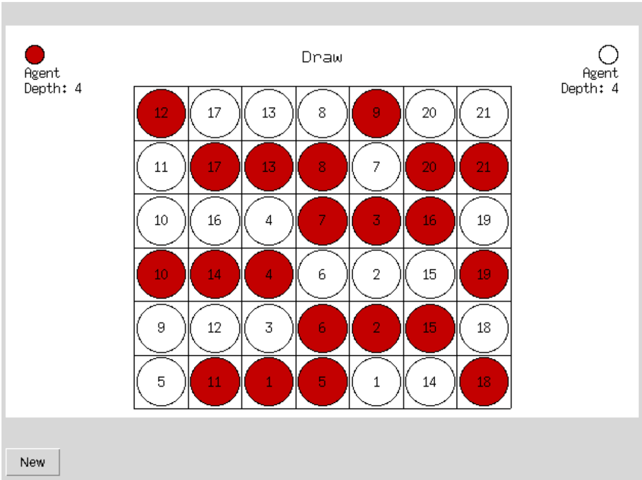
3v2


Player 2 Wins
Agent
Depth: 3
Agent
Depth: 2
New

2v2

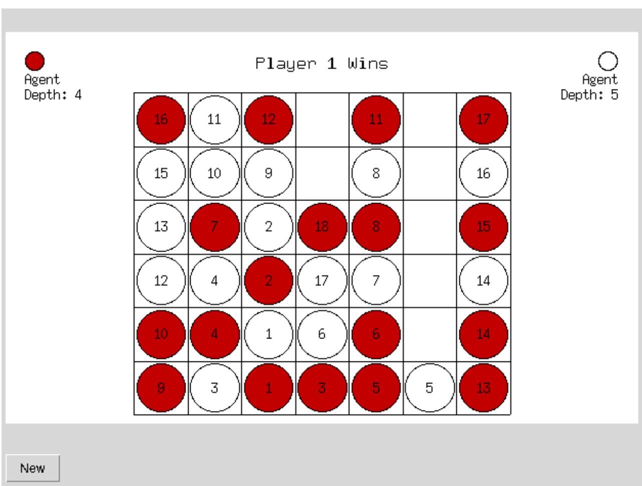

Player 1 Wins
Agent
Depth: 2
Agent
Depth: 2
New

4v4 Comparison



Pruning time: 2.49s
Minimax time: 7.43s

4v5 comparison

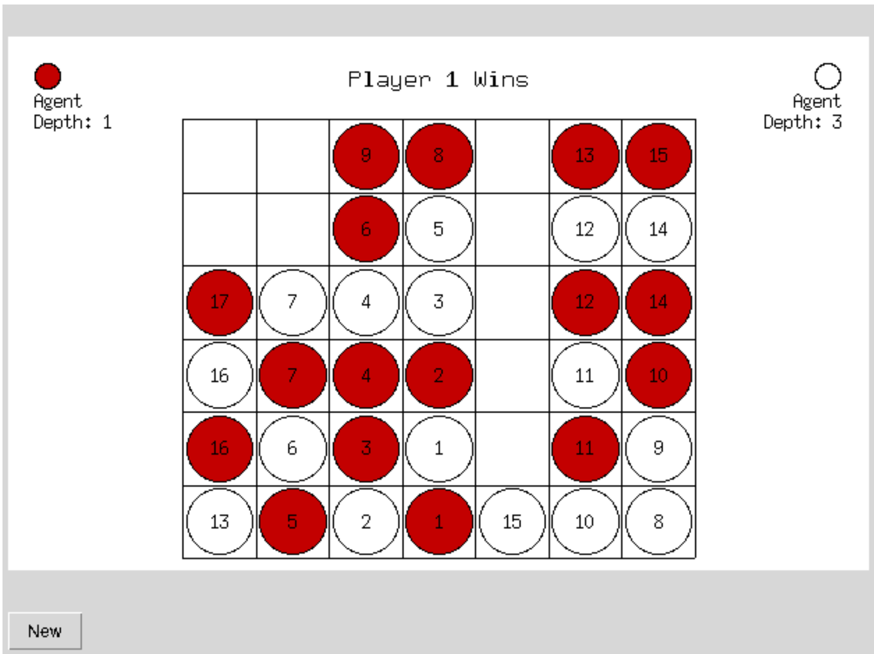


Pruning time: 5.39s
Minimax time: 15.23s

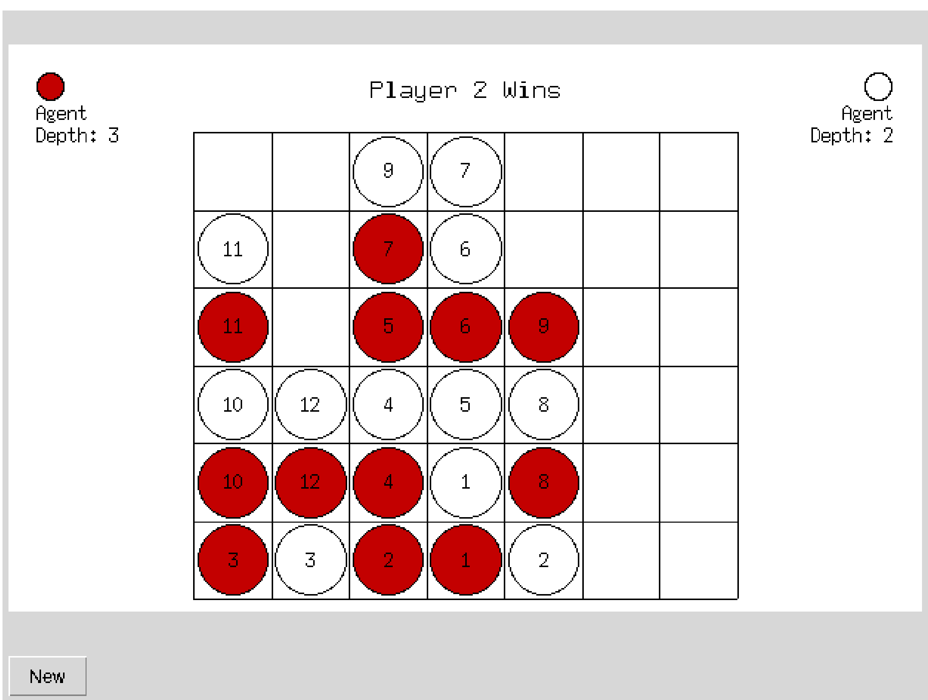
Notes on alpha beta pruning: The results on the original minimax algorithm and the alpha beta pruning algorithm are the same with improved runtime. For the 4v4, pruning was able to improve the time by 66%. For the 4v5, pruning was able to improve the time by 65%.

Expectimax

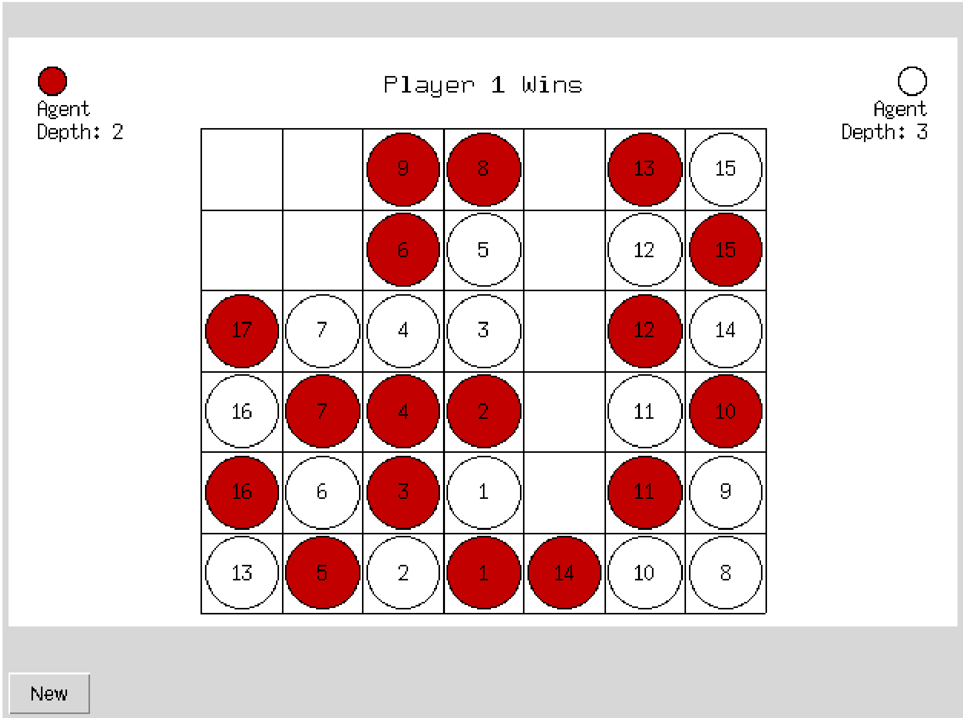
1v3



3v2



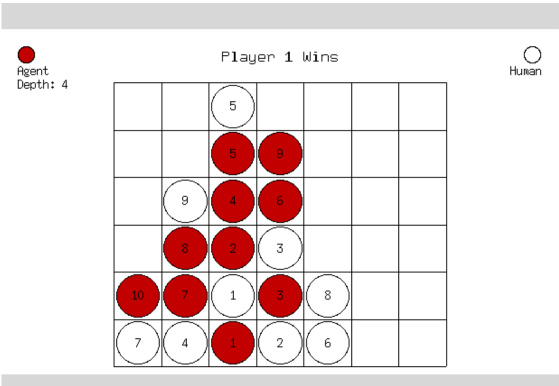
2v3

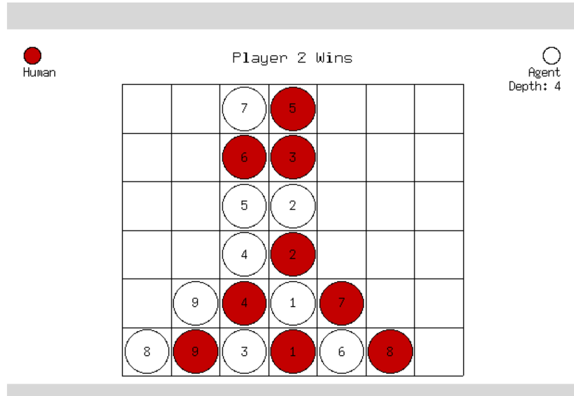


The Expectimax algorithm demonstrates clear behavioral differences across depth levels, with deeper searches providing substantially better performance against random players through improved strategic planning and multi-move threat recognition. Against random opponents, shallow depth agents rely primarily on immediate tactical responses like blocking obvious threats, while deeper agents can establish complex winning combinations and positional advantages. Expectimax also takes a noticeably more risky path.

Algorithm Recommendation

Human vs. AI





I would recommend Alpha-Beta pruning with depth 4-5 for optimal performance against human opponents.

Human players represent strategic opponents who make deliberate, goal-oriented decisions rather than random moves, which immediately rules out Expectimax as the optimal choice. Between Minimax and Alpha-Beta, both use identical decision-making logic (assuming the opponent plays optimally), but Alpha-Beta provides the crucial advantage of computational efficiency through pruning, allowing deeper search within the same time constraints.

Unlike Expectimax, which might make overly aggressive moves assuming random opponent responses, Alpha-Beta's conservative "assume optimal opponent" approach leads to solid, defensible play that humans find challenging to exploit.

Improvements on Evaluation

The current function doesn't distinguish between different types of threats.

Winning moves (immediate 4-in-a-row opportunities)
Forced moves (must block opponent's immediate win)
Double threats (positions that create multiple winning opportunities)

4. Connectivity Scoring

- Reward pieces that are connected or have potential for connection, even in segments with opponent pieces. The current function ignores mixed segments entirely.

- Lower pieces are generally more valuable as they're harder to block. Add a height-based multiplier that favors pieces closer to the bottom of the board.

- Consider whose turn it is and whether moves force opponent responses. Some positions are stronger when it's your turn versus the opponent's turn.

- Use different evaluation strategies based on game phase - opening focuses on center control, endgame focuses on immediate threats.