Comp 472 Mini Project 2

DONOVAN UPSDELL - 40133717

JULIEN PICARD - 40158060

XAAVIAN ALI - 40082861

Heuristic e1

- Tallies the lines horizontally, vertically, and diagonally for each player
- Computes value based on the amount of these lines

Heuristic e2

- Searches through the board to evaluate all the open tiles as possible moves.

- Checks what lines can be formed to determine the value of those open tiles for each player as future moves.

- Ignores anything not directly connected to an open tile in a way that can form a line

- Branches out in every direction from that tile, evaluating both opposite directions to see what lines can be formed from that tile, summing that value up along with the a negative value for the opposing player

- Sums up these values for every open tile to give a heuristic to represent the value of the potential future moves

Scoreboard 4435 (d1 = 6, d2 = 6, MiniMax)

	E1 (5 wins)	E2 (0 Wins)
Average Evaluation Time Per State	3.39815e-05s	2.64866e-05s
Total States Evaluated	243938	204313
Average of Per-Move Average Depth	4.4998830281	5.9999725915
Total Evaluations at each Depth	{1: 9, 2: 1, 3: 0, 4: 7, 5: 5, 6: 243916}	1: 2, 2: 0, 3: 1, 4: 0, 5: 2, 6: 204308
Average of Per-Move Average Recursion Depth	3.9954102032	4.6726250601
Total Number of Moves	5	4.5

We see that for game 4435, e1 won 5 while e2 won none. e1 was able to evaluate more states than e2. An interesting thing to note is that e1 won every time it stated, and the game tied every time e2 started.

Scoreboard 4431 (d1 = 6, d2 = 6, AlphaBeta)

	E1 (5 wins)	E2 (5 Wins)
Average Evaluation Time Per State	4.31299e-05s	3.94293e-05s
Total States Evaluated	9099	3406
Average of Per-Move Average Depth	4.5	6.0
Total Evaluations at each Depth	{1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 9099}	{1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 3406}
Average of Per-Move Average Recursion Depth	1.2223889607	1.9097712631
Total Number of Moves	4.5	4.5

Here, both heuristics won an even amount of games. Similarly to the first game, every the heuristic that started would win the game. This may be a result of the small board size and limited available moves.

Scoreboard 5441 (d1 = 2, d2 = 6, AlphaBeta)

	E1 (4 wins)	E2 (1 Wins)
Average Evaluation Time Per State	5.16467e-05s	3.95302e-05s
Total States Evaluated	320	105622
Average of Per-Move Average Depth	1.8181818182	4.1980790731
Total Evaluations at each Depth	{1: 0, 2: 320}	{1: 36, 2: 39, 3: 29, 4: 21, 5: 4, 6: 105493}
Average of Per-Move Average Recursion Depth	0.0909090909	1.7821954505
Total Number of Moves	8.7	8.4

For this game, e1 has a smaller depth than e2. This is reflected by the low total states evaluated by e1. E1 was able to win more games despite this lower depth.

Scoreboard 5445 (d1 = 6, d2 = 6, AlphaBeta)

	E1 (1 win)	E2 (1 Win)
Average Evaluation Time Per State	4.5009e-05s	4.24566e-05s
Total States Evaluated	228560	175090
Average of Per-Move Average Depth	4.3635317267	4.2
Total Evaluations at each Depth	{1: 16, 2: 8, 3: 1, 4: 5, 5: 3, 6: 228527}	{1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 175090}
Average of Per-Move Average Recursion Depth	1.3379713013	1.4808931412
Total Number of Moves	9.7	9.7

This game is similar to the previous one, except with more time to compute and equal depths. Each heuristic won 1 game each, with the other games being ties. We can note that the average of per-move average depth are very similar, and that they are both below 6, meaning that often there was not enough time to compute everything.

Scoreboard 8551 (d1 = 2, d2 = 6, AlphaBeta)

	E1 (10 wins)	E2 (0 Wins)
Average Evaluation Time Per State	0.0001248027s	0.0001116623s
Total States Evaluated	4092	129401
Average of Per-Move Average Depth	2.0	5.9362350333
Total Evaluations at each Depth	{1: 0, 2: 4092}	{1: 660, 2: 627, 3: 486, 4: 456, 5: 58, 6: 127114}
Average of Per-Move Average Recursion Depth	0.0044444444	4.9713627161
Total Number of Moves	9.2	8.7

In this game, we have a much larger board. e1 was able to win all games here, even with the lower depth. This may indicate that e1 is simply a better algorithm, or that e2 needs more depth of computation time to be effective.

Scoreboard 8555 (d1 = 2, d2 = 6, AlphaBeta)

	E1 (10 wins)	E2 (0 Wins)
Average Evaluation Time Per State	0.0001222397s	0.0001112295s
Total States Evaluated	5729	470883
Average of Per-Move Average Depth	2.0	5.9889362743
Total Evaluations at each Depth	{1: 0, 2: 5729}	{1: 517, 2: 461, 3: 32, 4: 336, 5: 7, 6: 469530}
Average of Per-Move Average Recursion Depth	0.0045045045	4.9689656773
Total Number of Moves	9.3	8.8

This game was almost identical to the last one, except we have s=5 instead of 1. The results are pretty much the same, with e1 winning every game.

Scoreboard 8651 (d1 = 6, d2 = 6, AlphaBeta)

	E1 (5 wins)	E2 (3 Wins)
Average Evaluation Time Per State	0.0001247862s	0.0001085724s
Total States Evaluated	92957	97746
Average of Per-Move Average Depth	5.9270631148	5.9341076066
Total Evaluations at each Depth	{1: 552, 2: 539, 3: 501, 4: 126, 5: 119, 6: 91120}	{1: 506, 2: 495, 3: 446, 4: 252, 5: 68, 6: 95979}
Average of Per-Move Average Recursion Depth	4.9774272703	4.9776633318
Total Number of Moves	16	15.8

Here, both heuristics have the same depth again. In this case, the games were a lot closer with e1 winning 5 and e2 winning 3. All of the stats are quite close in value.

Scoreboard 8655 (d1 = 6, d2 = 6, AlphaBeta)

	E1 (6 wins)	E2 (4 Wins)
Average Evaluation Time Per State	0.0001261791s	0.0001011559s
Total States Evaluated	679152	859494
Average of Per-Move Average Depth	5.9895359377	5.9908129584
Total Evaluations at each Depth	{1: 699, 2: 594, 3: 266, 4: 217, 5: 46, 6: 677330}	{1: 706, 2: 690, 3: 256, 4: 462, 5: 84, 6: 857296}
Average of Per-Move Average Recursion Depth	4.9205703543	4.6422803246
Total Number of Moves	16.8	16.7

In the final game, we increase the computation time to 5 seconds. Similarly to the previous game, the score was pretty close with e1 winning 6 and e2 winning 4.

Effects of depth, time, and board size

- With varying depth, we've noticed that when e1 was given less depth, it won more often than when it had a greater depth.
- With varying time constraints, we did not notice that it greatly affected the outcomes of the games.
- With larger boards, we notice that less draws happen. This may be because with so many more moves to make, the algorithms may follow a branch that seems good, but ends up losing if you were to search deeper, making it harder to defend against. This is in contrast to smaller boards where limited moved make it easier to see if each move is a winning one or not, being closer to end states, making it easier to defend against

Use of alphabeta

The use of alphabeta in this is beneficial to completing the search to the given depth. This is because it prunes branches that will not contribute to the solution. This allows for a deeper search, quicker than a minimax. This speed is important because of the time constraint given to the heuristics.

Conclusion

Based on the results of our experiment, we can conclude that e1 was a better choice for this game, winning x% of games, and tying x%. We did notice that with increased depth, e1 was becoming less effective, and that it won more games when its depth was at 2 compare to 6. Due to e2 being the supposed more complex algorithm, the results may have been different if allowed more time for e2 to calculate.