COMP30024 - Project Part B Report
Team Members: Connor Cates, Jad Saliba

The Approach

For our algorithm, we decided to use a modified Alpha-Beta min max algorithm feuled by book learning. To accomplish this, we first built a random algorithm. The initial random bot listed out every possible action a player could take and randomly selected an action from the overall list. We ran 1000 trials and recorded each action of both players, and the result of the game. In an 8x8 grid used to represent the board, every move that was taken resulting in a in for that player added +1 to the location on the grid associated with that move, and -1 for every location associated with a move resulting in a loss (+/-0 in the result of a draw). This book was then used as the first iteration. This form of book learning was inspired by the tutorial discussion of applying this technique to a game of Tic-Tac-Toe (Knots and Crosses).

We then created a 2-ply algorithm that tried out each move for our player, recording the value associated with that location in the book. We sorted the potential moves we found for the player in order of highest book value to lowest, creating a priority queue of moves to search the best moves for ourselves first. Next using a temporary state we calculated the optimal move for the opponent, storing the value associated for their move in the grid. The action chosen was the move with the best difference between our move value in the book and the opponents, maximizing our move in relation to minimizing the opponents. To decide whether or not to perform a 'boom', we first tested whether or not that boom action would result in a win, if so stop searching and execute that action. Additionally, any action that resulted in an improvement of the ratio between the number of our tokens and the number of their tokens, we executed whichever boom resulted in the best new ratio.

We then ran 1000 more trials and recreated the book with the new algorithm used to score the book location values. This was repeated multiple times, slowly improving the accuracy of the book we have created. After that we investigated the best weighting scheme for the book learning algorithm, linear weights (each move is +/- 1) versus exponential weights (each move, n, is weighted +/- n^2). The record of the two algorithms over 1000 trials against the random bot are as follows:

	2-Ply Record
Linear Weights	Win: 776 Loss: 0 Draw: 224
Exponential Weights	Win: 652 Loss: 0 Draw: 348

Figure 1.

The linear weights had about a 12% increase in win percentage over exponential, so we decided to continue with the linear weighting scheme.

Timing each game, the 2-ply approach took on average a total of 1.5 seconds for the entire game's decision making. Because this was well under the allotted time specifications, we improved our algorithm to a 4-ply search. The same algorithm design was applied, testing our move, finding the opponents best, and repeating both steps to search two moves further. The 4-ply algorithm on average totalled 8 seconds per game for decision making, and showed a significant improvement on win percentage, with a new win percentage of 94.8%.

Additional Optimization

To optimize our algorithm, we used the theory of the Optimal Stopping Problem described in the book *Algorithms to Live By,* by Brian Christian and Tom Griffiths. The theory states that when deciding when to stop searching in a scenario when you cannot go back to a previously searched node, one should search through 37% of the sample size and then take the next best option found. In our case, we can go back to previously searched nodes. In our algorithm, we incorporated this by searching through 37% of the potential moves, and then choosing the first node that is better according to our evaluation. Best case the first option after the 37% is the best of all the moves, worst case it is the very last move. On average, it should be somewhere in the middle and thus pruning the number of branches we search down in our search tree. This was tested with our 2ply algorithm and the results are as follows:

	37% Cutoff	2-ply Record
Linear Weights	Yes	Win: 789 Loss: 0 Draw: 211
	No	Win: 776 Loss: 0 Draw: 224
Exponential Weights	Yes	Win: 683 Loss: 0 Draw: 317
	No	Win: 652 Loss: 0 Draw: 348

Figure 2

As shown, the combination of linear weights, paired with the 37% optimization from the Optimal Stopping Problem resulted in the best record. Once we applied our 4-ply algorithm with these constraints, the record improved to:

Wins: 948
Loss: 0
Draw: 52

Thus we came to our final algorithm with a win rate of 94.8% against a random opponent.

Overall Effectiveness and Performance Assessment

	37% Cutoff	2-ply Record	4-ply Record
Linear Weights	Yes	Win: 789 Loss: 0 Draw: 211	Wins: 948 Loss: 0 Draw: 52
	No	Win: 776 Loss: 0 Draw: 224	
Exponential Weights	Yes	Win: 683 Loss: 0 Draw: 317	
	No	Win: 652 Loss: 0 Draw: 348	

Figure 3

Figure 3 (above) summarizes the overall effectiveness and assessment we used in deciding our algorithm. The 4-ply algorithm we chose to submit was chosen based on the record against the random opponent. Our algorithm runs in Big-O complexity of *O(n^2)* where n is the number of potential moves the player has each turn, and the decision making for the entire game averages about 8 seconds.