Muhammad Tuzcu
Tom Nguyen
Vivien Nguyen

Final AI Project - Proposal

Our group proposes to do our final Artificial Intelligence project on the Dots and Boxes game. It is a pencil and paper game that usually involves 2 players in which each player draws lines and tries to get the most number of boxes. Therefore each player draws lines in a way that the opponent cannot draw boxes so that they can maximize the number of boxes created from themselves.

Our project will involve a human playing against the computer where the computer attempts to select the best move to win. The algorithm used for this is the minimax algorithm. The way the program and the minimax algorithm works with this game is that the winner is determined only when all the boxes are filled on the world state. From there we are able to tally up the total accumulated boxes each player made and compare to see who created more boxes. Whoever created the greater amount of boxes wins. Otherwise, if both players have the same number of boxes it is a tie.

The minimax algorithm will return the highest utility achievable by the max player. To solve the best move, a function will be implemented to count the number of boxes created in a single move for the player's turn. Then, a utility function will calculate the sum of the total boxes accrued by the minimizer and the maximizer. If at least a box is created, the current player will go again in the minimax algorithm, whether it is the min or max player. In a sense, this has a cascading effect on the minimax tree where the min and max player are not full turn based but is dynamic based on the outcome of the current move. It will return a 0 until it gets to the end where there is another function to check to see if the entire world state is filled. It then compares the number of boxes accumulated for the maximizer versus the minimizer and selects the best move with the most number of boxes.

Performance:

We first proposed the structure to be at most 3 by 3 boxes, meaning it would have at most 24 edges to make. However, this would cause a time complexity of O(e!) for the # of open edges after each move. With a time complexity of O(24!), it would be too large. This is due to the recursive nature of the minimax algorithm. So we may start with 4 boxes first (9 dots and 12 edges giving at most O(12!) for each recursion) in order to prevent the program from breaking. Or we may need to use depth limited search.