CSD311 : ARTIFICIAL INTELLIGENCE
PROJECT REPORT
TEAM 12 :DOTS AND BOXES GAME

PROF. Snehasis Mukherjee

Kushagra Aggarwal : 2110110831
Aditya Doneria : 2110110030
Harshdeep Singh : 2110110236
Aditya Mukerji : 2110110032
Akshat Saihgal : 2110110933
Pranav Uppal: 2110110382

Introduction :

Dots-and-Boxes is one civilian chess popular in America. It is a point matrix game played by two persons. The chessboard of Dots-and-Boxes is shown in Fig. 1. Fig.1 indicates the description of the Dots and Boxes board, it is a 6*6 dots matrix or 5*5 boxes matrix, and it can be another size. The rules of the game are that each player connects adjacent dots on alternate turns, it is required that the player should only connect adjacent dots and should not make cater corners; the lines, which are also called edges, do not belong to any side, and players only need to take boxes into account. The player can get a box by connecting the four adjacent dots with lines, and when one player makes a box, he(or she) can put his(or her) name in it, and move again, when all boxes are formed, the game is over, and the winner is the player who gets the most boxes.

Figure 1. Dots-and-Boxes chessboard

Problem analysis :

- Predictable strategies:Many players tend to fall into predictable patterns, especially in the opening moves, making it easier for opponents to anticipate their next moves and counter effectively.
- Complexity vs. Accessibility: The game's simplicity in rules often masks its strategic depth. New players might find it easy to learn but challenging to master, leading to a disparity in skill levels among players.
- First Move Advantage: The player who starts the game has a slight advantage in controlling the initial board structure, potentially dictating the flow of the game. This can make it challenging for the second player to catch up.

- Endgame Scenarios: As the game progresses, it becomes crucial to foresee and plan for endgame scenarios. Assessing potential box closures and calculating the risk of leaving open ends can be complex and challenging.
- Limited Variability: The size of the board and the fixed rules may limit the variability of gameplay, leading to repeated patterns or certain strategic paths that experienced players can exploit.

Solver Ideation :

Edges: To organise and keep track of these edges, you can utilise structures like lists, graphs, or arrays.
When a player completes a square's fourth edge, a box is created. A data structure to monitor owned or finished boxes is required to represent boxes. This might be a data structure, such as a matrix, that is connected to the grid and indicates which boxes belong to which player.
Details about the player: Variables or structures can be used to hold data about the players, their scores, and their turns. This might provide information about who is taking turns, how many boxes each player has claimed, and their overall score.
Player Data: Structures or variables can be used to hold data on the players, their scores, and their turns. This might provide information about who is taking turns, how many boxes each player has claimed, and their overall score.
Game State: It is usual practise to handle the overall state of the game by using a structure that encompasses all aspects, including the grid, edges, boxes, and player data. With this framework, it is simple to manipulate and update the state of the game after every action.
Graph Representation: The Dots and Boxes game may be represented as a graph data structure for more complex algorithms such as pathfinding, AI solvers, or graph-based tactics. Every junction is a node, while lines or connections are represented by the edges that separate intersections.

Goal Ordering :

Goal ordering within the Dots and Boxes game means organizing moves according to their potential to accomplish particular objectives. Rather than treating all moves equally, this method arranges them based on the likelihood of achieving strategic goals. For example, prioritizing moves that immediately secure a box or block the opponent from completing one. This strategy allows players to concentrate on tactics that maximise their own box closures while hindering the opponent's advancements, resulting in a more strategic and deliberate gameplay style.

Deadlock Conditions Used:

Deadlock conditions in Dots and Boxes occur when certain strategic situations arise that limit or prevent players from making optimal moves, leading to a stalemate or inevitable advantage for one player. These conditions often occur in scenarios where:

1. Forced Box Completion: A move forces the opponent to complete a box, providing them an advantage, while preventing the opponent's move results in a similar disadvantage.

2. **Closing-in Strategy**: Players strategically create patterns that force the opponent to complete boxes, thereby gaining control of the board or setting up advantageous future moves.

3. **Lack of Alternatives**: When there are very few or no moves left that don’t hand the opponent an immediate advantage, resulting in limited options and an inevitable loss or draw for the player.

Deadlock conditions can significantly influence the game's outcome, making it crucial for players to anticipate and navigate these situations strategically to avoid being at a disadvantage or facing a stalemate.

Algorithms Used :

MINIMAX ALGORITHM :

The minimax algorithm in Dots and Boxes explores potential moves in a game tree, with each player seeking to maximize their score while minimizing the opponent's. It begins from the current state, examining all possible moves, and evaluating outcomes based on factors like completed boxes and future opportunities. Through recursive evaluation, it selects the move that maximizes the AI's score while assuming optimal play from both sides. This process culminates in the AI choosing the move that offers the highest potential score, considering the opponent's counter-strategies.

ALPHA BETA PRUNING :

Alpha-beta pruning is an enhancement to the minimax algorithm, streamlining the evaluation of a game tree by disregarding certain branches that won't influence the final result. It uses two values, alpha and beta, representing the minimum and maximum assured scores for players. By discarding inferior branches, the algorithm significantly reduces the number of nodes evaluated, enhancing efficiency in complex games like Dots and Boxes. Both techniques enable AI players to make strategic moves while conserving computational resources, considering potential responses from opponents to select optimal moves.

Code Flow:

We first create a class which contains the whole game and necessary functions aside from the testing function. We intialize the board with size of our choice which is 10x10 as per the question while rendering middle rows, verticals and ends in different functions such as to avoid unnecessary out of the board moves. Then the alphabeta algorithm takes charge after the human as the alphabeta player which makes the then best move according to the algorithm.We also adhered to the reference hence making a game controller which assigns player1 as human and player2 as the AI player which can be changed at any point of time. The Dots are marked with numbers and you draw a dash between the numbers by giving a start and end point at time of input and claimed boxes are marked for the player(human) to keep track. The game utilizes is_over to track when all available moves have already been played and then declares a winner by counting the no. of boxes claimed by each player.

REFERENCES :

- https://www.researchgate.net/publication/301454149_Technique_Analysis_of_Dots-and-Boxes
- https://link.springer.com/chapter/10.1007/978-3-642-21783-8_3
- http://www.jsoftware.us/vol7/jsw0702-02.pdf
- https://sapientia.ualg.pt/bitstream/10400.1/18772/1/Dots_and_Boxes.pdf