A REPORT

ON

Tic-Tac-Toe Game

By

Group 7

Name of the student	Registration No.
Yaram Sandeep Reddy	AP21110011538
Teja Kukatla	AP21110011553
Nandini Ganthi	AP21110011526
Bickey kumar Jaiswal	AP21110011279
Dhanushree Yamala	AP21110011491
Gayatri Naga Prishita Kapuganti	AP21110011542

Prepared in the partial fulfillment of the

Artificial Intelligence Lab (CSE 413L)

Submitted to

Dr. Anirban Bhar
Assistant Professor
Department of Computer science and Engineering

SRM UNIVERSITY, AP

Table of contents

1 Problem Statement

Implement an agent that will play tic-tac-toe game against human. The agent will try to choose action to maximize the chance of winning the game.

2 Abstract

The minmax algorithm is a fundamental decision-making tool in game theory, widely utilized to determine optimal moves for players in games. Specifically, in the context of tic tac toe, the minmax algorithm can be employed to create a computer player capable of playing the game at an optimal level. This algorithm involves evaluating the game state, generating potential moves, and recursively applying the algorithm to select the move that leads to the best possible outcome for the player.

By alternating between maximizing and minimizing scores for each player, the algorithm simulates potential moves and selects the one that offers the most favorable result for the current player. Through the implementation of the minmax algorithm, the computer player can strategically make decisions to enhance its chances of winning the game. Furthermore, this algorithm's versatility extends to various other applications, including decision-making in games like chess, poker, and financial forecasting.

3 Introduction

3.1 What is Tic-Tac-Toe Game

The familiar paper-and-pencil game tic tac toe is popular across players of all ages. Two players alternately mark spaces on a 3x3 grid to play this game. "X" has been given to one player, and "O" to the other. The goal is to get three of your marks in a row before your opponent, whether they are horizontal, vertical, or diagonal.

3.2 How to Play

3.2.1 The Board

A 3 x 3 grid is used to play the game. It is a game of turns for each player to mark a square with their symbol ("X" or "O").

3.2.2 Game Play

The board is empty when the game begins. By placing their symbol on an empty square, players take turns marking it.

3.2.3 Winning

The round lasts until the board is full or until one player obtains three of their symbols in a row (horizontally, vertically, or diagonally).

3.2.4 Draw

The player wins if they can line up three of their symbols in a row! The game ends in a draw if the board fills up before either player achieves it.

3.3 involving in AI Play

User will play against an AI opponent in this version of the game who was created to challenge you. The AI chooses its moves based on a strategic algorithm (Use minimax algorithm to select the best possible action for the agent), trying to win or force a draw as much as it can.

3.4 Instructions

In order to place your "O" on the board, enter the row and column numbers (1–3). By placing your marks strategically to get three in a row and stopping the AI from doing the same, attempt to beat the AI.

4 Formulation of the Problem Statement

4.1 Define the game state: Represent the tic tac toe board as a 3x3 grid, where each cell can be empty, occupied by the player's symbol (X or O), or occupied by the opponent's symbol.

4.2 Generate possible moves: For a given game state, generate all possible moves that the current player can make. This can be done by iterating through the empty cells on the board.

4.3 Evaluate the game state: Assign a score to the game state based on how favorable it is for the current player. For tic tac toe, a simple evaluation function can be used to assign a score to each game state, such as +1 for a win, -1 for a loss, and 0 for a draw.

4.4 Apply the minmax algorithm: Recursively apply the minmax algorithm to all possible moves, alternating between maximizing and minimizing the score for each player. This involves simulating the game for each possible move and selecting the move that leads to the best outcome for the current player.

4.5 Make the best move: Once the minmax algorithm has been applied to all possible moves, the computer player selects the move that leads to the highest score.

5 Representation of a Tree Diagram

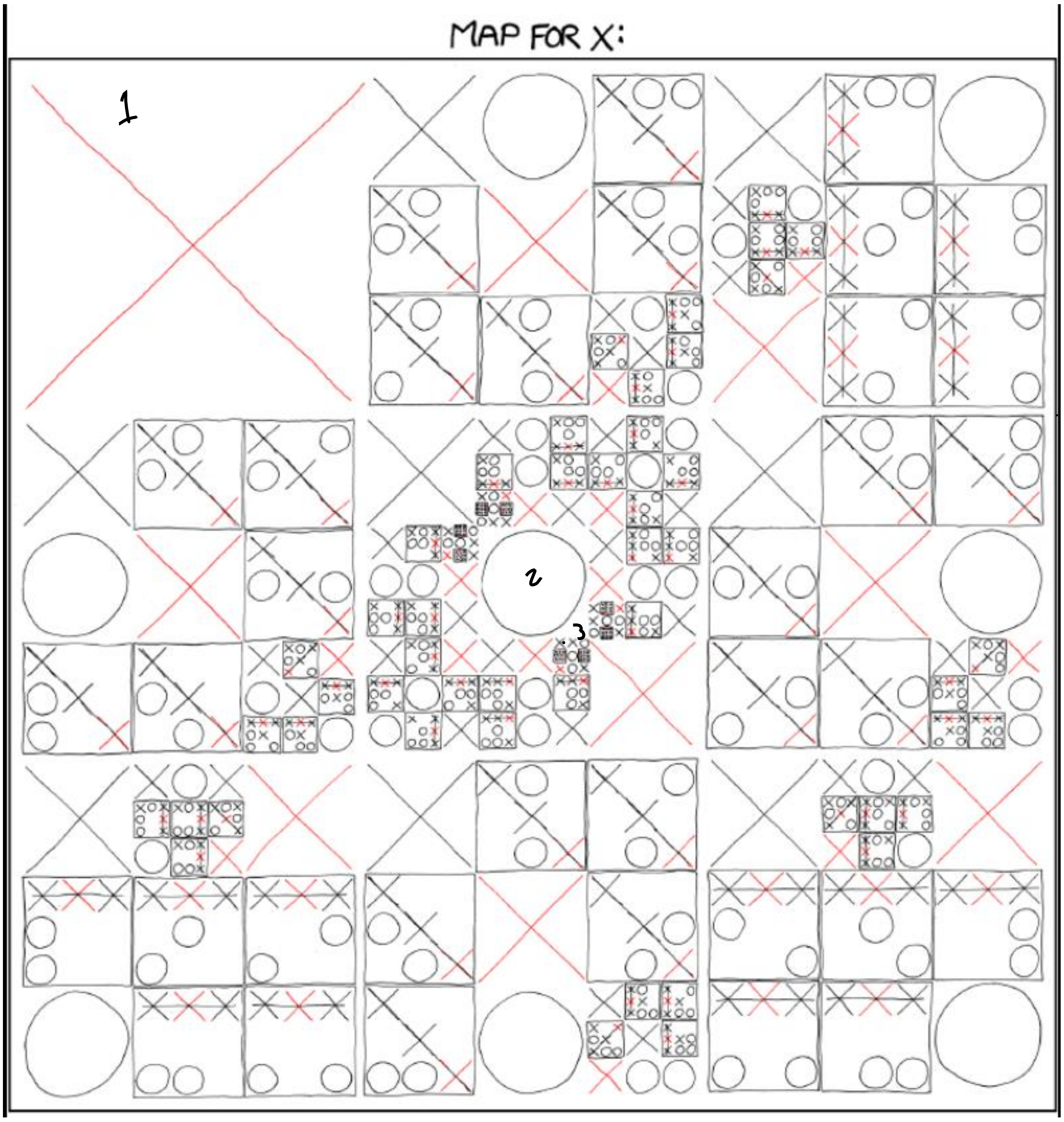
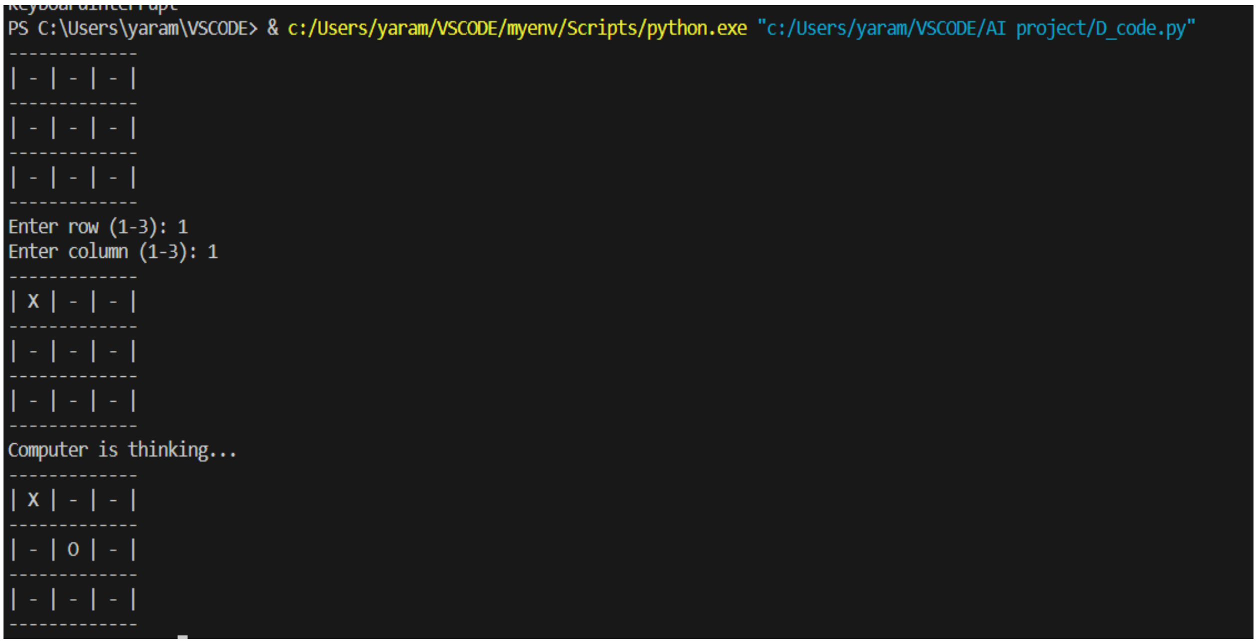


Fig.1 This is one of the possible trees if we keep (1,1) as X.

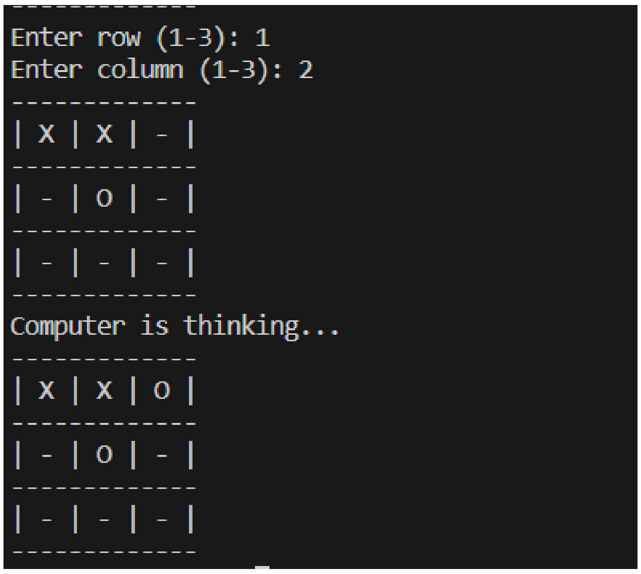
6 Output:

6.1 Tie between user and computer



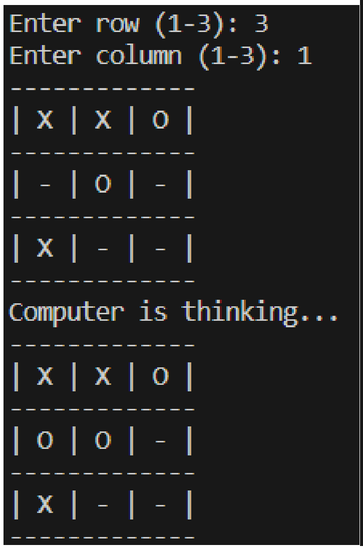
```
PS C:\Users\yaram\VSCODE> & c:/Users/yaram/VSCODE/myenv/Scripts/python.exe "c:/Users/yaram/VSCODE/AI project/D_code.py"
-------------
| - | - | - |
-------------
| - | - | - |
-------------
| - | - | - |
-------------
Enter row (1-3): 1
Enter column (1-3): 1
-------------
| X | - | - |
-------------
| - | - | - |
-------------
| - | - | - |
-------------
Computer is thinking...
-------------
| X | - | - |
-------------
| - | O | - |
-------------
| - | - | - |
-------------
```

1.We choose (1,1)
2.AI taken (2,2)

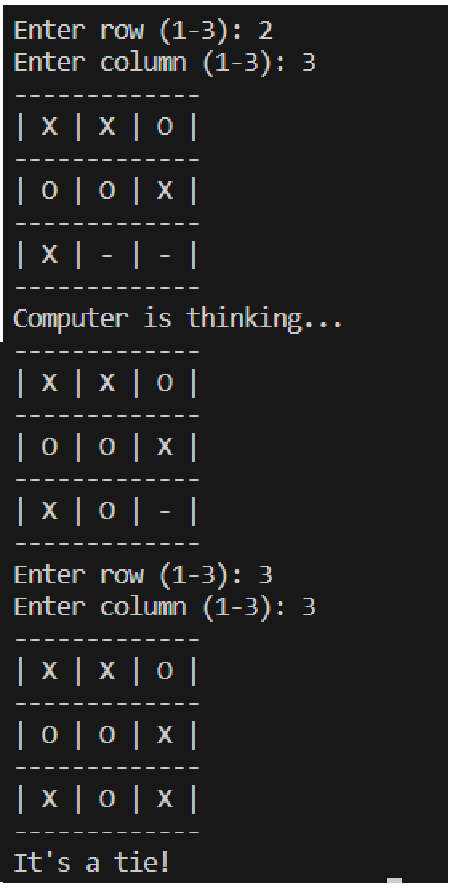


```
-------------
Enter row (1-3): 1
Enter column (1-3): 2
-------------
| X | X | - |
-------------
| - | O | - |
-------------
| - | - | - |
-------------
Computer is thinking...
-------------
| X | X | O |
-------------
| - | O | - |
-------------
| - | - | - |
-------------
```

3.We choose (1,2)
4.AI taken (1,3)



```
Enter row (1-3): 3
Enter column (1-3): 1
-------------
| x | x | o |
-------------
| - | o | - |
-------------
| x | - | - |
-------------
Computer is thinking...
-------------
| x | x | o |
-------------
| o | o | - |
-------------
| x | - | - |
-------------
```



```
Enter row (1-3): 2
Enter column (1-3): 3
-------------
| x | x | o |
-------------
| o | o | x |
-------------
| x | - | - |
-------------
Computer is thinking...
-------------
| x | x | o |
-------------
| o | o | x |
-------------
| x | o | - |
-------------
Enter row (1-3): 3
Enter column (1-3): 3
-------------
| x | x | o |
-------------
| o | o | x |
-------------
| x | o | x |
-------------
It's a tie!
```

3.We choose (3,1)
4.AI taken (2,1)

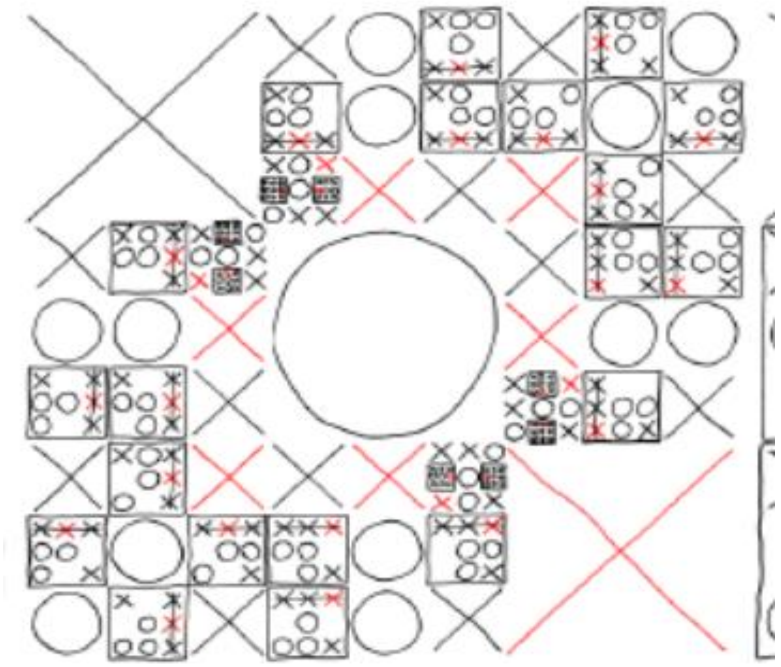
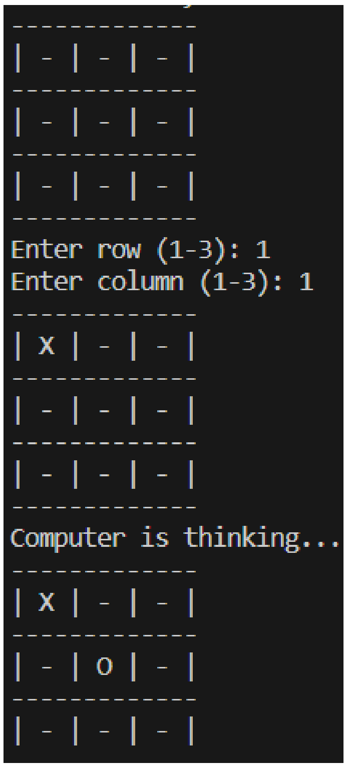
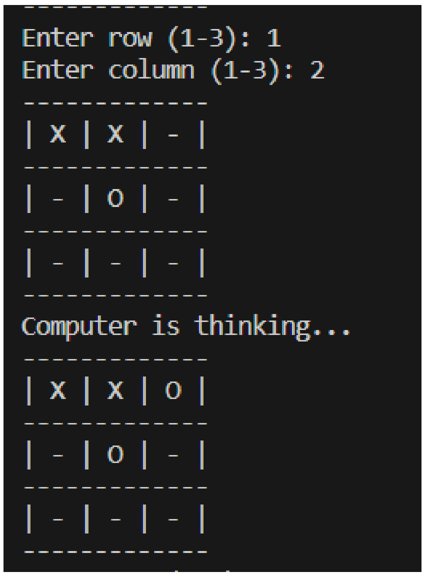


Fig.2 Tree for first X possibilities to win X.

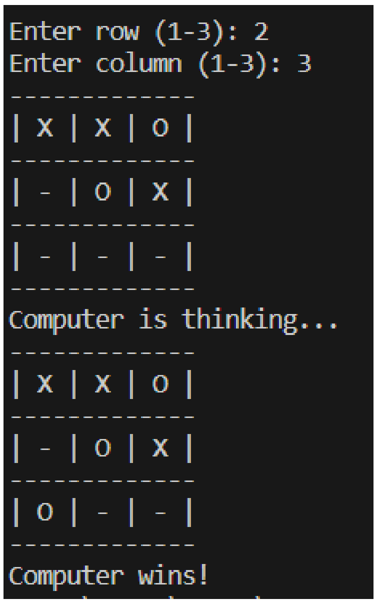
2nd Output: The AI is winning



```
-------------
| - | - | - |
-------------
| - | - | - |
-------------
| - | - | - |
-------------
Enter row (1-3): 1
Enter column (1-3): 1
-------------
| X | - | - |
-------------
| - | - | - |
-------------
| - | - | - |
-------------
Computer is thinking...
-------------
| X | - | - |
-------------
| - | O | - |
-------------
| - | - | - |
```



```
-------------
Enter row (1-3): 1
Enter column (1-3): 2
-------------
| X | X | - |
-------------
| - | O | - |
-------------
| - | - | - |
-------------
Computer is thinking...
-------------
| X | X | O |
-------------
| - | O | - |
-------------
| - | - | - |
-------------
```



```
Enter row (1-3): 2
Enter column (1-3): 3
-------------
| X | X | O |
-------------
| - | O | X |
-------------
| - | - | - |
-------------
Computer is thinking...
-------------
| X | X | O |
-------------
| - | O | X |
-------------
| O | - | - |
-------------
Computer wins!
```

Above figures are moves of Ai and user.

7 Conclusion

Despite its apparent simplicity, the game of Tic Tac Toe offers an engaging way to investigate the fundamentals of artificial intelligence. This timeless game, which illustrates ideas like search algorithms, strategic thinking, and decision-making, acts as a basic AI playground.

We've explored the intersection of AI and gaming with this Tic-Tac-Toe simulation by adding an AI opponent through the use of the minimax algorithm. The algorithm's capacity to compute and optimize movements shows how powerful AI is for developing strategies and coming to intelligent choices in a competitive setting.

8 References

1. HarvardX CS50AI CS50's Introduction to Artificial Intelligence with Python.
2. Elaine Rich and Kevin Knight, “Artificial Intelligence”, McGraw-Hill, 3rd edition, 2017.
3. 2. E Charniak and D McDermott, “Introduction to Artificial Intelligence”, Pearson.