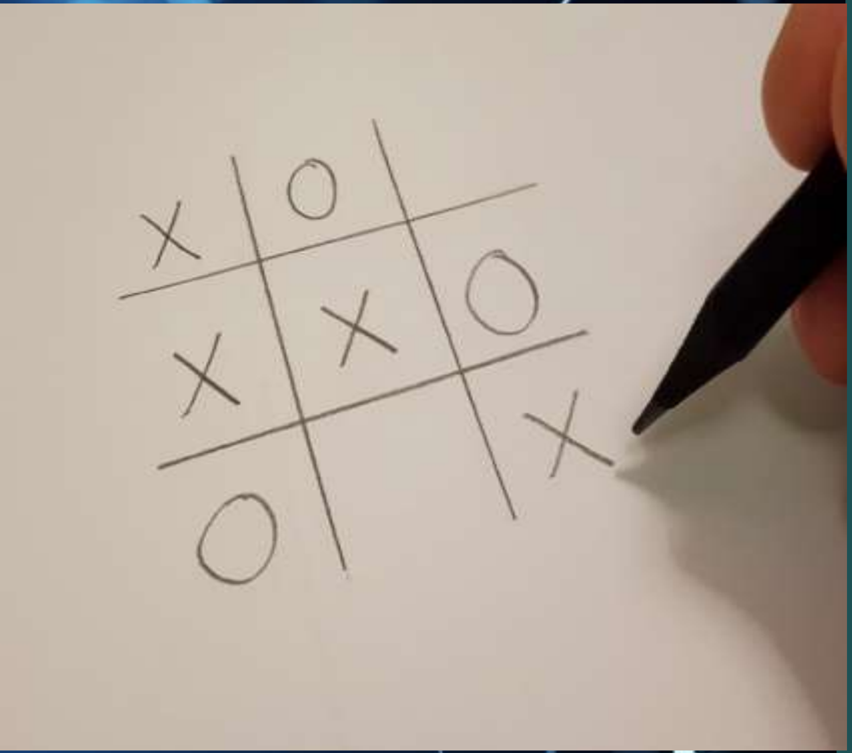


IMPLEMENTATION OF TIC-TAC-TOE

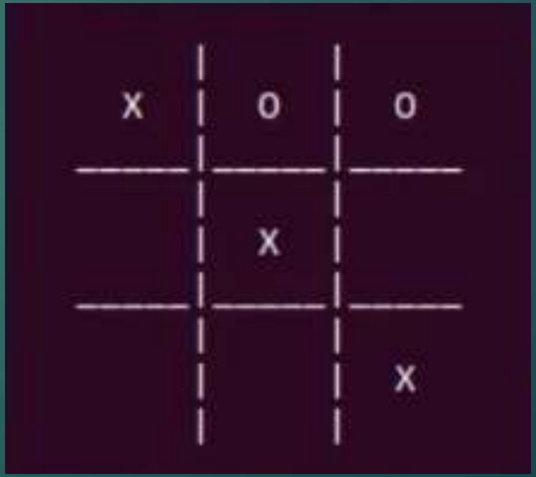
SILAS CHAPPIDI

PRANAY SIKA

SAI SRIJA BANDARU

PROJECT OBJECTIVE

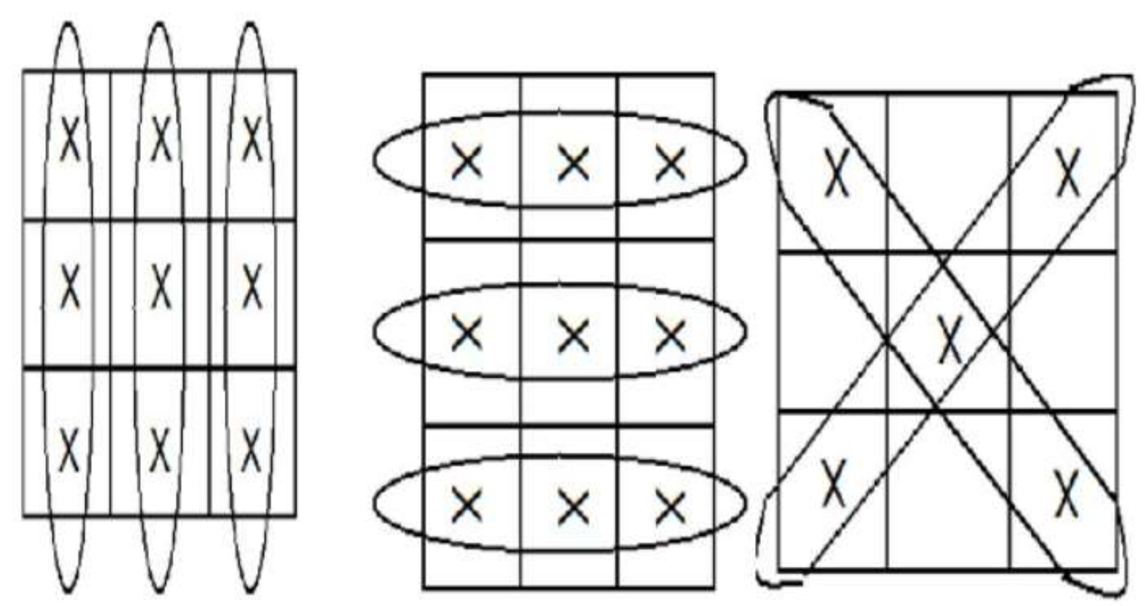
- Our aim is to focus on building two agents that play against each other in a Tic-Tac-Toe game
- User must choose 2 different agents that are going to play the game
- There are 3 different possibilities to end this game:
  - Agent1 wins against Agent2
  - Agent2 wins against Agent1
  - There can be a Tie between both agents



```
     |     |
  X  |  O  |  O
_____|_____|_____
     |     |
     |  X  |
_____|_____|_____
     |     |
     |     |  X
     |     |
```

AGENDA

- The goal for each agent is to win the game by forming a horizontal, vertical, or diagonal line of all X or all O in a grid in which each agent plays one after the other.
- The second goal is to ensure that your adversary is unable to create an X or O pattern since this is a zero-sum game.

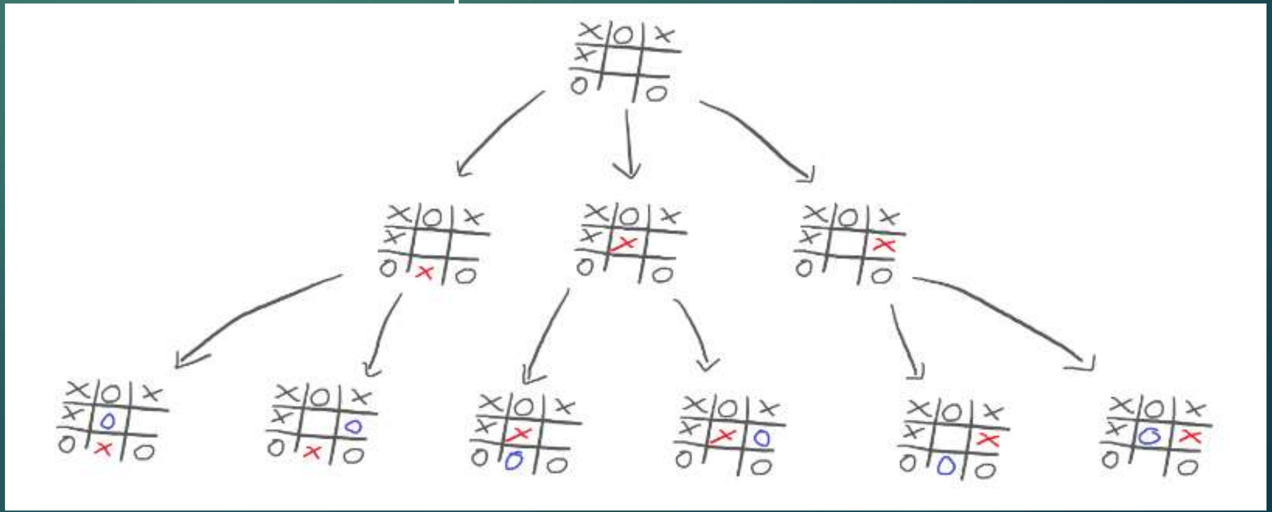


APPROACH

- We are going to create a CLI tic-tac-toe game using programming language **PYTHON(version 3.7)**
- Artificial Intelligence calls python for resolving challenging algorithms as it offers the advantage of a logical code and supports interpretive run-time

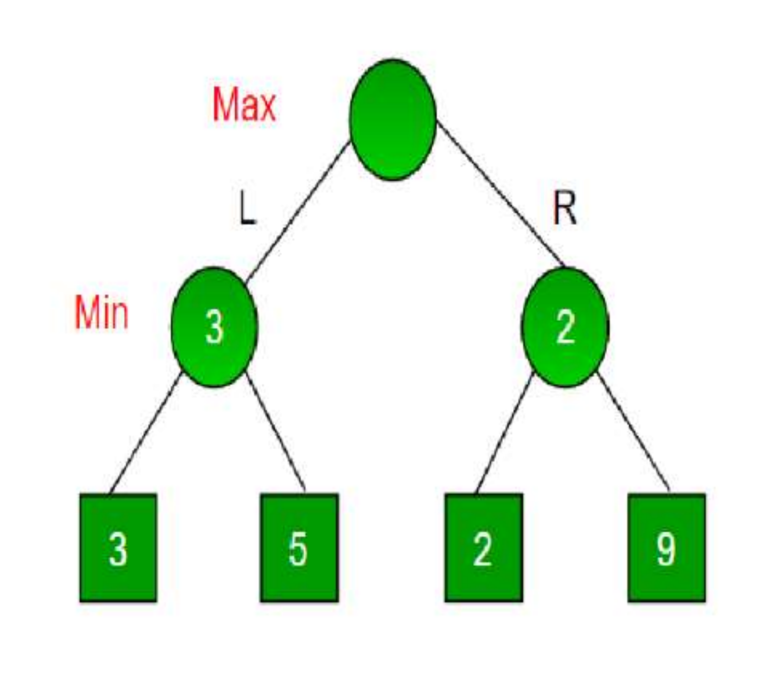
ALGORITHMS USED IN THE PROJECT

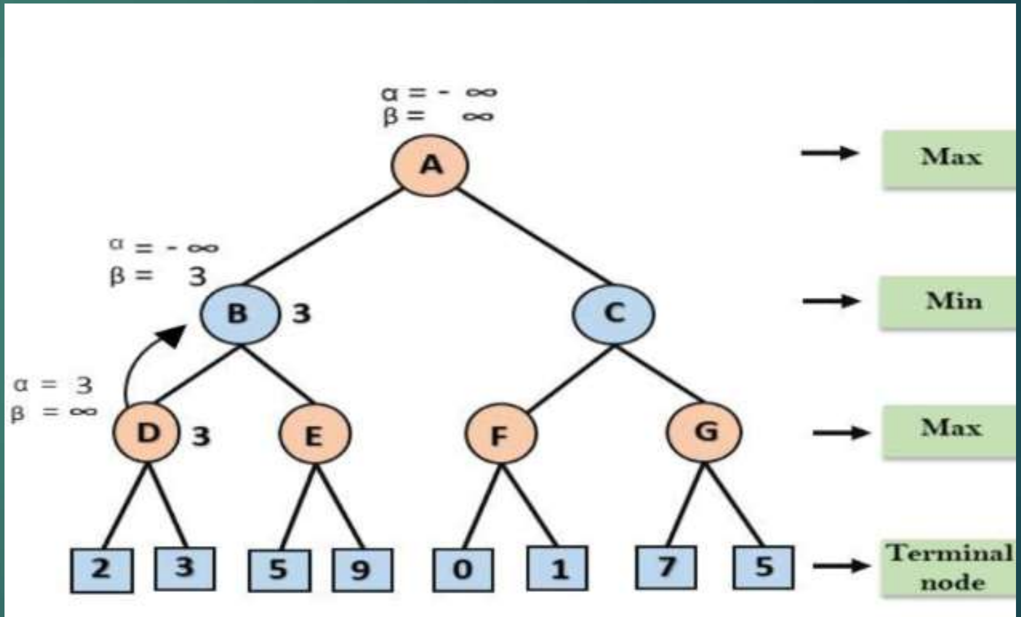
- 1. **MINIMAX**: This algorithm is used for minimizing the possible loss in a worst-case scenario.
- 2. **ALPHA BETA PRUNING MINIMAX:** It is an algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm
- 3. **EXPECTIMAX:** It is a game theory algorithm used to maximize the expected utility
- 4. **QLEARNING – REINFORCEMENT LEARNING:** A model-level algorithm to learn the value of an action in a particular state



MINIMAX GRAPH

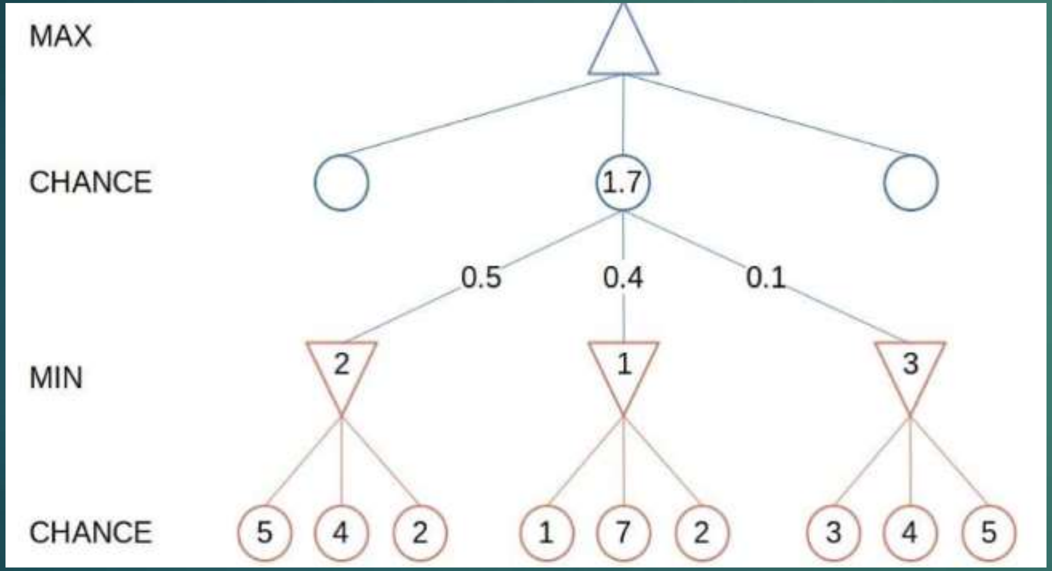
ALPHA-BETA GRAPH

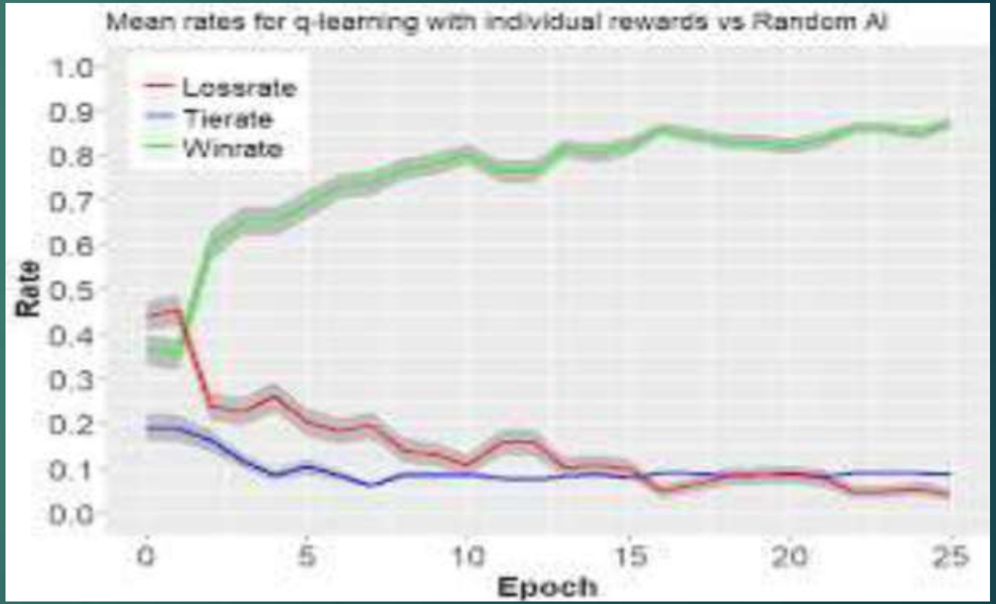




EXPECTIMAX GRAPH

Q-LEARNING GRAPH

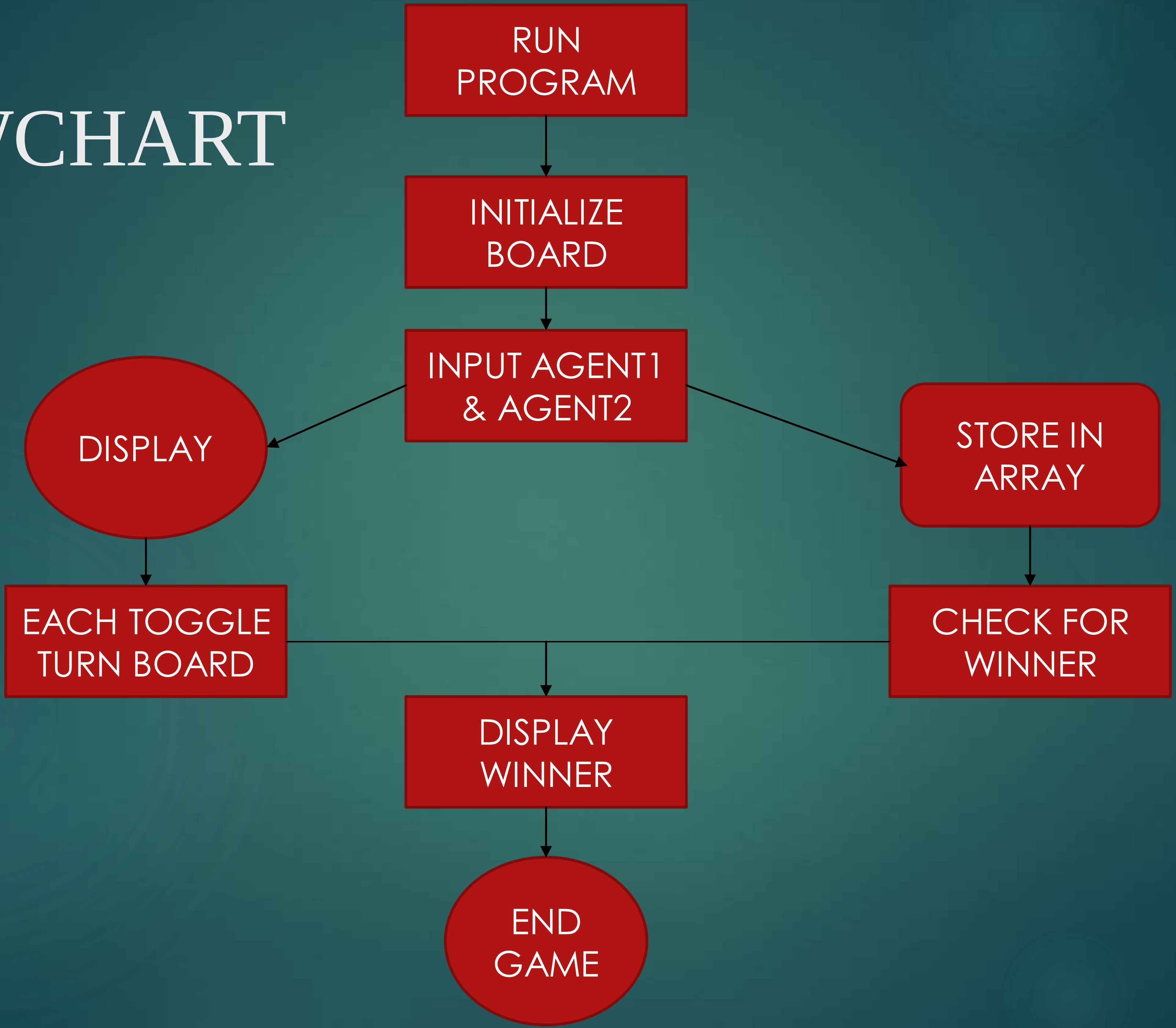




EVALUATION METHODOLOGY

- The efficient implementation of four AI algorithms will determine the project's success.
- The code for this project will implement all 4 algorithms mentioned in the previous slide.
- The two agents we use in the project should be trained to play games with the same level of accuracy as two human brains would.
- We will also look at how the CLI asks for the selection of 2 different agents with different algorithms and see how the agents play against each other.

FLOWCHART


RUN PROGRAM
INITIALIZE BOARD
INPUT AGENT1 & AGENT2
DISPLAY
STORE IN ARRAY
EACH TOGGLE TURN BOARD
CHECK FOR WINNER
DISPLAY WINNER
END GAME

IMPLEMENTATION

- We will open the command prompt in the respective folder.
- Command to run:

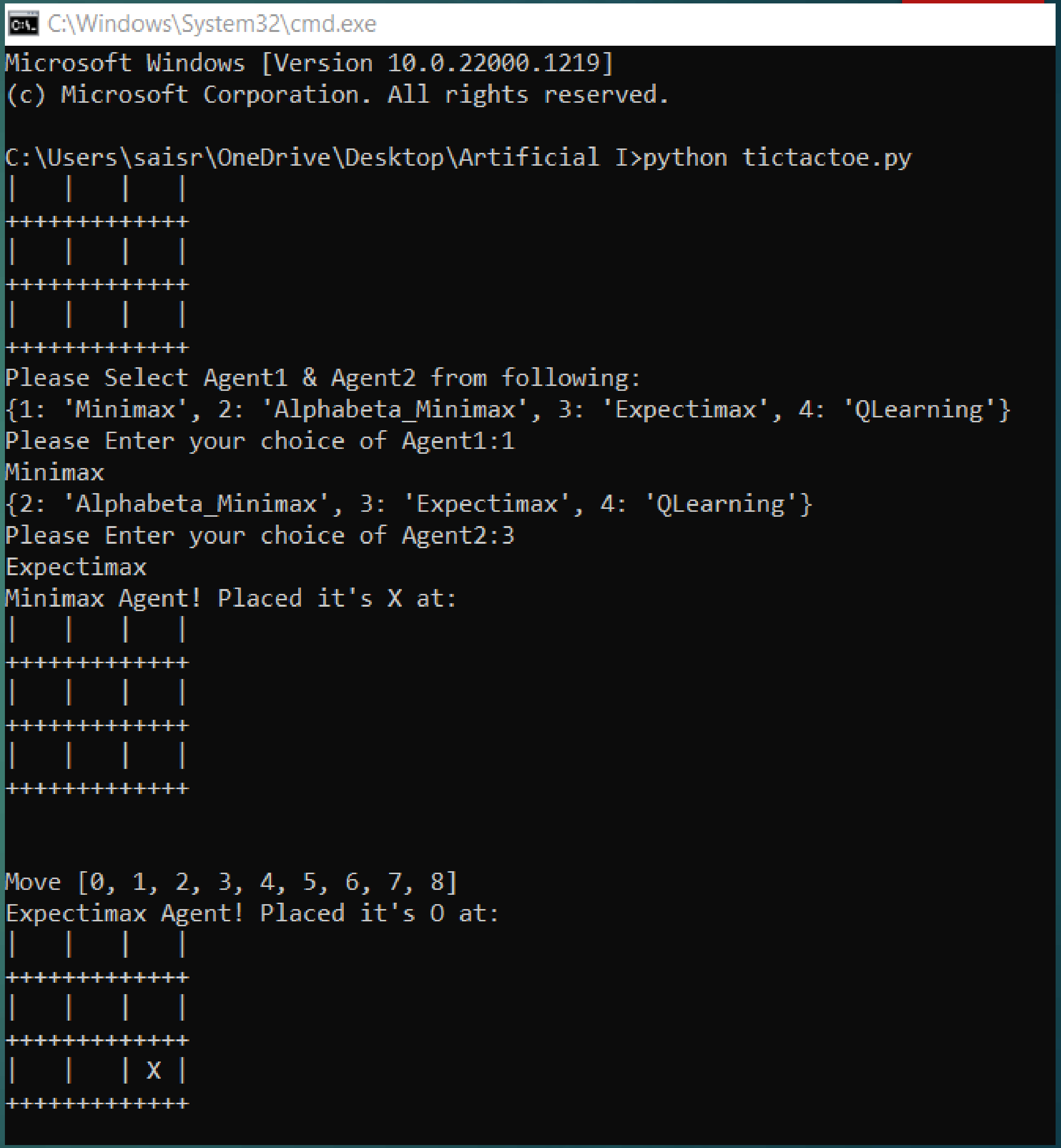
python tictactoe.py

This will execute the python code and will start the program to run

Step1: Select one algorithm for Agent1

Step 2: Select another algorithm for Agent2 as shown in the image.

Step 3: click enter for the program to run.



```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\Artificial I>python tictactoe.py
|   |   |   |
+++++++++++++
|   |   |   |
+++++++++++++
|   |   |   |
+++++++++++++
Please Select Agent1 & Agent2 from following:
{1: 'Minimax', 2: 'Alphabeta_Minimax', 3: 'Expectimax', 4: 'QLearning'}
Please Enter your choice of Agent1:1
Minimax
{2: 'Alphabeta_Minimax', 3: 'Expectimax', 4: 'QLearning'}
Please Enter your choice of Agent2:3
Expectimax
Minimax Agent! Placed it's X at:
|   |   |   |
+++++++++++++
|   |   |   |
+++++++++++++
|   |   |   |
+++++++++++++


Move [0, 1, 2, 3, 4, 5, 6, 7, 8]
Expectimax Agent! Placed it's O at:
|   |   |   |
+++++++++++++
|   |   |   |
+++++++++++++
|   |   | X |
+++++++++++++
```

THANK YOU !!