Aakriti Singh 20190802006

Fundamentals of AI & ML

Monsoon Semester V 2021-22

Lab - 3

Date: 30 September 2021

Topic: Intelligent Agent

AIM

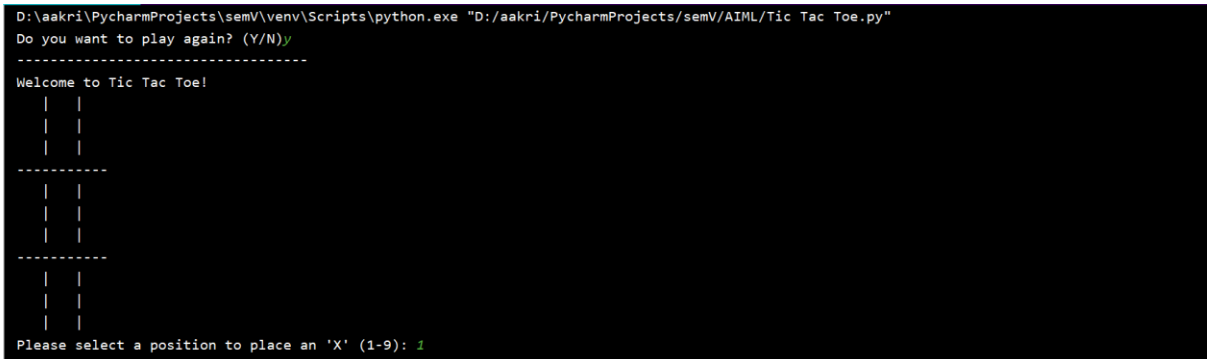
Write a program to implement the Tic-Tac-Toe game using python.

PROGRAM CODE

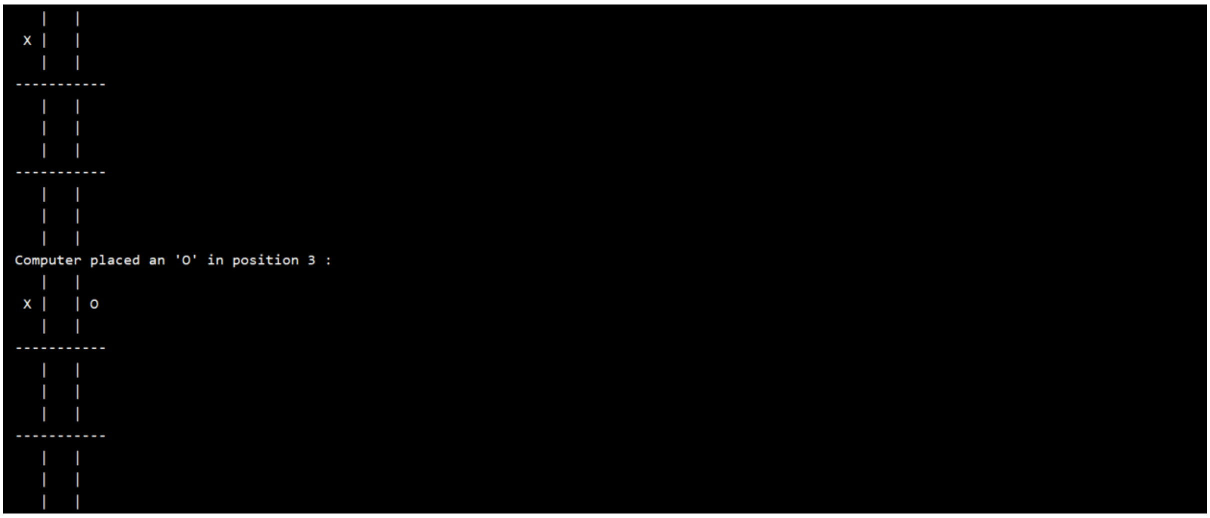
```
import random


def place(num, x):
    # Returns the value in x at position num
    for i, v in enumerate(x):
        if v == num:
            return x[(i + 1)]
    return str(num)


def print_grid(move, player, x):
    pos_list.extend([move, player])

    # Grid on which the player and computer play on
    template = """
    -------------------
    |     |     |     |
    |  {}  |  {}  |  {}  |
    |     |     |     |
    |-----------------|
    |     |     |     |
    |  {}  |  {}  |  {}  |
    |     |     |     |
    |-----------------|
    |     |     |     |
    |  {}  |  {}  |  {}  |
    |     |     |     |
    -------------------"""

    if x == 2:
        # Only prints if the player has made a move
        print(template.format(*(place(num + 1, pos_list) for num in
range(9))))
```

```
def winner(x, player, xx):
    # Checks if there is a winner
    wins = ((1, 2, 3), (4, 5, 6), (7, 8, 9),  # Horizontal
            (1, 4, 7), (2, 5, 8), (3, 6, 9),  # Vertical
            (1, 5, 9), (3, 5, 7))  # Diagonal

    if any(all(pos in x for pos in win) for win in wins):
        if xx != 1:
            print('\n' * 5, "'{}'".format(player), "HAS WON!")
        return True
    return False


def computer_AI_part(moves):
    global computer_move

    # Checks all possible values which the player can and enter to win and 
blocks it
    for x in range(1, 10):
        if x not in pos_list:
            moves.append(x)
            if winner(moves, 'Computer', 1):
                del moves[-1]
                computer_move = x
                return 1
            del moves[-1]


if __name__ == "__main__":
    global computer_move, pos_list, player_list, computer_list
    replay, draw = 0, 0

    while True:

        # Replay's the game
        if replay:
            restart = input("Would you like to replay?: ").lower()
            if restart in ("y", "yes"):
                pass
            elif restart in ("n", "no"):
                exit()
            else:
                print("Say 'yes' or 'no'")
                continue
        else:
            print("\nTic Tac Toe - Computer vs You", '\n' * 2, "Computer goes 
first\n")

        replay, computer_move, players_move, loop_count = 0, 0, 0, 0
        pos_list, player_list, computer_list = [], [], []

        for each in "XXXXX":
            loop_count += 1
            # Computer's Move
            if computer_AI_part(computer_list) or
computer_AI_part(player_list) == 1:
                pass
            else:
                while True:
                    computer_move = random.randint(1, 9)
                    if computer_move not in pos_list:
                        break
            computer_list.append(computer_move)
            # Prints Grid
            print_grid(computer_move, 'O', 2)

            if loop_count == 5:
                if winner(player_list, 'player', 2) or winner(computer_list,
'Computer', 2):
                    pass
                else:
                    print("Match Was a draw!")
                replay = 1
                break

            # Checks winner
            if winner(computer_list, 'Computer', 2):
                replay = 1
                break

            # Player's Move
            while True:
                try:
                    players_move = int(input("\n\'%s\' Enter a value from the
grid to plot your move: " % each))
                    if players_move in pos_list or players_move < 1 or
players_move > 9:
                        print("Enter an available number that's between 1-9")
                        continue
                    break
                except ValueError:
                    print("Enter a number")
                except (EOFError, KeyboardInterrupt):
                    exit()

            player_list.append(players_move)
            # Sets player's move for printing
            print_grid(players_move, each, 1)

            # Checks winner again
            if winner(player_list, 'player', 1):
                print_grid(players_move, each, 2)
                winner(player_list, 'player', 2)
                replay = 1
                break
```

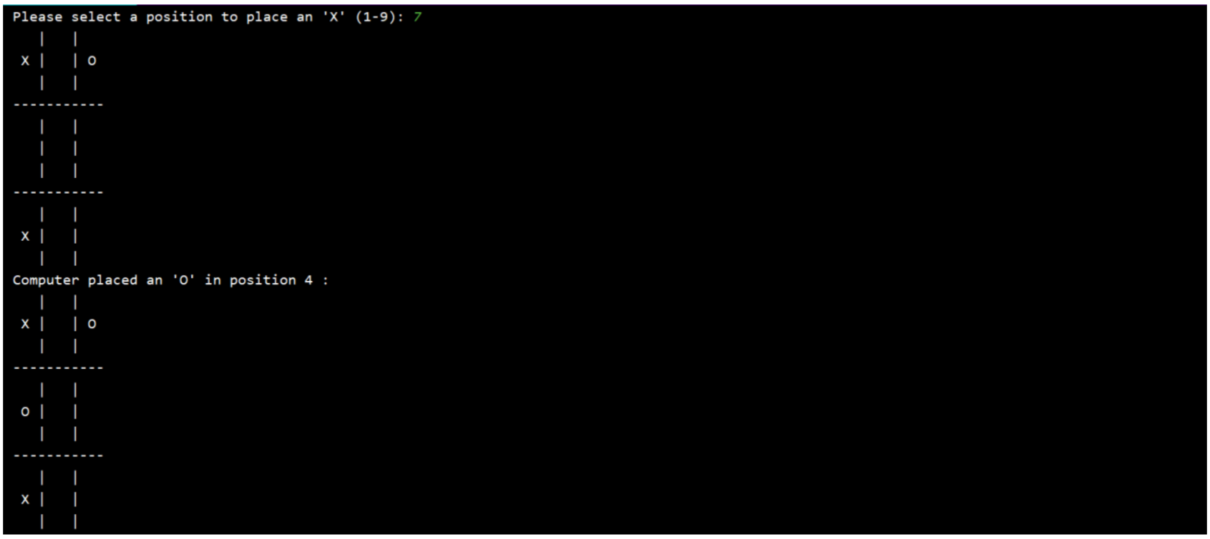
OUTPUT



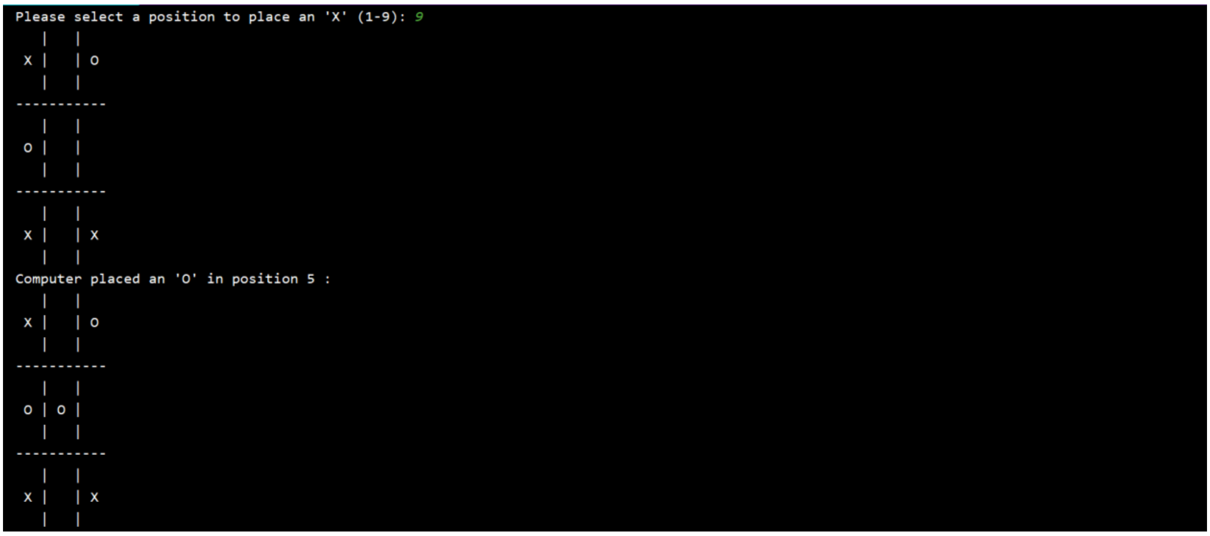
```
D:\aakri\PycharmProjects\semV\venv\Scripts\python.exe "D:/aakri/PycharmProjects/semV/AIML/Tic Tac Toe.py"
Do you want to play again? (Y/N)y
----------------------------------
Welcome to Tic Tac Toe!
   |   |
   |   |
   |   |
-----------
   |   |
   |   |
   |   |
-----------
   |   |
   |   |
   |   |
Please select a position to place an 'X' (1-9): 1
```



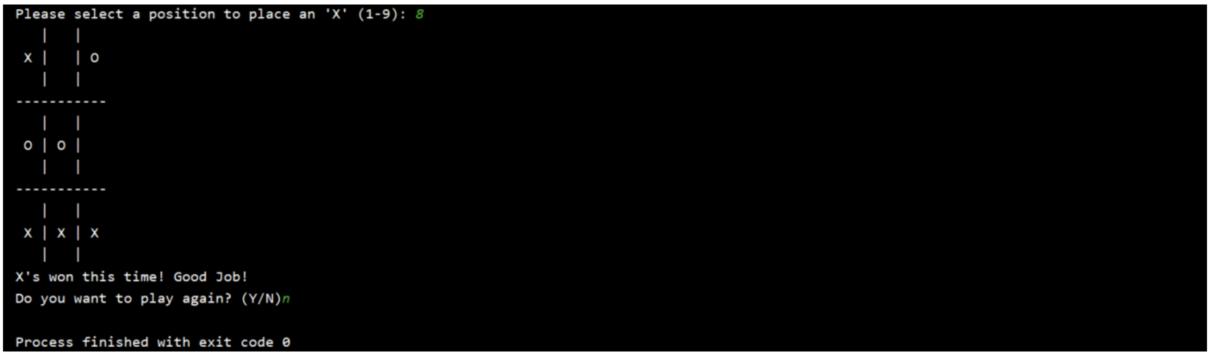
```
   |   |
 X |   |
   |   |
-----------
   |   |
   |   |
   |   |
-----------
   |   |
   |   |
   |   |
Computer placed an 'O' in position 3 :
   |   |
 X |   | O
   |   |
-----------
   |   |
   |   |
   |   |
-----------
   |   |
   |   |
   |   |
```



```
Please select a position to place an 'X' (1-9): 7
   |   |
 X |   | O
   |   |
-----------
   |   |
   |   |
   |   |
-----------
   |   |
 X |   |
   |   |
Computer placed an 'O' in position 4 :
   |   |
 X |   | O
   |   |
-----------
   |   |
 O |   |
   |   |
-----------
   |   |
 X |   |
   |   |
```



```
Please select a position to place an 'X' (1-9): 9
   |   |
 X |   | O
   |   |
----------
   |   |
 O |   |
   |   |
----------
   |   |
 X |   | X
   |   |
Computer placed an 'O' in position 5 :
   |   |
 X |   | O
   |   |
----------
   |   |
 O | O |
   |   |
----------
   |   |
 X |   | X
   |   |
```



```
Please select a position to place an 'X' (1-9): 8
   |   |
 X |   | O
   |   |
----------
   |   |
 O | O |
   |   |
----------
   |   |
 X | X | X
   |   |
X's won this time! Good Job!
Do you want to play again? (Y/N)n

Process finished with exit code 0
```

CONCLUSION

Tic Tac Toe with Intelligent Agent was implemented successfully using Python.