24/09/24

Tic Tac Toe

i) write an algorithm

Step 1 > Create a 2D array

Step 2 :- 2D array = 3x3 and initializing empty spaces '–'

=> Create 3 rows a 3 different lists [[ ], [ ], [ ]]

```
[[–, –, –]
 [–, –, –]
 [–, –, –]]
```

Step 3 :- Use of random function to choose first move.
The first move will always have X as start.

Display the board :

```
def print_board(board):
    for row in board :
    print ('|'.join(row))
    print ('–') for empty spaces.
```

Step 4 :- Check for a winner

=> 8 possibilities (2 rows, 3 col, 3 diagonals)

```
def check_winner(board):
for i in range(3):
    if board[i][0] == board[i][i] == board[i][2] != " "
        return board[i][0]
```

Similarly check for col & diag

Step 5:- Player move :- The player should enter row & col index to make a move

```
def player - move (board) :

    if board [row][col] = '_';

      board [row] [col] = "x"

      break

  else
      print (" Try again.");
```

Step 6:- Computer move :-

i) Check for winning move

```
 for i in range (3);
 for j in range (3);

      if board(i) [j] = ='_';
         board [i][j] : = 'O';

      if  check. winner (board) = ='O';
          return
```

→ If no winning move, pick a random move such that player cant win.

```
import random

def print_board (board):
        for row in board :
        print ("| ". join (row))
        print ("_"* 9)

def     check_winner (board) :
      for i in range (3) :
            if board [i] [0] = =
              return board

            if board [o] [i] ==
              return board

      if board [0] [0] ==board
           return board [

      if board [0] [2] == ba
             return boa

      return None

def is_full (board) :
    return all ( cell ( ='

def find_winning_
      for i in range(3
        for j in ra
           if board
           board
           if (
             bo
             r
```

```
import random

def print_board (board):
        for row in board:
        print ("| ". join (row))
        print ("_"* 9)

def check_winner (board):
   for i in range (3):
        if board [i] [0] = = board (i) (1) = = board (i) [2] ! = " ":
            return board (i) [0]

        if board [0] (i) == board [1] (i) = = board [2] (i) ! = " ";
            return board [0] (i)

   if board [0] [0] = = board [1] [1] = board [2] [2] ! = " ":
        return board [0] (0)

   if board [0] [2] = = board [1] [1] == board [2] [0]. ! = " ":
        return board (0) [2]

   return None

def is_full (board):
  return all (cell ! = " " for row in board for cell in row)

def find_winning_move (board, player):
   for i in range (3):
        for j in range (3):
            if board [i] [j] = = " ";
              board [i] [j] = = player
              if check_winner (board) = = player;
                board [i] [j] = " "
                return (i, j)
        board [i] (j) =" "

  return None

def get - computer_ move (board) :

        move = find- winning - move (board,"O")
          if move:
              return move

        if board [1][1] == " ":

          return (1, 1)

        Corners =[(0,0), (0,2), (2,0), (2,2)]
        for corner in corners ;

              if  board [corner [0] ][corner [1]] = =" ":

           for i in range (3):
           for j in range (3):
              if board [i][j] ==" ":
                return (i, j)

def tic- tac-toe ():

        board = [[" " for _ in range (3)] for _ in range (3)]
        current _ player = "X" .
        computer- player = "O"
        print ("player X goes first.")
```

```
while True :
    print_board (board)
  if  current - player = = "X" :
      while True :
      try :
          row = int (input ( "Player X, enter the row (0-2): "))
        col = int (input ("player X, enter the col (0-2):"))
     if  board [row] [col] = = " " :
       break

   else:
     print (" cell is already taken! Try again")

 except (ValueError, IndexError) :
   print ("Invalid input ! Please enter number b/w 0 & 2.")

else :
  print ( "Computer's turn .. ")
   row, col = get-computer_move (board)
  print (f"Computer chooses row {row}, col {col}")

 board [row] [col] = current - player

 winner  = check - winner (board)
  if winner :
       print - board (board)
     print (f"player {winner} wins !")
       break
 if is full (board) :
    print - board (board)
    print (" It's a tie !")
    break
```

```
current_player = comput
if _name_ == "_main_":
    tic_tac_toe()
```

Output:-

```
Player X goes first
```


```
Player X, enter the row (0-2): 2
Player X, enter the col (0-2): 1
```

	X	

```
Computer's turn:
        row 1, col = 1
Player     row = 1  col = 2
```

	O	X
	X	

```
Computer turn   row = 0, col = 0
```

24/9/24

O		
	O	X
	X	

```
Player x   row 2  col = 0
```

Computer row = 2 col=2

O		
	O	X
X	X	O

```
Player O wins
```