classmate
Date 1/10/24
Page 01

LAB 1

Implement a tic-tac-toe game

* Program title:
Implement a tic-tac-toe game.

* Algorithm:
function print_board (board)
print board format

function check_winner (board)
for each row in board
if all elements in row are same and not em
return the element
for each column
if all elements in col are same and not em
return the element
if diagonal checks yields same element and
not empty:
return the element.
return none

function is_board_full (board):
return True if no empty spaces, else false.

function main():
initialise board with empty spaces
iteration = 0
winner = none
while winner is none:
if iteration is even:
print_board (board)
get user input for row & column.

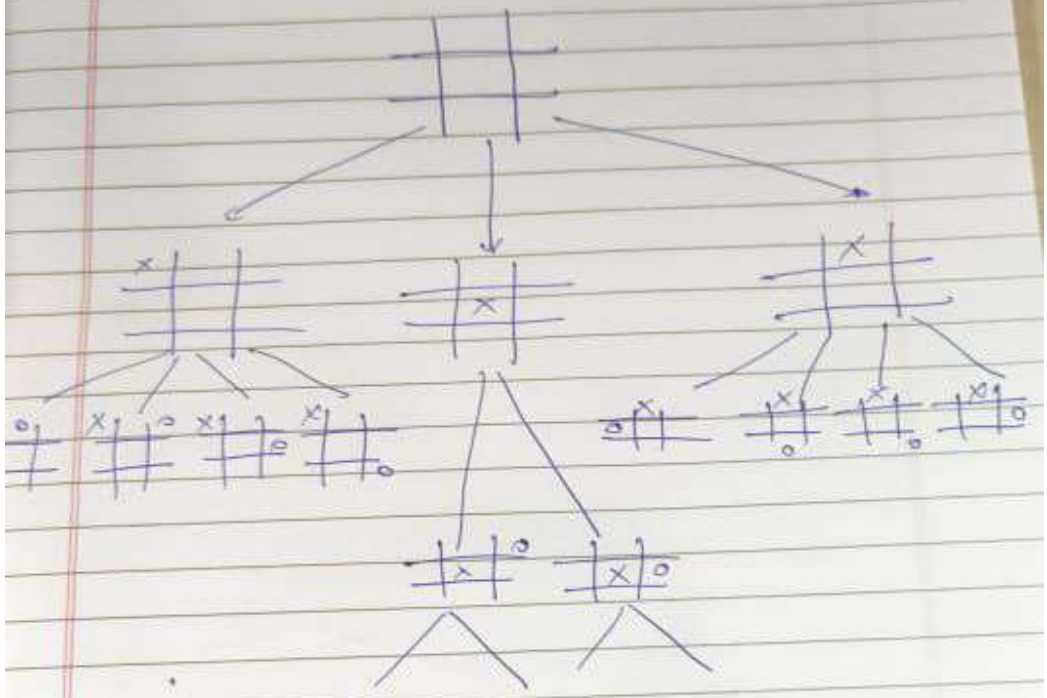
```
        validate input
        place 'O' in board
    else:
        randomly select empty position for 'X'.
iteration t=1
winner = check_winner(board)

if winner is not None:
    print winner message
    break

print game introduction
main()
```

State space

① Tic-Tac-Toe



classmate
Date 1/10/24
Page

Project title :
Implement vaccuum cleaner agent.

Algorithm :

```
function vaccuum cleaner Agent (environment)
  position = (0,0)
  cleaned_cells_count = 0

  while True:
    if environment [position] is dirty:
      clean (environment [position])
      cleaned_cells_count += 1
      print ("cleaned position:", position)

    next_position = find NextDirty (environment)
    if next_position exists:
        position = next_position
    else:
        print ("No more dirty cells found")
        break.

function findNextDirty (environment):
  for each cell in environment:
    if cell is dirty:
      return cell's position.
  return None.
```

② Vacuum cleaner