EXPERIMENT NO: 1 **A N V SREEVISHNU**
DATE: 29/01/2021 **RA1811003010333**

TOY PROBLEM IN AI

Aim: To implement Toy Problem (Tic Tac Toe) in AI using Python where no input is taken. The game is played automatically between AIs and the winner is declared after the game.

Procedure/Algorithm:

- A 9×9 board is created and all the places a re initialized with 0s.
- Two players (1 and 2) are considered to be playing, whose moves are decided by a random number generator function (random_place()).
- The random place() function is called for a player 1 and is marked on the board with a ' 1' corresponding to the player.
- The new state of the board is printed after each individual move with the player's number (1 or 2).
- The 2nd player chooses a position on the board which is still empty using the random place() function.
- The marking of position is done and the state of t he board is printed.
- It is checked whether a row,a column or a diagonal has the same player number on the board after each move. If so, the winner's n ame is displayed.
- Display -1 if there are no winners after 9 moves.

Code:

```
import numpy as np
import random from
time import sleep
  def create_board():
return(np.array([[0, 0, 0],
                 [0, 0, 0 ],
                 [0, 0, 0 ]]))
  def possibilities(board):
l = []          for i in
range(len(board)):          for j
in range(len(board)):
                    if
board[i][j] ==          0:
```

```
            l.append((i, j))
return(l)

def random_place(board, player):
selection = possibilities(board)
current_loc = random.choice(selection)
board[current_loc] = player
   return(board)

def row_win(board, player):
      for x in range(len(board)):
       win = True
               for y in
range(len(board)):            if
board[x, y] != player:
            win = False
            continue
                     if
win == True:
return(win)
   return(win)
   def col_win(board,
player):      for x in
range(len(board)):
       win = True
               for y in
range(len(board)):            if
board[y][x] != player:
            win = False
            continue
                     if
win == True:
return(win)
   return(win)
   def diag_win(board,
player):
   win = True     y = 0
for x in range(len(board)):
if board[x, x] != player:
         win = F alse     if win:
return win      win = True
if win:         for x in
range(len(board)):            y
```

```
= len(board) -  1 - x
if board[x, y] != player:
            win = False
return win
   def
evaluate(board):
   winner = 0
         for player in
[1, 2]:
       if (row_win(board, player) or
col_win(board,player) or
diag_win(board,player)):

         winner = player
               if np.all(board != 0) and
  winner == 0:
      winner = -1
return winner

def play_game():
   board, winner, counter = create_board(), 0, 1
   print(board)
   sleep(2)
         while winner
== 0:         for player
in [1, 2]:
         board = random_place(board, player)
print("Board after " + str(counter) +  " move")
print(board)             sleep(2)             counter += 1
winner = evaluate(board)             if winner != 0:
            break
   return(winner)

print("Winner is: " + str(play_game()))
```

Output:

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
Board after 1 move
[[0 0 0]
 [0 0 0]
 [0 0 1]]
Board after 2 move
[[0 0 0]
 [0 0 0]
 [0 2 1]]
Board after 3 move
[[0 0 0]
 [0 0 1]
 [0 2 1]]
Board after 4 move
[[0 0 0]
 [0 0 1]
 [2 2 1]]
Board after 5 move
[[0 0 0]
 [0 1 1]
 [2 2 1]]
Board after 6 move
[[0 0 0]
 [2 1 1]
 [2 2 1]]
Board after 7 move
[[0 1 0]
 [2 1 1]
 [2 2 1]]
Board after 8 move
[[0 1 2]
 [2 1 1]
 [2 2 1]]
Board after 9 move
[[1 1 2]
 [2 1 1]
 [2 2 1]]
Winner is: 1
```

Result: Thus, the implementation of Toy Problem (Tic Tac Toe) in AI using Python has been successfully done.