Program 1: Implement Tic –Tac –Toe Game..

PRANAV JAGADEESH 1BM18CS071

Program 1 Implement Tic – Tac – Toe Game

Theory :-

This AI program intelligently responds to player's moves. In this game two players X and O take turns by playing X or O. If a player gets three of their marks on the board in a row, column or one of the two diagnols, they win. When the board fills up with neither player winning the game ends in a draw

PRANAV JAGADEESH 1BM18CS071 5B

TIC TAC TOE PROGRAM B2 BATCH

```
import random
def drawBoard(board):
    print(' | |')
    print(' ' + board[7] + '/' + board[8] + '/' + board[9])
    print(' | |')
    print('-----------')
    print(' | |')
    print(' ' + board[4] + '/' + board[5] + '/' + board[6])
    print(' | |')
    print('-----------')
    print(' | |')
    print(' ' + board[1] + '/' + board[2] + '/' + board[3])
    print(' | |')
def InputPlayerLetter():
    letter = ''
    while not (letter == 'X' or letter == 'O'):
        print('Do you want to be X or O?')
        letter = input().upper()
    if letter == 'X':
        return ['X', 'O']
    else: return ['O', 'X']
def whoGoesFirst():
    if random.randint(0, 1) == 0:
        return 'Computer'
    else: return 'player'
def playAgain():
    print('Do you want to play again? (yes or no)')
    return input().lower().startswith('y')
def makeMove(board, letter, move):
    board[move] = letter
def isWinner(bo, le):
    return ((bo[7] == le and bo[8] == le and bo[9] == le
```

```
or (bo[4] == le and bo[5] == le and bo[6] == le)
or (bo[1] == le and bo[2] == le and bo[3] == le)
or (bo[7] == le and bo[4] == le and bo[1] == le)
or (bo[8] == le and bo[5] == le and bo[2] == le)
or (bo[9] == le and bo[6] == le and bo[3] == le)
or (bo[7] == le and bo[5] == le and bo[3] == le)
or (bo[9] == le and bo[5] == le and bo[1] == le)
def getBoardCopy(board):
    dupeBoard = []
    for i in board:
        dupeBoard.append(i)
    return dupeBoard
def isSpaceFree(board, move):
    return board[move == ' '
def getPlayerMove(board):
    move =
    while move not in '1 2 3 4 5 6 7 8 9'.split() or not
    isSpaceFree(board, int(move)):
        print('What is your next move? (1-9)')
        move = input()
    return int(move)
def chooseRandomMoveFromList(board, movelist):
    possibleMoves = []
    for i in movelist:
        if isSpaceFree(board, i):
            possibleMoves.append(i)
    if len(possibleMoves) != 0:
        return random.choice(possibleMoves)
    else: return None
def getComputerMove(board, computerLetter):
if computerLetter == 'X': playerLetter = 'O'
else: playerLetter = 'X'
```

```
for i in range (1, 10):
    copy = getBoardCopy (board)
    if isSpaceFree (copy, i):
        makeMove (copy, computerLetter, i)
        if isWinner (copy, computerLetter):
            return i

for i in range (1, 10):
    copy = getBoard Copy (board)
    if isSpace Free (copy, i):
        make Move (copy, playerLetter, i)
        if isWinner (copy, playerLetter):
            return i

move = chooseRandomMoveFromList (board, [1, 3, 7, 9])
if move != None:
    return move
if isSpace Free (board, 5):
    return 5
return chooseRandomMoveFromList (board, [2, 4, 6, 8])
def isBoardFull (board):
    for i in range (1, 10):
        if isSpace Free (board, i):
            return false
        return True
    print ('Welcome to Tic Tac Toe !')
while True:
    theBoard = [' '] * 10
    playerLetter, computerLetter = inputPlayerLetter ()
        turn = whoGoesFirst ()
        print ('The ' + turn + ' will go first.')
    gameIsPlaying = True
```

```
while gameIsPlaying:
    if turn == 'player':
        drawBoard(theBoard)
        move = getPlayerMove(theBoard)
        makeMove(theBoard, playerLetter, move)
        if isWinner(theBoard, playerLetter):
            drawBoard(theBoard)
            print('Hooray! You have won the game!')
            gameIsPlaying = False
        else:
            if isBoardFull(theBoard):
                drawBoard(theBoard)
                print('The game is a tie!')
                break
            else:
                turn = 'computer'
    else:
        move = getComputerMove(theBoard,
                               computerLetter)
        makeMove(theBoard, computerLetter, move)
        if isWinner(theBoard, computerLetter):
            drawBoard(theBoard)
            print('The computer has beaten you! You lose.')
            gameIsPlaying = False
        else:
            if BoardFull(theBoard):
                drawBoard(theBoard)
                print('The game is a tie!')
                break
            else:
                turn = 'player'
    if not playAgain():
        break
```

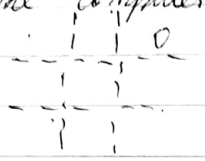
Output :-

Welcome to Tic Tac Toe

Do you want to be X or O?

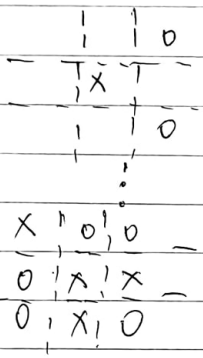
X

The computer will go first.



What is your next move? (1-9)

5



The game is a tie

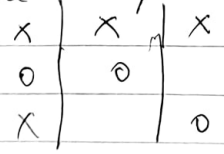
Do you want to play again? (yes or no)

y

Do you want to be X or O?

O

The computer will go first



The computer has beaten you! You lose.

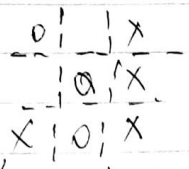
Do you want to play again? (yes or no)

y

Do you want to be X or O?

X

The player will go first.



Hooray! You won the game!
Do you want to play again? (yes or no)
n

```
# Tic Tac Toe
import random

def drawBoard(board):
    # prints  the board .

    print('   |   |')
    print(' ' + board[7] + ' | ' + board[8] + ' | ' + board[9])
    print('   |   |')
    print('-----------')
    print('   |   |')
    print(' ' + board[4] + ' | ' + board[5] + ' | ' + board[6])
    print('   |   |')
    print('-----------')
    print('   |   |')
    print(' ' + board[1] + ' | ' + board[2] + ' | ' + board[3])
    print('   |   |')

def inputPlayerLetter():
    # player type which letter they want.

    letter = ''
    while not (letter == 'X' or letter == 'O'):
        print('Do you want to be X or O?')
        letter = input().upper()

    # the first element  is the player's letter, the second is computer's letter.
    if letter == 'X':
        return ['X', 'O']
```

```
    else:
        return ['O', 'X']


def whoGoesFirst():
    # Randomly choose the player who goes first.
    if random.randint(0, 1) == 0:
        return 'computer'
    else:
        return 'player'


def playAgain():
    # returns True if player wants to play again,else returns False.
    print('Do you want to play again? (yes or no)')
    return input().lower().startswith('y')


def makeMove(board, letter, move):
    board[move] = letter


def isWinner(bo, le):
    #True if that player has won.
    # bo instead of board and le instead of letter.
    return ((bo[7] == le and bo[8] == le and bo[9] == le) or #top
    (bo[4] == le and bo[5] == le and bo[6] == le) or #middle
    (bo[1] == le and bo[2] == le and bo[3] == le) or #bottom
    (bo[7] == le and bo[4] == le and bo[1] == le) or #left side
    (bo[8] == le and bo[5] == le and bo[2] == le) or #middle
    (bo[9] == le and bo[6] == le and bo[3] == le) or #right side
    (bo[7] == le and bo[5] == le and bo[3] == le) or #diagonal
    (bo[9] == le and bo[5] == le and bo[1] == le)) #diagonal
```

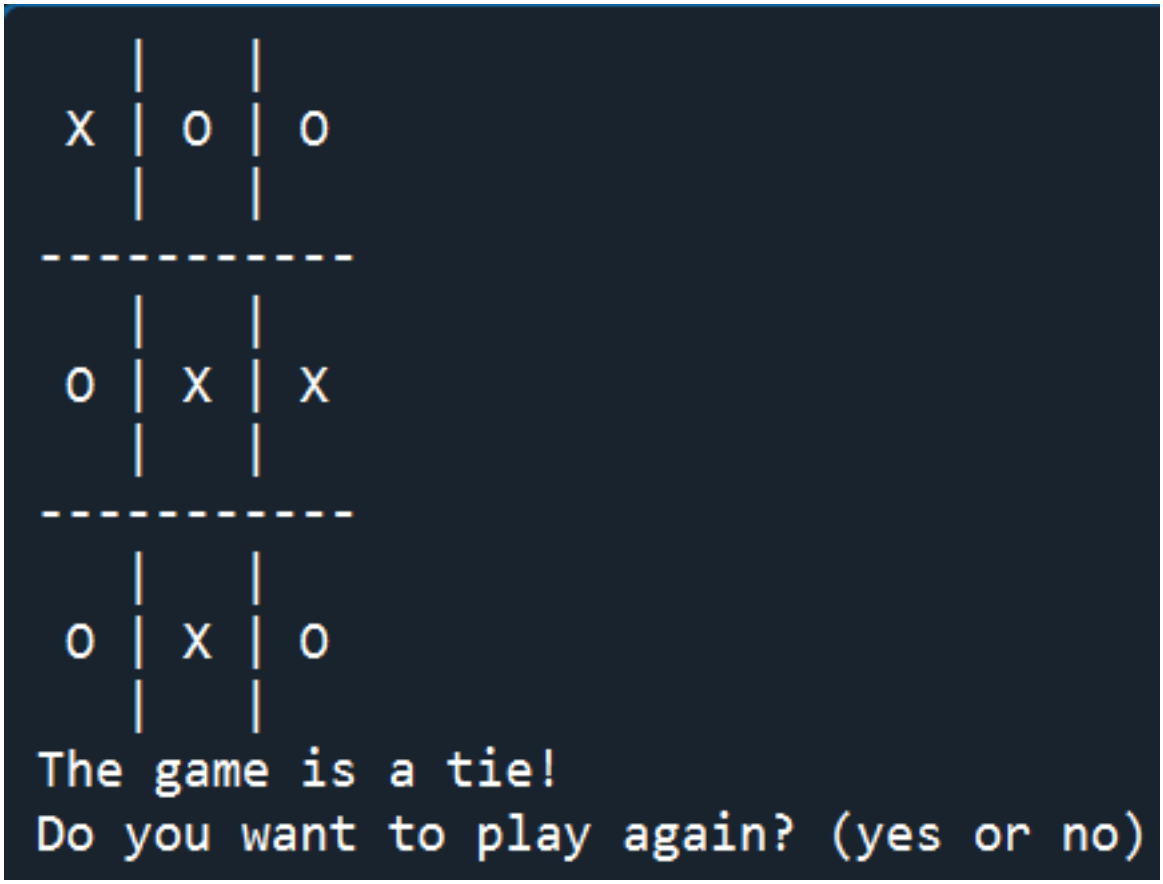
```
def getBoardCopy(board):
    #duplicate of board list and return duplicate.
    dupeBoard = []

    for i in board:
        dupeBoard.append(i)

    return dupeBoard

def isSpaceFree(board, move):
    # Return true if passed move is free on board.
    return board[move] == ' '

def getPlayerMove(board):
    # Let player type his move.
    move = ' '
    while move not in '1 2 3 4 5 6 7 8 9'.split() or not isSpaceFree(board, int(move)):
        print('What is your next move? (1-9)')
        move = input()
    return int(move)

def chooseRandomMoveFromList(board, movesList):
    # Returns a valid move.
    # Returns None if no valid move.
    possibleMoves = []
    for i in movesList:
        if isSpaceFree(board, i):
            possibleMoves.append(i)

    if len(possibleMoves) != 0:
```

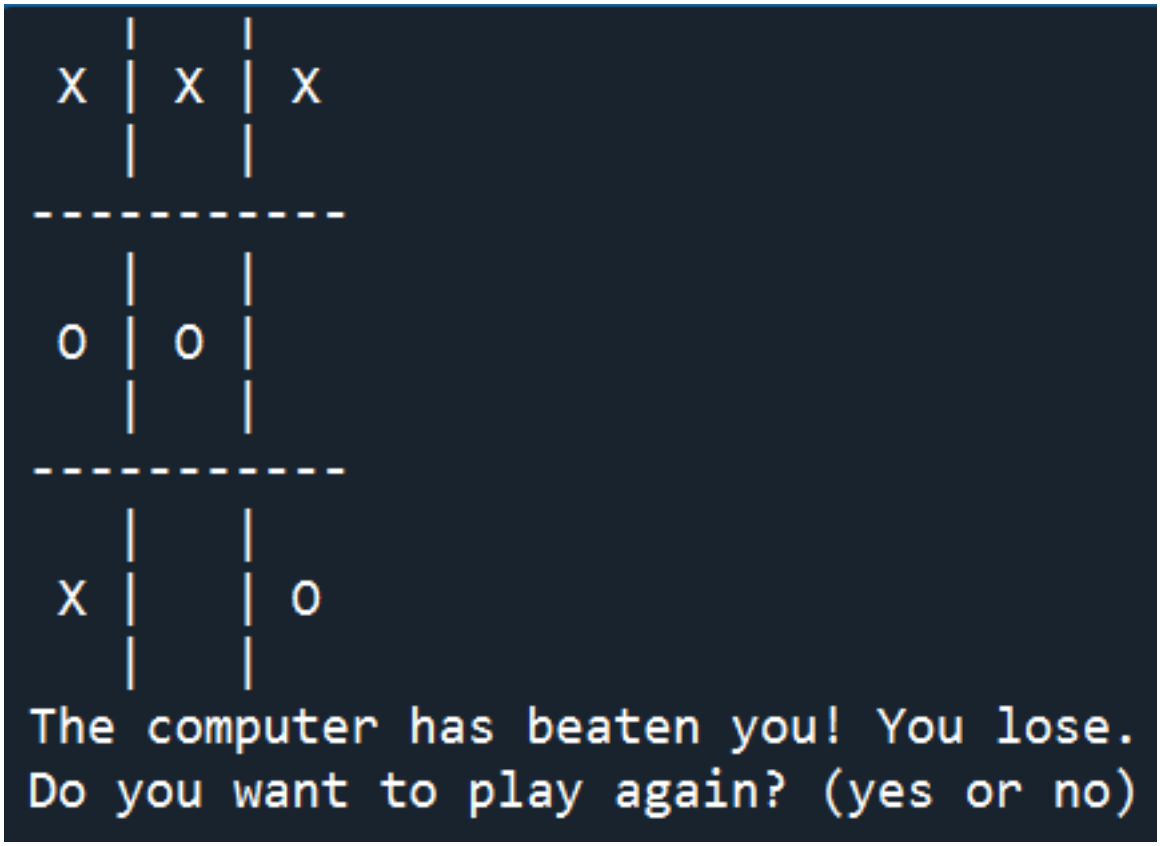
```
        return random.choice(possibleMoves)
    else:
        return None

def getComputerMove(board, computerLetter):
    #determine where to move and return that move.
    if computerLetter == 'X':
        playerLetter = 'O'
    else:
        playerLetter = 'X'
    #check if we can win in next move
    for i in range(1, 10):
        copy = getBoardCopy(board)
        if isSpaceFree(copy, i):
            makeMove(copy, computerLetter, i)
            if isWinner(copy, computerLetter):
                return i

    # Check if the player could winand block them.
    for i in range(1, 10):
        copy = getBoardCopy(board)
        if isSpaceFree(copy, i):
            makeMove(copy, playerLetter, i)
            if isWinner(copy, playerLetter):
                return i

    #take one of the corners, if they are free.
    move = chooseRandomMoveFromList(board, [1, 3, 7, 9])
    if move != None:
        return move
```

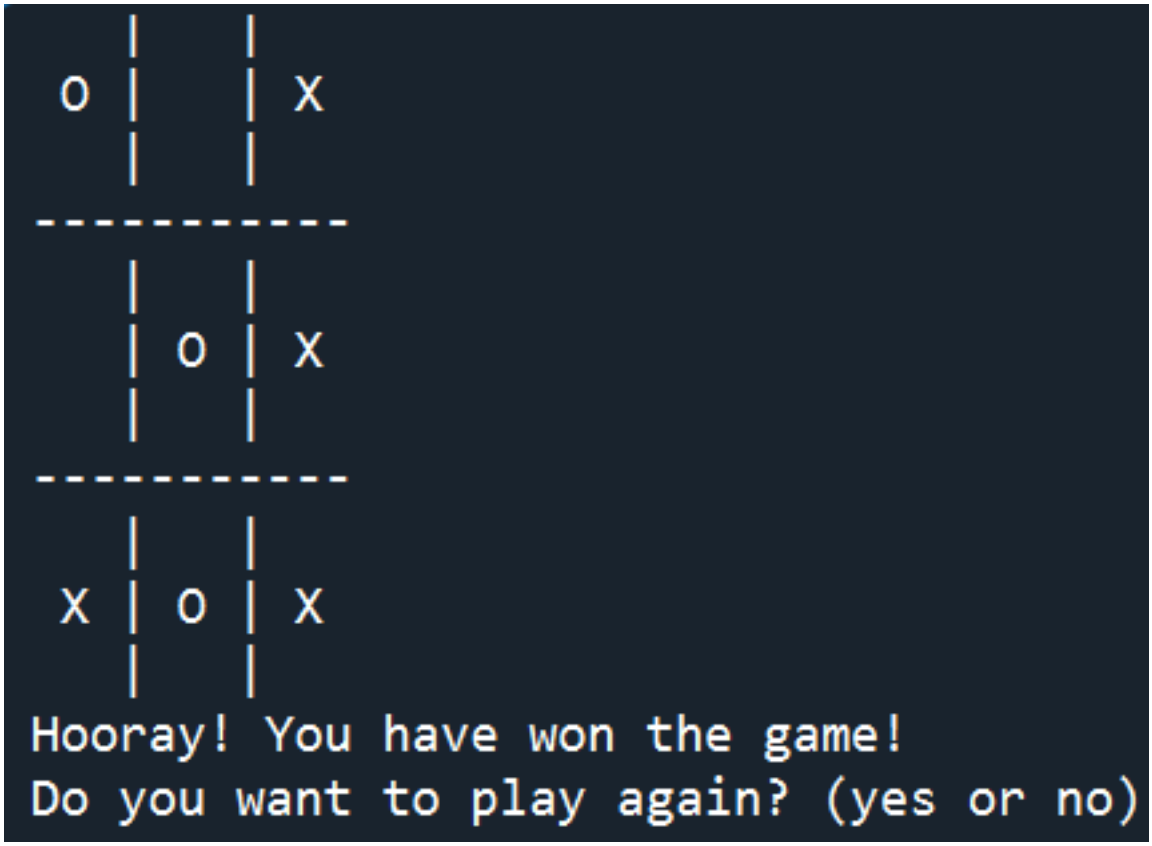
```
    # Try to take the center, if it is free.
    if isSpaceFree(board, 5):
        return 5

    # Move on one of the sides.
    return chooseRandomMoveFromList(board, [2, 4, 6, 8])

def isBoardFull(board):
    # Return True if every space on the board is filled. Otherwise return False.
    for i in range(1, 10):
        if isSpaceFree(board, i):
            return False
    return True


print('Welcome to Tic Tac Toe!')

while True:
    # Reset the board
    theBoard = [' '] * 10
    playerLetter, computerLetter = inputPlayerLetter()
    turn = whoGoesFirst()
    print('The ' + turn + ' will go first.')
    gameIsPlaying = True

    while gameIsPlaying:
        if turn == 'player':
            # Player's turn.
            drawBoard(theBoard)
```

```
        move = getPlayerMove(theBoard)
        makeMove(theBoard, playerLetter, move)

        if isWinner(theBoard, playerLetter):
            drawBoard(theBoard)
            print('Hooray! You have won the game!')
            gameIsPlaying = False
        else:
            if isBoardFull(theBoard):
                drawBoard(theBoard)
                print('The game is a tie!')
                break
            else:
                turn = 'computer'

    else:
        # Computer's turn.
        move = getComputerMove(theBoard, computerLetter)
        makeMove(theBoard, computerLetter, move)

        if isWinner(theBoard, computerLetter):
            drawBoard(theBoard)
            print('The computer has beaten you! You lose.')
            gameIsPlaying = False
        else:
            if isBoardFull(theBoard):
                drawBoard(theBoard)
                print('The game is a tie!')
                break
            else:
```

```
            turn = 'player'

if not playAgain():
    break
```




X | X | X
O | O |
X | | O
The computer has beaten you! You lose.
Do you want to play again? (yes or no)


O | | X
| O | X
X | O | X
Hooray! You have won the game!
Do you want to play again? (yes or no)