Experiment 2

- ***Ivan Dsouza - 9601-***
- *T.E. COMPS A(Batch C)*

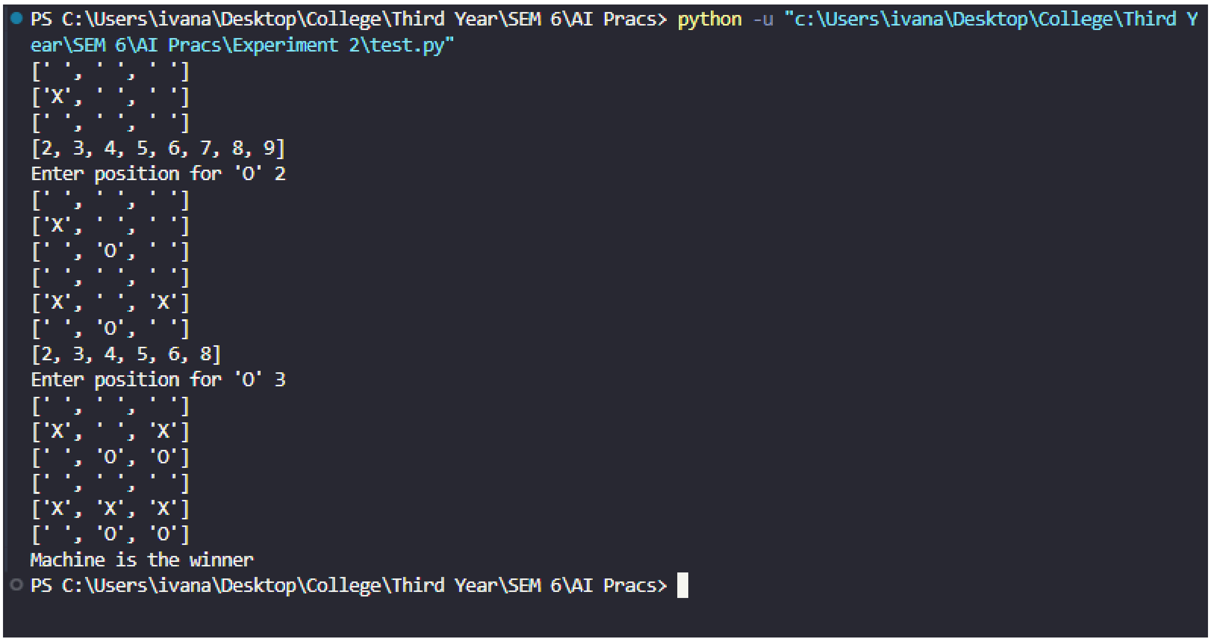
Tic-Tac-Toe by Magic Square Method

Code:

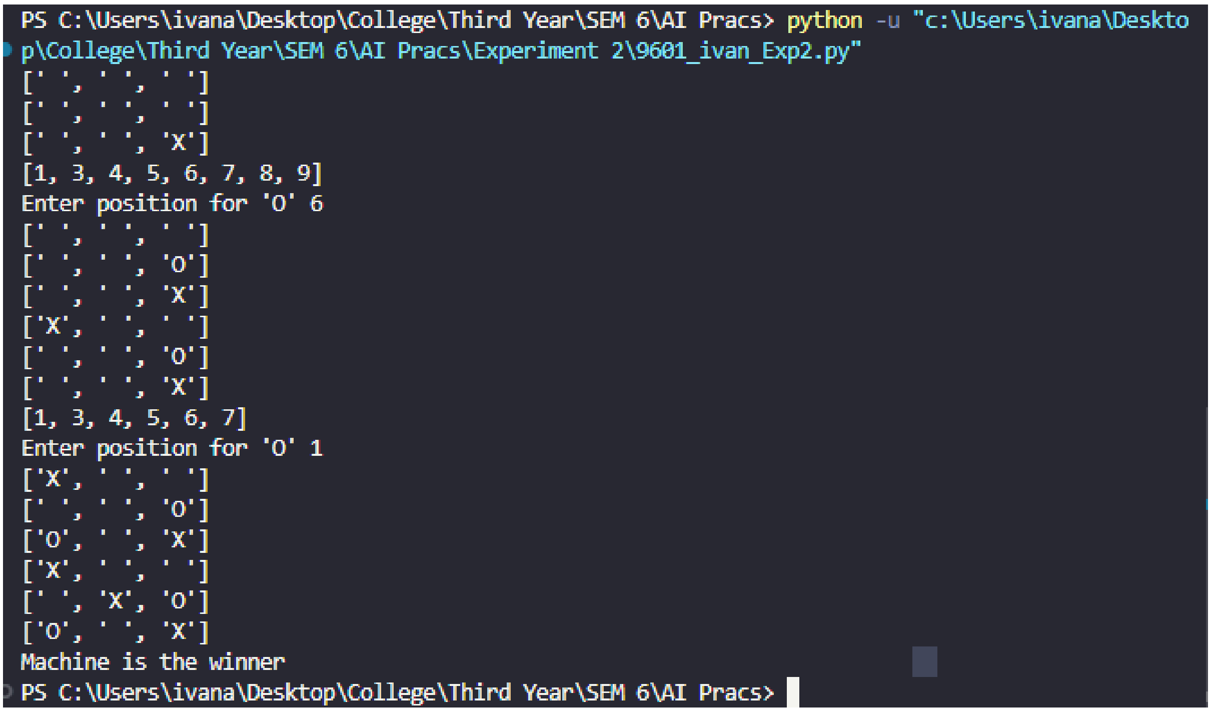
```
import random
x=[]
o=[]
empty =[1,2,3,4,5,6,7,8,9]
corner = [6,2,8,4]
board =[[" "," "," "],[" "," "," "],[" "," "," "]]
flag =True
game_over = False
temp = False
num_index={
 9: [0, 2],
 8: [0, 1],
 7: [0, 0],
 6: [1, 2],
 5: [1, 1],
 4: [1, 0],
 3: [2, 2],
 2: [2, 1],
 1: [2, 0]}
#do num magic change in name
index_magic ={
   9:4,
   8:3,
   7:8,
   6:9,
   5:5,
   4:1,
   3:2,
   2:7,
   1:6
 }
 #do  magic num change in name
magic_index ={
   4:9,
   3:8,
   8:7,
   9:6,
   5:5,
    1:4,
    2:3,
    7:2,
    6:1
 }
def winPoss(arr):
  for i in range(0,len(arr)):
      for j in range(i+1,len(arr)):
        ans = 15-(arr[i]+arr[j])
        for num in empty:
          if num == ans:
            return True, magic_index[ans]
  return False, 0
  i = 3
  while i>0:
   if flag  :
    if len(corner)==0:
       bot = random.choice(empty)
    else:
       bot = random.choice(corner)
       t=corner.remove(bot)
   else:
    bot = random.choice(empty)
  t=empty.remove(bot)
  x.append(bot)
  print(x)
  bot = magic_index[bot]
  board[num_index[bot][0]][num_index[bot][1]] ="X"
  print('\n'.join(map(str,board)))
  player = int(input("Enter position for 'O' "))
  empty.remove(index_magic[player])
  try:
      t = corner.remove(index_magic[player])
  except :
      pass
  o.append(index_magic[player])
  board[num_index[player][0]][num_index[player][1]]= "O"
  print('\n'.join(map(str,board)))
  if (player == 5):
    i=i-1
    flag = False
  i = i -1
while not game_over:
  winx,num = winPoss(x)
  if winx:
      game_over = True
      board[num_index[num][0]][num_index[num][1]] ="X"
      print('\n'.join(map(str,board)))
      t=empty.remove(index_magic[num])
      x.append(index_magic[num])
      print("Machine is the winner")
      break

  else :
      wino,num1 = winPoss(o)
      if wino:
        board[num_index[num1][0]][num_index[num1][1]] ="X"
        x.append(index_magic[num1])
        t=empty.remove(index_magic[num1])
        print('\n'.join(map(str,board)))
      else:
        bot = random.choice(empty)
        t=empty.remove(bot)
        bot = magic_index[bot]
        board[num_index[bot][0]][num_index[bot][1]] = "X"
        x.append(bot)
        print('\n'.join(map(str,board)))
  print(empty)
  if len(empty) == 0 :
   print("Draw")
   break
  player = int(input("Enter position for 'O' "))
  empty.remove(index_magic[player])
  o.append(index_magic[player])
  board[num_index[player][0]][num_index[player][1]]= "O"
  print('\n'.join(map(str,board)))
  for i in range(0,len(o)):
    for j in range (i+1,len(o)-1):
      sum = o[i]+o[j]+o[j+1]
      if sum == 15:
        temp =True
        break
  if temp :
      game_over = True
      winner = "x"
```

Output:



```
PS C:\Users\ivana\Desktop\College\Third Year\SEM 6\AI Pracs> python -u "c:\Users\ivana\Desktop\College\Third Y
ear\SEM 6\AI Pracs\Experiment 2\test.py"
[' ', ' ', ' ']
['X', ' ', ' ']
[' ', ' ', ' ']
[2, 3, 4, 5, 6, 7, 8, 9]
Enter position for 'O' 2
[' ', ' ', ' ']
['X', ' ', ' ']
[' ', 'O', ' ']
[' ', ' ', ' ']
['X', ' ', 'X']
[' ', 'O', ' ']
[2, 3, 4, 5, 6, 8]
Enter position for 'O' 3
[' ', ' ', ' ']
['X', ' ', 'X']
[' ', 'O', 'O']
[' ', ' ', ' ']
['X', 'X', 'X']
[' ', 'O', 'O']
Machine is the winner
PS C:\Users\ivana\Desktop\College\Third Year\SEM 6\AI Pracs>
```



```
PS C:\Users\ivana\Desktop\College\Third Year\SEM 6\AI Pracs> python -u "c:\Users\ivana\Deskto
p\College\Third Year\SEM 6\AI Pracs\Experiment 2\9601_ivan_Exp2.py"
[' ', ' ', ' ']
[' ', ' ', ' ']
[' ', ' ', 'X']
[1, 3, 4, 5, 6, 7, 8, 9]
Enter position for 'O' 6
[' ', ' ', ' ']
[' ', ' ', 'O']
[' ', ' ', 'X']
['X', ' ', ' ']
[' ', ' ', 'O']
[' ', ' ', 'X']
[1, 3, 4, 5, 6, 7]
Enter position for 'O' 1
['X', ' ', ' ']
[' ', ' ', 'O']
['O', ' ', 'X']
['X', ' ', ' ']
[' ', 'X', 'O']
['O', ' ', 'X']
Machine is the winner
PS C:\Users\ivana\Desktop\College\Third Year\SEM 6\AI Pracs>
```

Postlab:

1. Relationship between Tic-Tac-Toe and Magic Square:

Representation:

- In the context of the provided program, the Tic-Tac-Toe board is modeled as a 3x3 magic square. The cells in the magic square are analogous to the cells on the Tic-Tac-Toe board. Each number in the magic square corresponds to a potential move on the board.

Strategy:

- The magic square provides a strategic framework for the computer to make intelligent moves. The computer aims to place its coins in positions corresponding to numbers in the magic square, creating combinations that lead to victory or block the opponent from winning. This strategy is based on the unique properties of the magic square.

Winning Conditions:

- Just as in a traditional Tic-Tac-Toe game, where a player wins by having their coins in a line across the same row, column, or diagonal, the magic square simplifies the identification of winning conditions. The numbers in the magic square represent specific patterns that, if achieved on the board, result in a win.

2. Magic Square of Order n:

Definition:

- A magic square of order n is a square matrix of size n x n filled with distinct positive integers from 1 to n^2. The arrangement is such that the sum of numbers in each row, each column, and both main diagonals is the same, known as the "magic constant."
- Properties:
  - In the provided program, the magic square is of order 3, creating a 3x3 matrix.
  - Each row, column, and diagonal of the magic square adds up to the magic constant, which is 15 in this case.
  - The numbers in the magic square uniquely determine winning combinations and guide the computer's moves.
- Application to Tic-Tac-Toe:
  - The magic square is applied to Tic-Tac-Toe by associating the numbers in the magic square with the cells on the game board.
  - Players aim to achieve the magic constant by strategically placing their coins on the board according to the magic square.
  - The computer, using the magic square method, identifies potential winning moves and attempts to block the opponent, enhancing its chances of winning most of the time.