Department of Computer Engineering
Academic Term II: 23-24

Class: B.E (Computer), Sem – VI   Subject Name: Artificial Intelligence

Student Name: Royce Dmello   **Roll No: 9533**

Practical No:	2
Title:	Tic Tac Toe game implementation by Magic Square Method
Date of Performance:	2/2/24
Date of Submission:	8/2/24

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Marks
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Logic/Algorithm Complexity analysis (03)	03(Correct)	02(Partial)	01 (Tried)	
3	Coding Standards (03): Comments/indention/Naming conventions Test Cases /Output	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Assignment (03)	03(done well)	2 (Partially Correct)	1(submitted)	
Total					

Fr. Conceicao Rodrigues College of Engineering Fr. Agnel
Ashram, Bandstand, Bandra (W), Mumbai - 400050

Experiment No: 2

Title: Tic Tac Toe game implementation by Magic Square Method

Objective: To write a computer program in such a way that computer wins most of the time using Magic Square Method

Theory:

A player who places his coins first across the same row or same column or same diagonal wins the game. Let us take a magic square of order 3 x 3 (for 3 coins game). The sum of the numbers across rows, columns and diagonals are the same - it is 15. That is, a player who places his coins such that he gets the perfect score of 15 takes the prize.

1) Board is considered to be a magic square of size 3 X 3 with 9 blocks numbered by numbers indicated by the magic square.

2) This representation makes the process of checking for a possible win simpler.
Board Layout as magic square. Each row, column and diagonals add to 15.

8	3	4	**15**
1	5	9	**15**
6	7	2	**15**

3) Maintain the list of each player's blocks in which he has played.
Consider each pair of blocks that the player owns.
Compute difference D between 15 and the sum of the two blocks

If D < 0 or D > 9 then

   i) These two blocks are not collinear and so can be ignored.

      ii) Otherwise, if the block representing difference is blank (i.e., not in either list) then a move in that block will produce a win.

Code:

```python
import random

# Function to print the Tic Tac Toe board
def print_board(board):
    for row in board:
        print(" | ".join(row))
        if row != board[-1]:
            print("-" * 9)

# Function to check if a player has won
def check_win(board, player):
    for i in range(3):
        if all([cell == player for cell in board[i]]) or all([board[j][i] == player for j in range(3)]):
            return True
    if all([board[i][i] == player for i in range(3)]) or all([board[i][2 - i] == player for i in range(3)]):
        return True
    return False

# Function to check if the board is full
def is_board_full(board):
    return all([cell != " " for row in board for cell in row])

# Function to get the best move using Magic Square Method
def get_best_move(board):
    magic_square = [[8, 1, 6], [3, 5, 7], [4, 9, 2]]
    best_move = None
    max_score = -1
    for i in range(3):
        for j in range(3):
            if board[i][j] == " ":
                score = 0
                for k in range(3):
                    score += magic_square[i][k] if board[i][k] == "O" else 0
                    score += magic_square[k][j] if board[k][j] == "O" else 0
                score += magic_square[i][j] if i == j else 0
```

```python
            score += magic_square[i][2 - j] if i + j == 2 else 0
            if score > max_score:
                max_score = score
                best_move = (i, j)
    return best_move

# Function for the computer's turn
def computer_turn(board):
    move = get_best_move(board)
    board[move[0]][move[1]] = "O"
    print("Computer's move:")
    print_board(board)

# Function for the player's turn
def player_turn(board):
    while True:
        try:
            row = int(input("Enter row (1, 2, or 3): ")) - 1
            col = int(input("Enter column (1, 2, or 3): ")) - 1
            if 0 <= row < 3 and 0 <= col < 3 and board[row][col] == " ":
                board[row][col] = "X"
                print_board(board)
                break
            else:
                print("Invalid move! Try again.")
        except ValueError:
            print("Invalid input! Please enter a number.")

# Main function to control the game
def play_game():
    board = [[" " for _ in range(3)] for _ in range(3)]
    print("Welcome to Tic Tac Toe with Magic Square Method!")
    print_board(board)
    while True:
        player_turn(board)
        if check_win(board, "X"):
            print("Congratulations! You win!")
            break
        if is_board_full(board):
            print("It's a draw!")
            break
        computer_turn(board)
        if check_win(board, "O"):
            print("Computer wins! Better luck next time.")
            break

# Start the game
play_game()
```

OUTPUT:

Post Lab Assignment:

1. What is the relationship between tic-tac-toe and magic square?
2. What is a magic square of order n?

Name: Royce Dmello

Postlab Assignment Exp 2

Q.1.) What is the relationship between
tac-toe and magic square?
→ 1.) Tic-tac-toe and magic square
related through the arragnments
game board
2.) In Tic-tac-toe players aim to
winning combinations of their mar
rows, column and diagonals.
3.) A magic square is a grid where the
of numbers position on the tic-tac-
the same
4.) By using the numbers of a magic
we can easily identify winning con
in tic-tac-toe.

(Q2.) What is a meyic square of order?
→ A magic square is a square grid containi
arranged in a way that each row, column a
add up to the same constant sum.
The order of the magic square re
the numbers of rows and column
for a magic square of order n, it
2 rows and n columns
The number used in a magic square
n range from 1 to n²
The sum of each row, column, and di
in a magic square of order n is call
magic constant denoted by M.
formulas for calculating the magic
(M) of the magic square of order

$$M = \frac{n(n^2+1)}{2}$$

where M → magic constant
       n → order of magic squar