Department of Computer Science and Design
Rajalakshmi Engineering College

Ex. No. : **Date:**

Register No.: **Name:**

Create a tic tac toe game in a Simple User Interface

AIM:

To design and implement a simple two-player Tic Tac Toe game that demonstrates core programming concepts such as control structures, data handling, and user interaction.

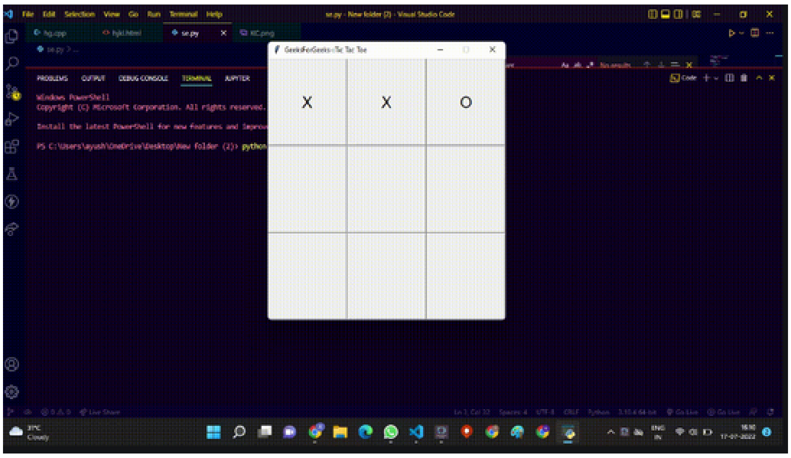
PROCEDURE:

- Define the Game Board
- Represent the board as a 3×3 matrix using a list of lists or a 2D array.
- Initialize all cells with a placeholder (e.g., empty string or dash).
- Set Up Player Turns
- Assign symbols to players (e.g., Player 1: X, Player 2: O).
- Use a loop to alternate turns between players.
- Accept Player Input
- Prompt the current player to enter their move (row and column).
- Validate input to ensure the selected cell is within bounds and unoccupied.
- Update the Board
- Place the player’s symbol in the chosen cell.
- Display the updated board after each move.
- Check for Win Conditions
- After each move, check rows, columns, and diagonals for three matching symbols.
- If a win is detected, declare the winner and end the game.
- Check for Draw
- If all cells are filled and no winner is found, declare the game a draw.
- Restart Option
- Provide an option to restart the game after a win or draw.

PROGRAM:

```
# Tic Tac Toe Game in Python

def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 5)

def check_winner(board, player):
    # Check rows and columns
    for i in range(3):
        if all(board[i][j] == player for j in range(3)) or \
           all(board[j][i] == player for j in range(3)):
            return True
    # Check diagonals
    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

def is_draw(board):
    return all(cell != " " for row in board for cell in row)

def main():
    board = [[" " for _ in range(3)] for _ in range(3)]
    current_player = "X"

        current_player = "O" if current_player == "X" else "X"

if __name__ == "__main__":
    main()
```

```
while True:
print_board(board)
print(f"Player {current_player}'s turn.")
try:
row = int(input("Enter row (0-2): "))
col = int(input("Enter column (0-2): "))
if board[row][col] != " ":
print("Cell already taken. Try again.")
continue
board[row][col] = current_player
except (ValueError, IndexError):
print("Invalid input. Try again.")
continue
if check_winner(board, current_player):
print_board(board)
print(f"Player {current_player} wins!")
break
elif is_draw(board):
print_board(board)
print("It's a draw!")
break
```


File Edit Selection View Go Run Terminal Help
se.py - New folder (2) - Visual Studio Code
hg.cpp
hjkl.html
se.py
KC.png
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
GeeksForGeeks-Tic Tac Toe
X
X
O

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improv

PS C:\Users\ayush\OneDrive\Desktop\New folder (2)> python
```

RESULT:

A multimedia application was successfully developed using Python that integrates image display, audio playback, and video streaming into a unified, interactive interface.