

Project Report

Student Name: Karan

Branch: MCA General

Semester: 1st

Subject Name: PYTHON PROGRAMMING LAB

UID: 24MCA20147

Section/Group: 3A

Date of Performance: 2-Oct

Subject Code: 24CAH-606

1. Aim of the project: Tic Tac Toe game Showdown

2. Software Requirements:

Python Interpreter: Install the desired version of Python from python.org (e.g., Python 3.12.5)
Ensure Python is added to your system's PATH.

Development Environment: Choose an integrated development environment (IDE) or text editor that supports python. Popular choices include: PyCharm, Visual Studio Code, Jupyter notebook

3. Program Logic:

The program logic for a Tic Tac Toe game can be structured into several key components. Below is a detailed breakdown of the logic:

1. Initialization:-

- Create a 3x3 game board, typically represented as a list of lists in Python. - Initialize the board with empty values (e.g., spaces or zeros).

2. Display the Board:- - Create a function to print the current state of the board to the console in a user-friendly format.

3. Player Input:-

- Alternate turns between two players (Player X and Player O).
 - Prompt the current player to enter their move (specifying the row and column). - Validate the input to ensure:
 - The input is within the bounds of the board (0 to 2).
 - The chosen cell is empty (not already occupied by X or O).
4. Update the Board:- - Place the current player's symbol (X or O) in the chosen cell on the board.
5. Check for Win Conditions
- After each move, check if the current player has won the game by:
 - Checking all rows for three of the same symbol.
 - Checking all columns for three of the same symbol.
 - Checking the two diagonals for three of the same symbol.
6. Check for Draw:- - If all cells are filled and there is no winner, declare the game a draw.
7. Game Loop:-
- Repeat the process of displaying the board, taking input, updating the board, and checking for win/draw conditions until the game ends.
8. End Game:-
- Announce the winner or if the game is a draw.
 - Optionally, ask if the players want to play again and reset the board if they do.

CODE:-

```
import tkinter as tk
from tkinter import messagebox

class TicTacToe:
    def __init__(self, root):
        self.root = root
        self.root.title("Tic-Tac-Toe")
```

```
self.board = [""] * 9  
self.current_player = "X"  
self.buttons = []
```

```
self.create_board()
```

```
def create_board(self):
```

```
    """Creates the 3x3 grid of buttons"""
```

```
    row = 0
```

```
    col = 0
```

```
    for i in range(9):
```

```
        button = tk.Button(self.root, text="", width=10, height=3, font=("Arial", 24),  
                           command=lambda i=i: self.on_button_click(i))
```

```
        button.grid(row=row, column=col)
```

```
        self.buttons.append(button)
```

```
        col += 1
```

```
        if col > 2:
```

```
            col = 0
```

```
            row += 1
```

```
def on_button_click(self, index):
```

```
    """Handles the click event on a button"""
```

```
    if self.board[index] == "":
```

```
        self.board[index] = self.current_player
```

```
        self.buttons[index].config(text=self.current_player)
```

```
        if self.check_winner(self.current_player):
```

```
            self.show_winner(self.current_player)
```

```
        elif all(spot != "" for spot in self.board):
```

```
            self.show_winner("Draw")
```

```
        else:
```

```
            self.switch_player()
```

```
def switch_player(self):
```

```
    """Switches the current player"""
```

```
    self.current_player = "O" if self.current_player == "X" else "X"
```

```
def check_winner(self, player):
```

```
"""Checks if the given player has won the game"""
```

```
win_conditions = [
```

```
    [0, 1, 2],
```

```
    [3, 4, 5],
```

```
    [6, 7, 8],
```

```
    [0, 3, 6],
```

```
    [1, 4, 7],
```

```
    [2, 5, 8],
```

```
    [0, 4, 8],
```

```
    [2, 4, 6],
```

```
]
```

```
for condition in win_conditions:
```

```
    if all(self.board[i] == player for i in condition):
```

```
        return True
```

```
return False
```

```
def show_winner(self, winner):
```

```
    """Displays a messagebox showing who won the game"""
```

```
    if winner == "Draw":
```

```
        messagebox.showinfo("Game Over", "It's a Draw!")
```

```
    else:
```

```
        messagebox.showinfo("Game Over", f"Player {winner} wins!")
```

```
    self.reset_game()
```

```
def reset_game(self):
```

```
    """Resets the game for a new round"""
```

```
    self.board = [""] * 9
```

```
    for button in self.buttons:
```

```
        button.config(text="")
```

```
    self.current_player = "X"
```

```
def main():
```

```
    root = tk.Tk()
```

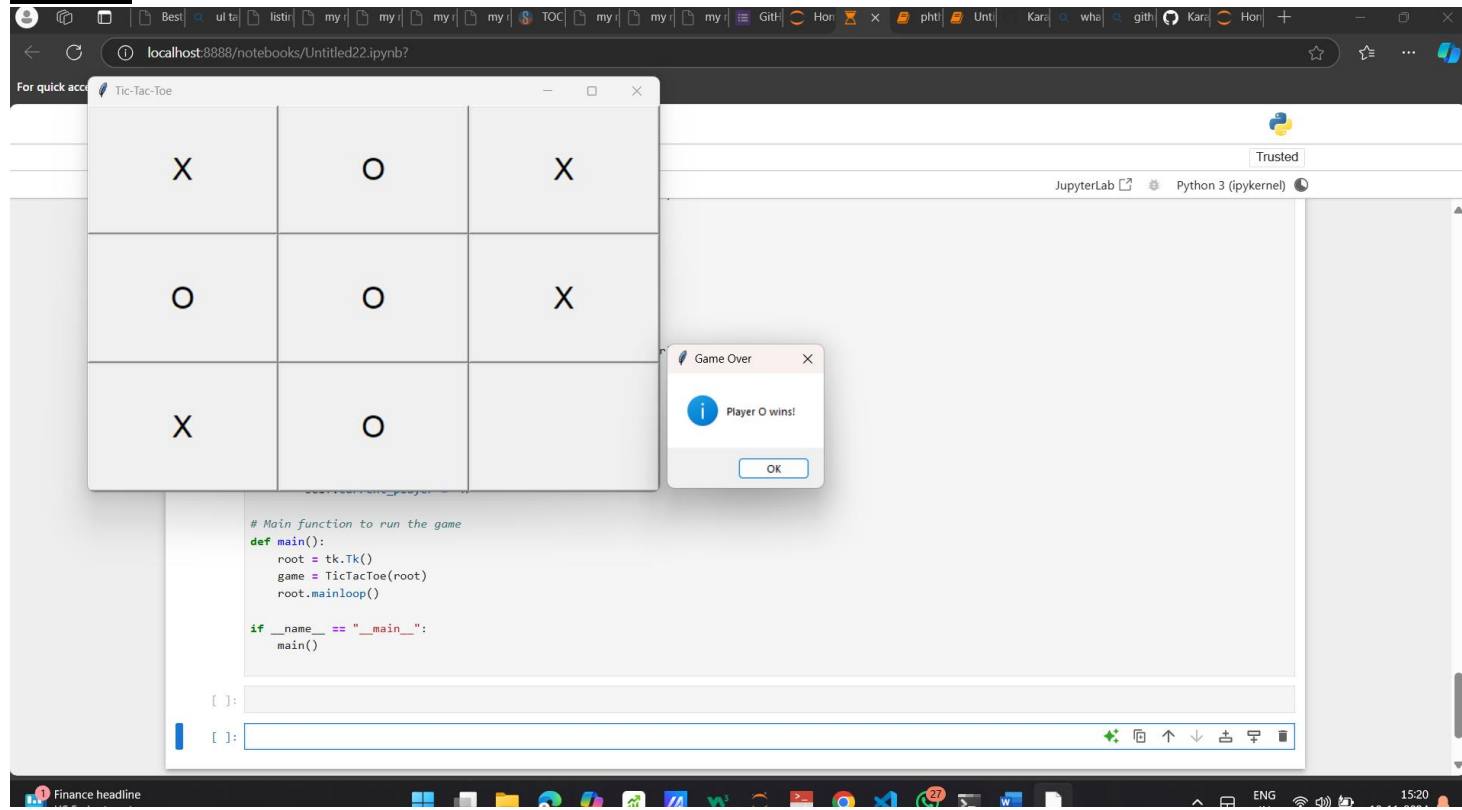
```
    game = TicTacToe(root)
```

```
    root.mainloop()
```

```
if __name__ == "__main__":
```

```
    main()
```

Output:-



1. Learning outcomes (What I have learnt):

1. Strategic Thinking : Players learn to think ahead and plan their moves. They must consider not only their own strategy but also anticipate their opponent's moves.
2. Problem Solving : The game encourages players to solve problems and make decisions based on the current state of the game. They must evaluate different possible moves and their consequences.
3. Critical Thinking: Players develop critical thinking skills as they analyze the game board and determine the best course of action. They learn to weigh options and make informed decisions.
4. Pattern Recognition: Players become adept at recognizing patterns, which is crucial for both offense (creating a winning line) and defense (blocking the opponent).

5. **Mathematical Skills:** The game can enhance basic mathematical skills, such as counting and spatial awareness, as players visualize the grid and potential outcomes.
6. **Sportsmanship:** Playing Tic Tac Toe can teach players about winning and losing gracefully, fostering a sense of fair play and respect for opponents.
7. **Memory and Concentration:** Players must remember previous moves and concentrate on the game, which can improve cognitive skills.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet		8 Marks
2.	Viva		10 Marks
3.	Simulation		12 Marks
4.	Total		30 Marks

Teacher's Signature