# Tic-Tac-Toe Game using Pygame - Code with Documentation

Imports the required libraries and initializes Pygame to start the game.

import pygame  
import sys  
  
# Initialize Pygame  
pygame.init()

Defines constants for the screen size, board layout, drawing measurements, and spacing for figures.

# Constants  
WIDTH, HEIGHT = 300, 300  
LINE\_WIDTH = 5  
BOARD\_ROWS = 3  
BOARD\_COLS = 3  
SQUARE\_SIZE = WIDTH // 3  
CIRCLE\_RADIUS = SQUARE\_SIZE // 3  
CIRCLE\_WIDTH = 10  
CROSS\_WIDTH = 15  
SPACE = SQUARE\_SIZE // 4

Defines color values used throughout the game using RGB tuples.

# Colors  
BG\_COLOR = (28, 170, 156)  
LINE\_COLOR = (23, 145, 135)  
CIRCLE\_COLOR = (239, 231, 200)  
CROSS\_COLOR = (84, 84, 84)

Sets up the Pygame display window, assigns a title, and fills the screen with the background color.

# Screen setup  
screen = pygame.display.set\_mode((WIDTH, HEIGHT))  
pygame.display.set\_caption("Tic Tac Toe")  
screen.fill(BG\_COLOR)

Initializes the game board as a 3x3 grid filled with zeros (empty cells).

# Board  
board = [[0 for \_ in range(BOARD\_COLS)] for \_ in range(BOARD\_ROWS)]

Defines the `draw\_lines` function to draw the tic-tac-toe grid lines.

# Draw grid lines  
def draw\_lines():  
 ...

Defines the `draw\_figures` function to draw Xs and Os depending on the player's turn.

# Draw Xs and Os  
def draw\_figures():  
 ...

Defines logic to check if a player has won the game either horizontally, vertically, or diagonally.

# Check for win  
def check\_win(player):  
 ...

Checks if all cells are filled and no player has won, indicating a draw.

# Check for draw  
def is\_draw():  
 ...

Resets the board and redraws the initial grid, allowing a new game to start.

# Restart game  
def restart\_game():  
 ...

Draws the initial grid lines and sets the first player and game state.

# Initial Draw  
draw\_lines()  
current\_player = 1 # Player 1 -> X, Player 2 -> O  
game\_over = False

Main game loop: listens for events (mouse clicks and key presses), updates the game state, and redraws the board as needed.

# Game Loop  
while True:  
 for event in pygame.event.get():  
 ...  
 draw\_figures()  
 pygame.display.update()