```
fimport pygame as pg,sys
from pygame.locals import *
import time

#initialize global variables
XO = 'x'
winner = None
draw = False
width = 400
height = 400
white = (255, 255, 255)
line_color = (10,10,10)

#TicTacToe 3x3 board
TTT = [[None]*3,[None]*3,[None]*3]

#initializing pygame window
pg.init()
fps = 30
CLOCK = pg.time.Clock()
screen = pg.display.set_mode((width, height+100),0,32)
pg.display.set_caption("Tic Tac Toe")

#loading the images
opening = pg.image.load('tic tac opening.png')
x_img = pg.image.load('x.png')
o_img = pg.image.load('o.png')

#resizing images
x_img = pg.transform.scale(x_img, (80,80))
o_img = pg.transform.scale(o_img, (80,80))
opening = pg.transform.scale(opening, (width, height+100))


def game_opening():
    screen.blit(opening,(0,0))
    pg.display.update()
    time.sleep(1)
    screen.fill(white)

    # Drawing vertical lines
    pg.draw.line(screen,line_color,(width/3,0),(width/3, height),7)
    pg.draw.line(screen,line_color,(width/3*2,0),(width/3*2, height),7)
    # Drawing horizontal lines
    pg.draw.line(screen,line_color,(0,height/3),(width, height/3),7)
    pg.draw.line(screen,line_color,(0,height/3*2),(width, height/3*2),7)
    draw_status()


def draw_status():
    global draw

    if winner is None:
        message = XO.upper() + "'s Turn"
    else:
        message = winner.upper() + " won!"
    if draw:
        message = 'Game Draw!'

    font = pg.font.Font(None, 30)
    text = font.render(message, 1, (255, 255, 255))

    # copy the rendered message onto the board
    screen.fill ((0, 0, 0), (0, 400, 500, 100))
    text_rect = text.get_rect(center=(width/2, 500-50))
    screen.blit(text, text_rect)
    pg.display.update()

def check_win():
    global TTT, winner,draw

    # check for winning rows
    for row in range (0,3):
        if ((TTT [row][0] == TTT[row][1] == TTT[row][2]) and(TTT [row][0] is not 
None)):
            # this row won
            winner = TTT[row][0]
            pg.draw.line(screen, (250,0,0), (0, (row + 1)*height/3 -height/6),\
                              (width, (row + 1)*height/3 - height/6 ), 4)
            break

    # check for winning columns
    for col in range (0, 3):
        if (TTT[0][col] == TTT[1][col] == TTT[2][col]) and (TTT[0][col] is not None):
            # this column won
            winner = TTT[0][col]
            #draw winning line
            pg.draw.line (screen, (250,0,0),((col + 1)* width/3 - width/6, 0),\
                          ((col + 1)* width/3 - width/6, height), 4)
            break

    # check for diagonal winners
    if (TTT[0][0] == TTT[1][1] == TTT[2][2]) and (TTT[0][0] is not None):
        # game won diagonally left to right
        winner = TTT[0][0]
        pg.draw.line (screen, (250,70,70), (50, 50), (350, 350), 4)


    if (TTT[0][2] == TTT[1][1] == TTT[2][0]) and (TTT[0][2] is not None):
        # game won diagonally right to left
        winner = TTT[0][2]
        pg.draw.line (screen, (250,70,70), (350, 50), (50, 350), 4)

    if(all([all(row) for row in TTT]) and winner is None ):
        draw = True
    draw_status()


def drawXO(row,col):
    global TTT,XO
    if row==1:
        posx = 30
    if row==2:
        posx = width/3 + 30
    if row==3:
        posx = width/3*2 + 30

    if col==1:
        posy = 30
    if col==2:
        posy = height/3 + 30
    if col==3:
        posy = height/3*2 + 30
    TTT[row-1][col-1] = XO
    if(XO == 'x'):
        screen.blit(x_img,(posy,posx))
        XO= 'o'
    else:
        screen.blit(o_img,(posy,posx))
        XO= 'x'
    pg.display.update()
    #print(posx,posy)
    #print(TTT)



def userClick():
    #get coordinates of mouse click
    x,y = pg.mouse.get_pos()

    #get column of mouse click (1-3)
    if(x<width/3):
        col = 1
    elif (x<width/3*2):
        col = 2
    elif(x<width):
        col = 3
    else:
        col = None

    #get row of mouse click (1-3)
    if(y<height/3):
        row = 1
    elif (y<height/3*2):
        row = 2
    elif(y<height):
        row = 3
    else:
        row = None
    #print(row,col)


    if(row and col and TTT[row-1][col-1] is None):
        global XO

        #draw the x or o on screen
        drawXO(row,col)
        check_win()
```

```
def reset_game():
    global TTT, winner,XO, draw
    time.sleep(3)
    XO = 'x'
    draw = False
    game_opening()
    winner=None
    TTT = [[None]*3,[None]*3,[None]*3]


game_opening()

# run the game loop forever
while(True):
    for event in pg.event.get():
        if event.type == QUIT:
            pg.quit()
            sys.exit()
        elif event.type == MOUSEBUTTONDOWN:
            # the user clicked; place an X or O
            userClick()
            if(winner or draw):
                reset_game()

    pg.display.update()
    CLOCK.tick(fps)
```