```
import pygame
import sys
import tictactoe as ttt

pygame.init()
size = width, height = 600, 400

# Colors
black = (0, 0, 0)
white = (255, 255, 255)

screen = pygame.display.set_mode(size)

# Use system fonts to avoid missing .ttf errors
mediumFont = pygame.font.SysFont("Arial", 28)
largeFont = pygame.font.SysFont("Arial", 40)
moveFont = pygame.font.SysFont("Arial", 60)

user = None
board = ttt.initial_state()

def draw_buttons():
    playXButton = pygame.Rect((width / 8), (height / 2), width / 4, 50)
    playX = mediumFont.render("Play as X", True, black)
    playXRect = playX.get_rect()
    playXRect.center = playXButton.center
    pygame.draw.rect(screen, white, playXButton)
    screen.blit(playX, playXRect)

    playOButton = pygame.Rect(5 * (width / 8), (height / 2), width / 4, 50)
    playO = mediumFont.render("Play as O", True, black)
    playORect = playO.get_rect()
    playORect.center = playOButton.center
    pygame.draw.rect(screen, white, playOButton)
    screen.blit(playO, playORect)
    return playXButton, playOButton

def draw_board(board):
    tile_size = 80
    tile_origin = (width / 2 - (1.5 * tile_size), height / 2 - (1.5 * tile_size))
    tiles = []
    for i in range(3):
        row = []
```

```
            for j in range(3):
                rect = pygame.Rect(
                    tile_origin[0] + j * tile_size,
                    tile_origin[1] + i * tile_size,
                    tile_size, tile_size
                )
                pygame.draw.rect(screen, white, rect, 3)

                if board[i][j] != ttt.EMPTY:
                    move = moveFont.render(board[i][j], True, white)
                    moveRect = move.get_rect()
                    moveRect.center = rect.center
                    screen.blit(move, moveRect)
                row.append(rect)
            tiles.append(row)
        return tiles

    def draw_title(text):
        title = largeFont.render(text, True, white)
        titleRect = title.get_rect()
        titleRect.center = (int(width / 2), 30)
        screen.blit(title, titleRect)

    def draw_play_again_button():
        againButton = pygame.Rect(width / 3, height - 65, width / 3, 50)
        again = mediumFont.render("Play Again", True, black)
        againRect = again.get_rect()
        againRect.center = againButton.center
        pygame.draw.rect(screen, white, againButton)
        screen.blit(again, againRect)
        return againButton

    while True:
        screen.fill(black)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                sys.exit()

            elif event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
                mouse = pygame.mouse.get_pos()

                if user is None:
```

```
            playXButton, playOButton = draw_buttons()
            if playXButton.collidepoint(mouse):
                user = ttt.X
            elif playOButton.collidepoint(mouse):
                user = ttt.O

        else:
            if not ttt.terminal(board) and user == ttt.player(board):
                tiles = draw_board(board)
                for i in range(3):
                    for j in range(3):
                        if board[i][j] == ttt.EMPTY and tiles[i][j].collidepoint(mouse):
                            board = ttt.result(board, (i, j))

            if ttt.terminal(board):
                againButton = draw_play_again_button()
                if againButton.collidepoint(mouse):
                    user = None
                    board = ttt.initial_state()

    if user is None:
        # Draw player selection buttons
        draw_buttons()
        draw_title("Play Tic-Tac-Toe")

    else:
        tiles = draw_board(board)

        game_over = ttt.terminal(board)
        current_player = ttt.player(board)

        if game_over:
            w = ttt.winner(board)
            if w is None:
                draw_title("Game Over: Tie.")
            else:
                draw_title(f"Game Over: {w} wins.")
            draw_play_again_button()
        else:
            if user == current_player:
                draw_title(f"Your turn ({user})")
            else:
                draw_title("Computer thinking...")
```

```
            # AI move immediately
            move = ttt.minimax(board)
            if move is not None:
                board = ttt.result(board, move)

    pygame.display.flip()
```