*# Name: Shantanu Rohile*

*# Roll No: 53*

*# ==============================================*

*# AI Practical 05*

*# Problem Statement:*

*# Write a program to develop a Tic-Tac-Toe game # using the appropriate concepts of Game Theory.*

*# ==============================================*

*#*

*# Importing Libraries #*

*import* pygame

*import* sys

*#*

*# Pygame Initialization #*

pygame.init()

*#*

*# Constants and Global Variables #*

WIDTH, HEIGHT = 300, 300

GRID\_SIZE = 3

CELL\_SIZE = WIDTH // GRID\_SIZE

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

*# Create the game window*

screen = pygame.display.set\_mode((WIDTH, HEIGHT)) pygame.display.set\_caption("Tic-Tac-Toe")

*# Initialize game board*

board = [[' ' *for* \_ *in* range(GRID\_SIZE)] *for* \_ *in* range(GRID\_SIZE)] turn = 'X' *# Player X starts*

*#*

*# Functions #*

*# Draw grid lines*

def **draw\_grid**():

*for* i *in* range(1, GRID\_SIZE):

pygame.draw.line(screen, BLACK, (i \* CELL\_SIZE, 0), (i \* CELL\_SIZE, HEIGHT), 2) pygame.draw.line(screen, BLACK, (0, i \* CELL\_SIZE), (WIDTH, i \* CELL\_SIZE), 2)

*# Draw X or O on the board*

def **draw\_symbol**(*row*, *col*, *symbol*):

font = pygame.font.Font(None, 100) text = font.render(*symbol*, True, BLACK)

text\_rect = text.get\_rect(*center*=((*col* \* CELL\_SIZE) + CELL\_SIZE // 2, (*row* \* CELL\_SIZE) + CELL\_SIZE // 2))

screen.blit(text, text\_rect)

*# Check if a player has won* def **check\_winner**(*symbol*): *for* i *in* range(GRID\_SIZE):

*if* all(board[i][j] == *symbol for* j *in* range(GRID\_SIZE)) or \ all(board[j][i] == *symbol for* j *in* range(GRID\_SIZE)): *return* True

*if* all(board[i][i] == *symbol for* i *in* range(GRID\_SIZE)) or \ all(board[i][GRID\_SIZE - 1 - i] == *symbol for* i *in* range(GRID\_SIZE)): *return* True

*return* False

*# Check for draw condition*

def **is\_board\_full**():

*return* all(board[i][j] != ' ' *for* i *in* range(GRID\_SIZE) *for* j *in* range(GRID\_SIZE))

*# Reset the board*

def **reset\_game**():

global board

board = [[' ' *for* \_ *in* range(GRID\_SIZE)] *for* \_ *in* range(GRID\_SIZE)]

*#*

*# Game Loop #*

running = True

*while* running:

*for* event *in* pygame.event.get():

*if* event.type == pygame.QUIT:

running = False

*elif* event.type == pygame.MOUSEBUTTONDOWN and event.button == 1: mouseX, mouseY = event.pos

clicked\_row = mouseY // CELL\_SIZE clicked\_col = mouseX // CELL\_SIZE

*if* board[clicked\_row][clicked\_col] == ' ': board[clicked\_row][clicked\_col] = turn

*if* check\_winner(turn): print(f'{turn} wins!') pygame.time.wait(1000) reset\_game()

*elif* is\_board\_full():

print("It's a draw!") pygame.time.wait(1000) reset\_game()

*else*:

turn = 'O' *if* turn == 'X' *else* 'X'

*# Drawing board state* screen.fill(WHITE) draw\_grid()

*for* row *in* range(GRID\_SIZE):

*for* col *in* range(GRID\_SIZE):

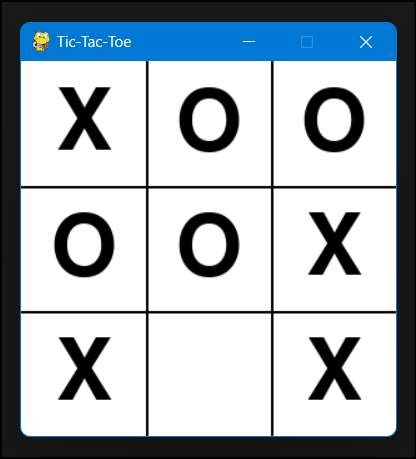
*if* board[row][col] != ' ': draw\_symbol(row, col, board[row][col])

pygame.display.flip()

*# Exit* pygame.quit() sys.exit()

*#*

*# Output:*



*#*