

#Name:-Sarthak Pagar

#Roll No.:-40

#Class:-TE(IT)

#Assignment No.5:-Write a program for the Tic-Tac-Toe game using the appropriate concepts from Game Theory

#Write a program for the Tic-Tac-Toe game.\

```
import pygame
```

```
import sys
```

```
import random
```

```
# Initialize Pygame
```

```
pygame.init()
```

```
# Constants
```

```
WIDTH, HEIGHT = 300, 300
```

```
GRID_SIZE = 3
```

```
CELL_SIZE = WIDTH // GRID_SIZE
```

```
# Colors
```

```
WHITE = (255, 255, 255)
```

```
BLACK = (0, 0, 0)
```

```
RED = (255, 0, 0)
```

```
# Create the game window
```

```
screen = pygame.display.set_mode((WIDTH, HEIGHT))
```

```
pygame.display.set_caption("Tic-Tac-Toe")
```

```
# Initialize the game board
```

```
board = [[' ' for _ in range(GRID_SIZE)] for _ in range(GRID_SIZE)]
```

```
# Function to draw the 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)
```

```
# Function to draw X or O in a cell
```

```
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)
```

```
# Function to check for a win
```

```
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

# Function to check for a draw
def is_board_full():
    return all(board[i][j] != ' ' for i in range(GRID_SIZE) for j in range(GRID_SIZE))

# Function to reset the game
def reset_game():
    global board
    board = [[' ' for _ in range(GRID_SIZE)] for _ in range(GRID_SIZE)]

# Main game loop
turn = 'X'
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: # Left mouse
button clicked
            mouseX, mouseY = event.pos
            clicked_row = mouseY // CELL_SIZE
            clicked_col = mouseX // CELL_SIZE

            if 0 <= clicked_row < GRID_SIZE and 0 <= clicked_col < GRID_SIZE and
board[clicked_row][clicked_col] == ' ':
                board[clicked_row][clicked_col] = turn

            # Check for a win
            if check_winner(turn):
                print(f'{turn} wins!')
                reset_game()

            # Check for a draw
            elif is_board_full():
                print("It's a draw!")
                reset_game()

            # Switch turn
            turn = 'O' if turn == 'X' else 'X'

# Draw the grid
screen.fill(WHITE)
draw_grid()

# Draw X or O in each cell
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()
```

```
# Quit Pygame
```

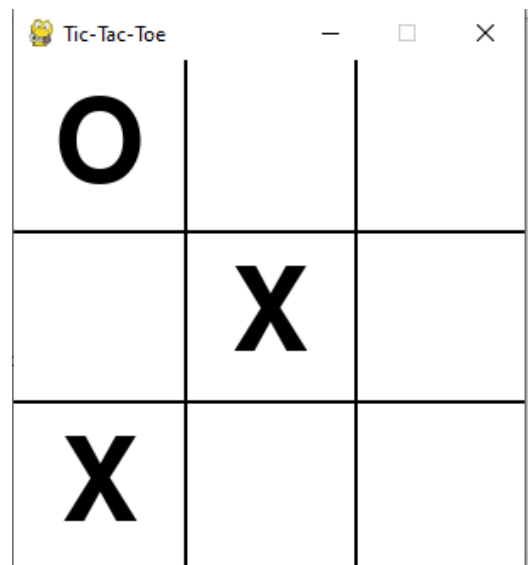
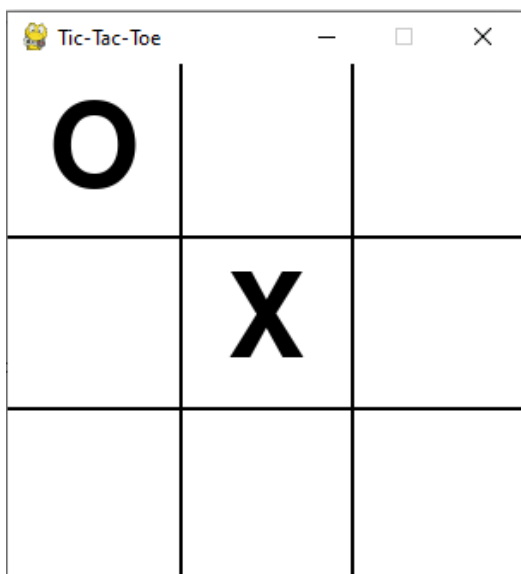
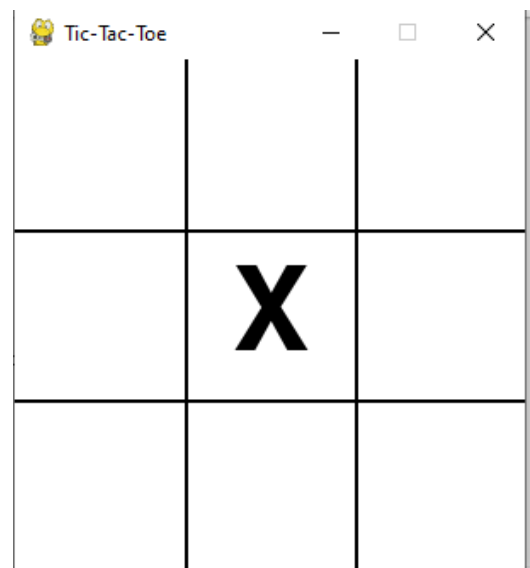
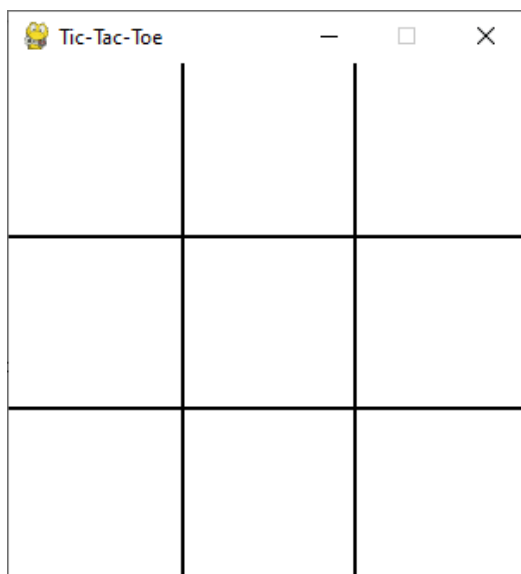
```
pygame.quit()
```

```
sys.exit()
```

#Output:-

pygame 2.6.1 (SDL 2.28.4, Python 3.11.3)

Hello from the pygame community. <https://www.pygame.org/contribute.html>



Tic-Tac-Toe		
O		O
	X	
X		

Tic-Tac-Toe		
O	X	O
	X	
X		

Tic-Tac-Toe		
O	X	O
	X	
X	O	

Tic-Tac-Toe		
O	X	O
	X	X
X	O	

Tic-Tac-Toe		
O	X	O
O	X	X
X	O	

it's a draw!