Report on TIC TAC TOE Game

Submitted to:

Sagar Dhanraj Pande

Lovely Professional University

Submitted by:

Amit Kalita

ROLL NO-02

SEC-K19HZ

REGISTRATION NO-11910112

Table of Contents. ..... Page

ACKNOLEDGEMENTS

Presentation inspiration and motivation have always played a key role in success of any venture.

I express my sincere thanks to **Sagar Dhanraj Pande, lovely professional university**

I pay my deep sense of gratitude to encourage me to the highest peak and to provide me the opportunity to prepare my project. I am immensely obliged to my friend for their elevating inspiration, encouraging guidance and kind supervision in the completion of my project.

I fell to acknowledge my indebtedness and deep sense of gratitude to **Sagar Dhanraj pande** whose valuable guidance and kind supervision given to me throughout the course which shaped the present work as its show.

Last but not least, **my parents** are also an important inspiration for me. So with due regards, I express my gratitude to them.

Summary of Game

Game Strategy

Tic-Tac-Toe game has many strategies that can be used. The main point of the strategy is the players have to block the opponent fork, either horizontally, vertically, or diagonally, while the players have to find their own fork to win.

Strategy

On playing tic-tac-toe, tactics and strategy are a must in order to win.

The simplest tactic is to complete a three-in-a-row: if you have two of your symbols lined up in a row (either horizontal, vertical, or diagonal) and the remaining square is empty, play on it, giving you the win. If you have two of your symbols lined up, and the last square is empty, this is called a **threat**.

The above tactic gives rise to **blocking**: if the opponent has two of their symbols lined up and the remaining square is empty, you must block it by playing on the last remaining square.

Given the blocking tactic, it's pretty natural to consider creating two different threats so that the opponent cannot block both at the same time. This is called a **fork** (as it has two (or more) directions, like a dining fork splitting from one handle to several tines; also compare with fork in chess, which is to make a threat in multiple directions).

SOURSE CODE

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

pygame.mixer.init()
pygame.mixer.music.load('TunePocket-Action-Trailer-Logo-Preview.mp3')
pygame.mixer.music.play()


XO = 'x'
winner = None
draw = None
width = 400
height = 400
aqua = (0, 255, 255)
RGB = (252, 3, 215)
line_color = (0, 0, 0)
board = [[None] * 3, [None] * 3, [None] * 3]
pg.init()
fps = 30
CLOCK = pg.time.Clock()
screen = pg.display.set_mode((width, height + 100), 0, 32)
pg.display.set_caption("My Tic Tac Toe")

initiating_window = pg.image.load("GameTile.jpg")
x_img = pg.image.load("X_-_Nicky_Jam_&_J_Balvin.jpg")
y_img = pg.image.load("40297482-stock-vector--letter-o-design-template-vector-
illustration.jpg")

initiating_window = pg.transform.scale(initiating_window, (width, height + 100))
x_img = pg.transform.scale(x_img, (80, 80))
o_img = pg.transform.scale(y_img, (80, 80))


def game_initiating_window():
    screen.blit(initiating_window, (0, 0))

    pg.display.update()
    time.sleep(3)
    screen.fill(aqua)

    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)

    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)
```

```
    # screen.fill(RGB)
    # pygame.display.update()

    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 !"
        pygame.mixer.music.load('SUPER MARIO - game over - sound effect (online-
audio-converter.com).mp3')
        pygame.mixer.music.play()

    font = pg.font.Font(None, 30)

    text = font.render(message, 1, (255, 255, 255))

    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 board, winner, draw

    for row in range(0, 3):
        if ((board[row][0] == board[row][1] == board[row][2]) and (board[row][0]
is not None)):
            winner = board[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


    for col in range(0, 3):
        if ((board[0][col] == board[1][col] == board[2][col]) and (board[0][col]
is not None)):
            winner = board[0][col]
            pg.draw.line(screen, (250, 0, 0), ((col + 1) * width / 3 - width / 6,
0),\
                         ((col + 1) * width / 3 - width / 6, height), 4)

 break
```

```
if (board[0][0] == board[1][1] == board[2][2]) and (board[0][0] is not None):
        winner = board[0][0]
        pg.draw.line(screen, (250, 70, 70), (50, 50), (350, 350), 4)

    if (board[0][2] == board[1][1] == board[2][0]) and (board[0][2] is not None):

        winner = board[0][2]
        pg.draw.line(screen, (250, 70, 70), (350, 50), (50, 350), 4)

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


    draw_status()


def drawXO(row, col):
    global board, 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

    board[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()

def user_click():
    x, y = pg.mouse.get_pos()
```

```
    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

    if (row and col and board[row - 1][col - 1] is None):
        global XO
        drawXO(row, col)

        pygame.mixer.music.load('beep-06.mp3')
        pygame.mixer.music.play()

        check_win()

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

game_initiating_window()

while (True):
    for event in pg.event.get():
        if event.type == QUIT:

            pg.quit()
            sys.exit()
elif event.type is MOUSEBUTTONDOWN:
            user_click()
            if (winner or draw):
                reset_game()
```

```
# screen.fill(RGB)
# pygame.display.update()

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

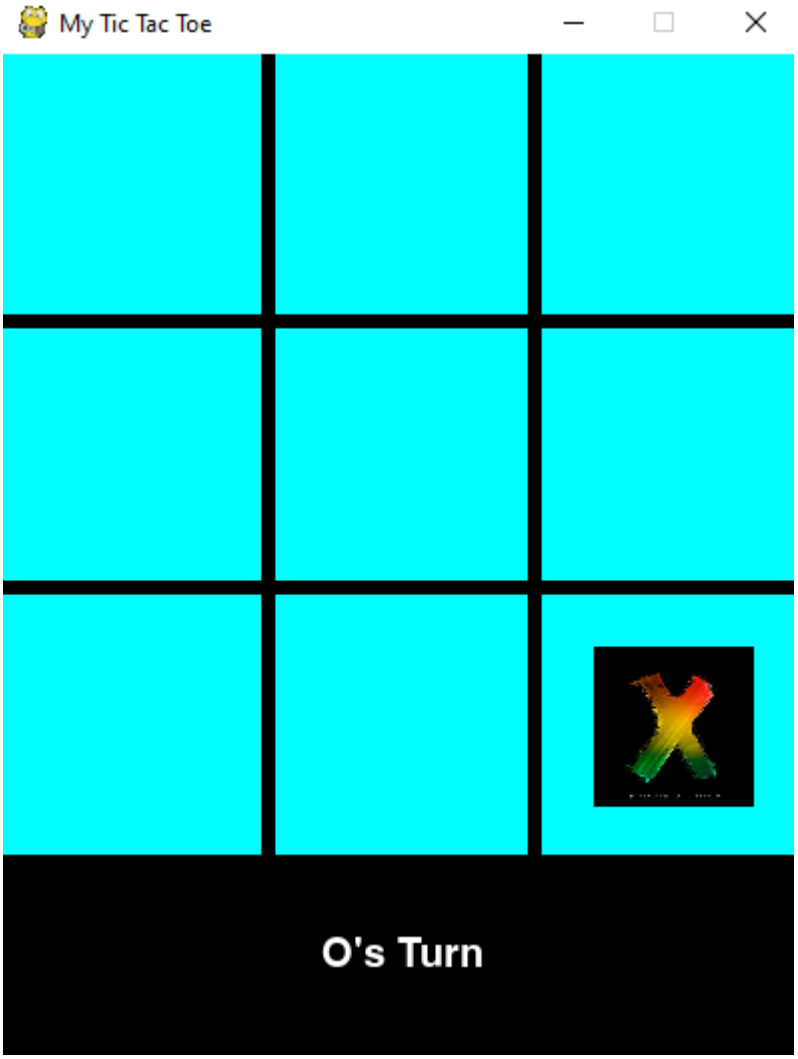
Material and Methods

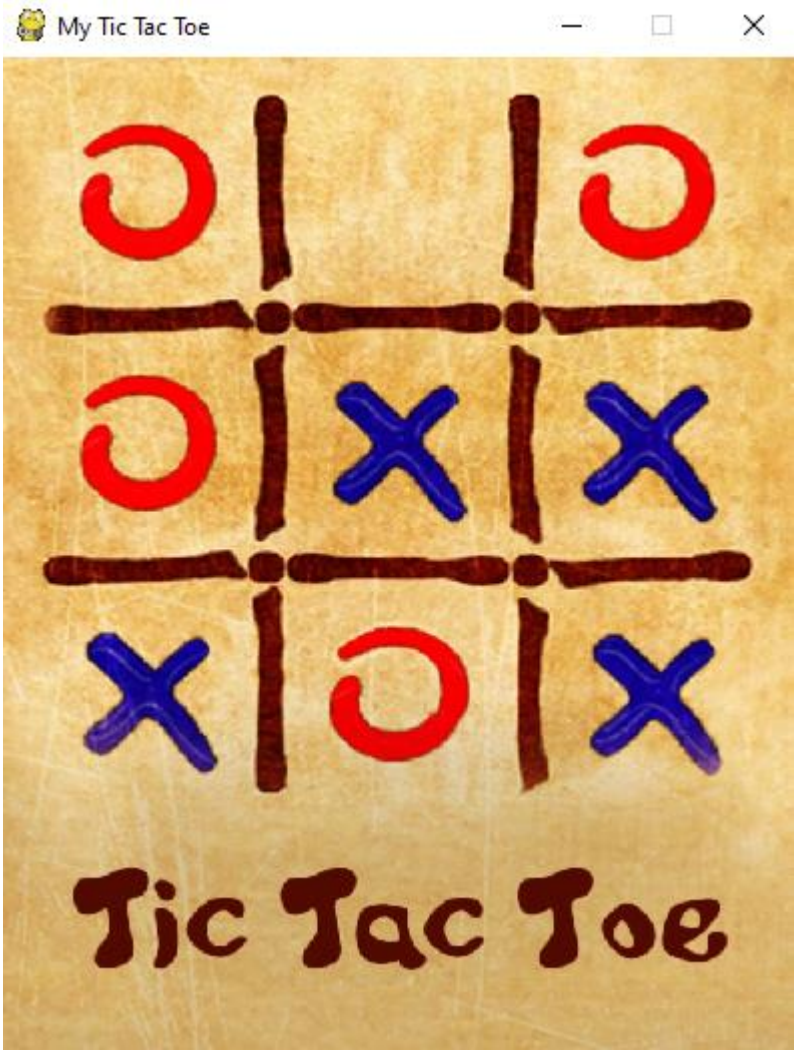
1. Pygame Library is used to make this game. Version of Pygame library I used is pygame-1.9.6.tar.gz
2. Thinker is used to make GUI.
3. Time and sys module are also used.
4. Mixer is used for music.

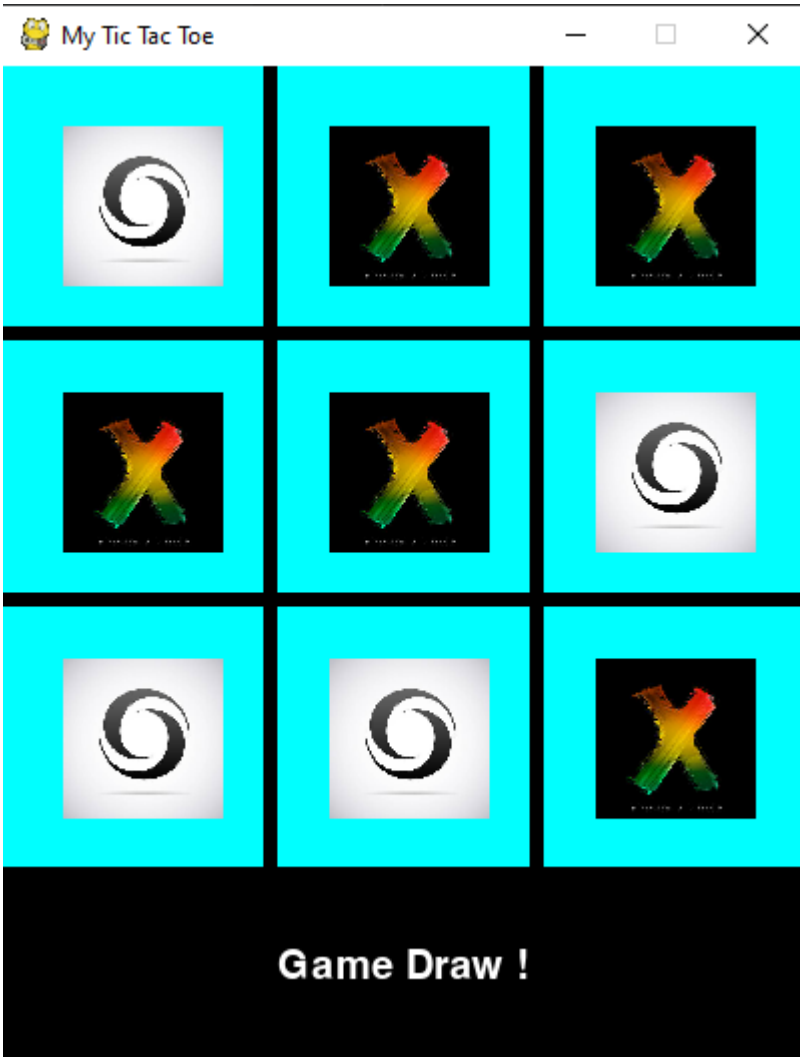
CONCLUSIONS

In the conclusion of this project, I would like to say that Python/Pygame is a fun and easy
And while creating a project like this, it has not just been a good experience but it also helped in the development of my creativity and logical thinking I would be more than happy to work on another project in python because It`s just amazing to work with python.

The programing is working and I hope, it's also bug-free


My Tic Tac Toe
O's Turn


My Tic Tac Toe
Tic Tac Toe


My Tic Tac Toe
Game Draw !