EXAMEN FINAL OPTATIVA I

PYTHON

DOCUMENTACION DEL CÓDIGO

NOMBRE: CLAUDIA DAHIANA GAONA MENA

DOCENTE: ING. RICARDO MAIDANA

SEMESTRE: 9NO

CAACUPÉ – PARAGUAY

2024

1. INTRODUCCION

1.1.Propósito del Programa

Cada jugador solo debe colocar su símbolo una vez por turno y no debe ser sobre una casilla ya jugada. En caso de que el jugador haga trampa el ganador será el otro. Se debe conseguir realizar una línea recta o diagonal por símbolo

1.2.Requisitos

Python 3.x

Juego Py

2. ESTRUCTURA DEL CÓDIGO

2.1.Importación de Bibliotecas

import pygame

import sys

Estas importaciones permiten utilizar funcionalidades de Pygame, manejo del sistema y almacenamiento de datos en archivos.

2.2.Inicialización y configuración

pygame.init()

fondo = pygame.image.load("fondo_juego.png")

circulo = pygame.image.load("circle.png")

equis = pygame.image.load("x.png")

Estas líneas inicializan Pygame y definen constantes para el tamaño de la pantalla y los fondos de pantalla.

2.3.Configuración de la Pantalla y Recursos

screen = pygame.display.set_mode((450, 450))

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

fondo = pygame.transform.scale(fondo, (450, 450))

circulo = pygame.transform.scale(circulo, (110, 100equis = pygame.transform.scale(equis, (110, 100)

fondo_inicio = pygame.image.load("fondo_inicio1.png")

fondo_inicio = pygame.transform.scale(fondo_inicio, (450, 450))

```
coor = [[(40, 50), (165, 50), (290, 50)],
        [(40, 175), (165, 175), (290, 175)],
        [(40, 300), (165, 300), (290, 300)]]
tablero = [['', '', ''],
           ['', '', ''],
           ['', '', '']]
turno = 'x'
game_over = False
clock = pygame.time.Clock()
```

Se configura la pantalla del juego y se establece el título de la ventana y la matriz para el diseño del tablero utilizada en el juego.

2.4. Variables de Estado del Juego

```
turno = 'x'
game_over = False
clock = pygame.time.Clock()
```

Se definen los diferentes estados del juego.

2.5. Cargar imágenes

```
fondo_inicio = pygame.image.load("fondo_inicio1.png")
fondo_inicio = pygame.transform.scale(fondo_inicio, (450, 450))
```

Se carga las imágenes de fondo para el menú principal y el dibujo para el tablero

2.6. Fuentes y Sonidos

```
pygame.mixer.music.load('musica.mp3')
pygame.mixer.music.play(-1)
```

Se cargarán las fuentes y el sonidos para el juego.

3. FUNCIONES

3.1. Funciones para mostrar ganador y mostrar turno

```
def mostrar_ganador(turno):
    font = pygame.font.Font(None, 74)
    nombre_ganador = nombre_jugador_x if turno == 'x' else nombre_jugador_o
    text = font.render(f"{nombre_ganador} ha ganado!", True, (255, 0, 0))
    screen.blit(text, (75, 200))
    pygame.display.flip()
    pygame.time.delay(2000)
def mostrar_turno():
    font = pygame.font.Font(None, 50)
    current_player = nombre_jugador_x if turno == 'x' else nombre_jugador_o
    text = font.render(f"Turno: {current_player}", True, (255, 255, 255))
    screen.blit(text, (10, 10))
```

Estas funciones permiten mostrar ganador y mostrar el turno de cuál de los jugadores le corresponde.

4. CLASES

4.1. Clase usuario

```
def input_names():
    global nombre_jugador_x, nombre_jugador_o
    input_active = False
    player = 'x'
    color_inactive = pygame.Color('lightskyblue3')
    color_active = pygame.Color('dodgerblue2')
    color = color_inactive
    font = pygame.font.Font(None, 50)
    input_box = pygame.Rect(100, 250, 250, 50)  # Ajustar posición más abajo
```

```
nombre_jugador_x = ""
nombre_jugador_o = ""


while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if input_box.collidepoint(event.pos):
                input_active = not input_active
            else:
                input_active = False
            color = color_active if input_active else color_inactive
        elif event.type == pygame.KEYDOWN:
            if input_active:
                if event.key == pygame.K_RETURN:
                    if player == 'x':
                        nombre_jugador_x = nombre_jugador_x.strip()
                        player = 'o'
                        input_box = pygame.Rect(100, 350, 250, 50)
                    else:
                        nombre_jugador_o = nombre_jugador_o.strip()
                        return
                elif event.key == pygame.K_BACKSPACE:
                    if player == 'x':
                        nombre_jugador_x = nombre_jugador_x[:-1]
```

```
                else:
                    nombre_jugador_o = nombre_jugador_o[:-1]
            else:
                if player == 'x':
                    nombre_jugador_x += event.unicode
                else:
                    nombre_jugador_o += event.unicode
    screen.blit(fondo_inicio, (0, 0))
    prompt = font.render("Nombre del jugador O:", True, (255, 255, 255)) if player == 'o' else font.render("Nombre del jugador X:", True, (255, 255, 255))
    screen.blit(prompt, (50, 150))
    if player == 'x':
        text_surface = font.render(nombre_jugador_x, True, color)
    else:
        text_surface = font.render(nombre_jugador_o, True, color)
    width = max(200, text_surface.get_width() + 10)
    input_box.w = width
    screen.blit(text_surface, (input_box.x + 5, input_box.y + 5))
    pygame.draw.rect(screen, color, input_box, 2)
    pygame.display.flip()
    clock.tick(30)
```

La clase usuario representa donde el jugador va ir ingresando su nombre

5. BUCLE PRINCIPAL DEL JUEGO

El bucle principal del juego se encarga de gestionar el estado del juego gestionar los eventos del usuario.

```
main_menu()
input_names()
reset_game()
```

```
while not game_over:
    clock.tick(30)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
        elif event.type == pygame.MOUSEBUTTONDOWN:
            mouseX, mouseY = event.pos
            if (40 <= mouseX < 415) and (50 <= mouseY < 425):
                fila = (mouseY - 50) // 125
                col = (mouseX - 40) // 125
                if tablero[fila][col] == '':
                    tablero[fila][col] = turno
                    if verificar_ganador():
                        print(f"El jugador {turno} ha ganado!!")
                        mostrar_ganador(turno)
                        game_over = True
                    turno = 'o' if turno == 'x' else 'x'

    graficar_board()
    pygame.display.update()

pygame.quit()
```

6. CONCLUSIÓN

El código del juego "Tic Tac Toe " es un juego simple pero eficaz para enseñar habilidades básicas de lógica y estrategia. Los jugadores aprenden a anticipar los movimientos del oponente y planificar sus próximos movimientos, proporcionando una experiencia de juego fluida y reactiva.

ANEXO


Tic Tac Toe
Tic Tac Toe
Jugar
Salir


Tic Tac Toe
Tic Tac Toe
Nombre del jugador X:


Tic Tac Toe
Turno: clau
Tic Tac Toe


Tic Tac Toe
Turno: clau
clau ha ganado
Tic Tac Toe