Universidad Autónoma del Estado de Baja California

Facultad de ingeniería, Arquitectura y diseño

MEMORAMA

“HY-THINK”

Integrantes:

Miguel Angel Portillo Attwell

Martin Santos Tirado

Grupo 432

Ingeniería en computación

Ensenada, Baja California; a 1 de diciembre de 2023

```
import pygame

import random

import time

# Inicializar Pygame

pygame.init()

# Definir colores

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

RED = (255, 0, 0)

AZUL_VIOLETA = (138,43,226)

NARANJA = (255,145,77)

MORADO = (94,23,235)

# Configuración de la pantalla

WIDTH, HEIGHT = 800, 600

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

pygame.display.set_caption('Memorama')

# Definir estados del juego

MENU_PRINCIPAL = 0

EN_JUEGO = 1

# Pila de estados para manejar las pantallas del juego

estado_pila = []

# Definir estado inicial

estado = MENU_PRINCIPAL

MENU = pygame.image.load("menus/main.png")
```

```
JUGAR = pygame.image.load("menus/juego2.png")

OPCION = pygame.image.load("menus/opcion.png")

CREDITOS = pygame.image.load("menus/creditos.png")

RN =pygame.image.load ("img/oculto.png")

PLAY = pygame.image.load ("botones/jugar_btn.png")

SALIR = pygame.image.load ("botones/salir_btn.png")

CREDITOS2 = pygame.image.load ("botones/creditos_btn.png")

REGRESAR = pygame.image.load ("botones/regreso_btn.png")

REINICIAR = pygame.image.load ("botones/reiniciar_btn.png")

PANTALLA_GANADOR = pygame.image.load("menus/ganaste.png")

# Tamaño de las cartas

CARD_WIDTH, CARD_HEIGHT = 101, 101

# Función para mostrar texto en pantalla

def text_objects(text, font):

    text_surface = font.render(text, True, BLACK)

    return text_surface, text_surface.get_rect()

#Clase para la carta

class Carta:

    def __init__(self, imagen=None):

        self.descubierta = False

        self.imagen = imagen

# Funcion para generar el tablero

def generar_tablero(fila, columna):

    rutas_imagenes = [
```

```
        'img/luigui.png',
        'img/mario.png',
        'img/flor_azul.png',
        'img/flor_roja.png',
        'img/cap_ama.png',
        'img/cap_azul.png',
        'img/cap_rojo.png',
        'img/cap_verde.png'
    ]
    rutas_imagenes *= 2  # Duplicar las rutas para formar parejas
    random.shuffle(rutas_imagenes)  # Mezclar las rutas de las imágenes

    tablero = []
    valor_carta = 0  # Identificador único para cada carta
    for _ in range(fila):
        fila_cartas = []
        for _ in range(columna):
            ruta_imagen_carta = rutas_imagenes.pop()
            carta = Carta(imagen=ruta_imagen_carta)
            carta.valor = valor_carta  # Asignar un valor único a cada
carta
            valor_carta += 1  # Incrementar el valor para la próxima
carta
            fila_cartas.append(carta)
        tablero.append(fila_cartas)
```

```
    return tablero

# Función para crear botones

def button(image, x, y, w, h, action=None):

    mouse = pygame.mouse.get_pos()

    click = pygame.mouse.get_pressed()



    if x + w > mouse[0] > x and y + h > mouse[1] > y:

        # Dibujar el botón activo (por ejemplo, con un cambio de color
o resaltado)

        screen.blit(pygame.transform.scale(image, (w, h)), (x, y))

        if click[0] == 1 and action is not None:

            action()

    else:

        # Dibujar el botón inactivo

        screen.blit(pygame.transform.scale(image, (w, h)), (x, y))

# Funcion para regresar al menu

def regresar_menu():

    global estado

    estado = MENU_PRINCIPAL

    limpiar_pantalla()  # Limpia la pantalla al regresar al menú
principal

    game_intro()  # Llama a la función game_intro para mostrar solo el
menú principal

# Funcion para reinicial el juego
```

```
def reiniciar_juego():
    global estado
    estado = EN_JUEGO
    limpiar_pantalla()  # Limpia la pantalla
    memorama_game()
# Limpiar pantalla
def limpiar_pantalla():
    screen.fill(BLACK)  # Rellena la pantalla con el color blanco o el
color deseado
# Funcion para iniciar el juego
def jugar():
    global estado
    estado = EN_JUEGO
    memorama_game()
# Función para el juego de memorama
def memorama_game():
    global estado
    fila = 4
    columna = 4
    tablero = generar_tablero(fila, columna)
    contador_intentos = 0  # Inicializar contador de intentos
    gano = False
    tiempo_texto_ganador = None  # Variable para controlar el tiempo
    cartas_volteadas = []  # Inicializar lista de cartas volteadas
```

```
    while True:
        for evento in pygame.event.get():
            if evento.type == pygame.QUIT:
                pygame.quit()
                quit()
            if estado == EN_JUEGO:
                if evento.type == pygame.MOUSEBUTTONDOWN:
                    mouse_x, mouse_y = pygame.mouse.get_pos()
                    fila = (mouse_y - 50) // 120
                    columna = (mouse_x - 50) // 120
                    if 0 <= fila < len(tablero) and 0 <= columna <
len(tablero[0]):
                        carta_seleccionada = tablero[fila][columna]
                        if not carta_seleccionada.descubierta and
len(cartas_volteadas) < 2:
                            carta_seleccionada.descubierta = True
                            cartas_volteadas.append((fila, columna))
                        if len(cartas_volteadas) == 2:
                            contador_intentos += 1
        screen.blit(JUGAR, (0, 0))
        if estado == EN_JUEGO:
            display_board(tablero)
            button(REGRESAR, 600, 500, 170, 50,regresar_menu)
            button(REINICIAR, 600, 420, 170, 50,reiniciar_juego)
```

```
            # Mostrar el contador de intentos en pantalla
            font = pygame.font.Font(None, 36)
            text = font.render(f"Intentos: {contador_intentos}", True,
MORADO)
            screen.blit(text, (620, 330))  # Posición del texto en
pantalla


            todas_descubiertas = all(all(carta.descubierta for carta in
fila) for fila in tablero)
            if todas_descubiertas and not gano:
                gano = True
            if gano:
                # Obtener el rectángulo de la imagen del ganador
                ganador_rect = PANTALLA_GANADOR.get_rect()


                # Centrar la imagen del ganador en la pantalla del
juego
                ganador_rect.center = (WIDTH // 2, HEIGHT // 2)


                # Dibujar la imagen del ganador centrada
                screen.blit(PANTALLA_GANADOR, ganador_rect)
                # Mostrar el total de intentos en la imagen
                font_intentos = pygame.font.Font(None, 30)
                text_intentos = font_intentos.render(f"Total de
intentos:{contador_intentos}", True, NARANJA)
                screen.blit(text_intentos, (300, 270))
```

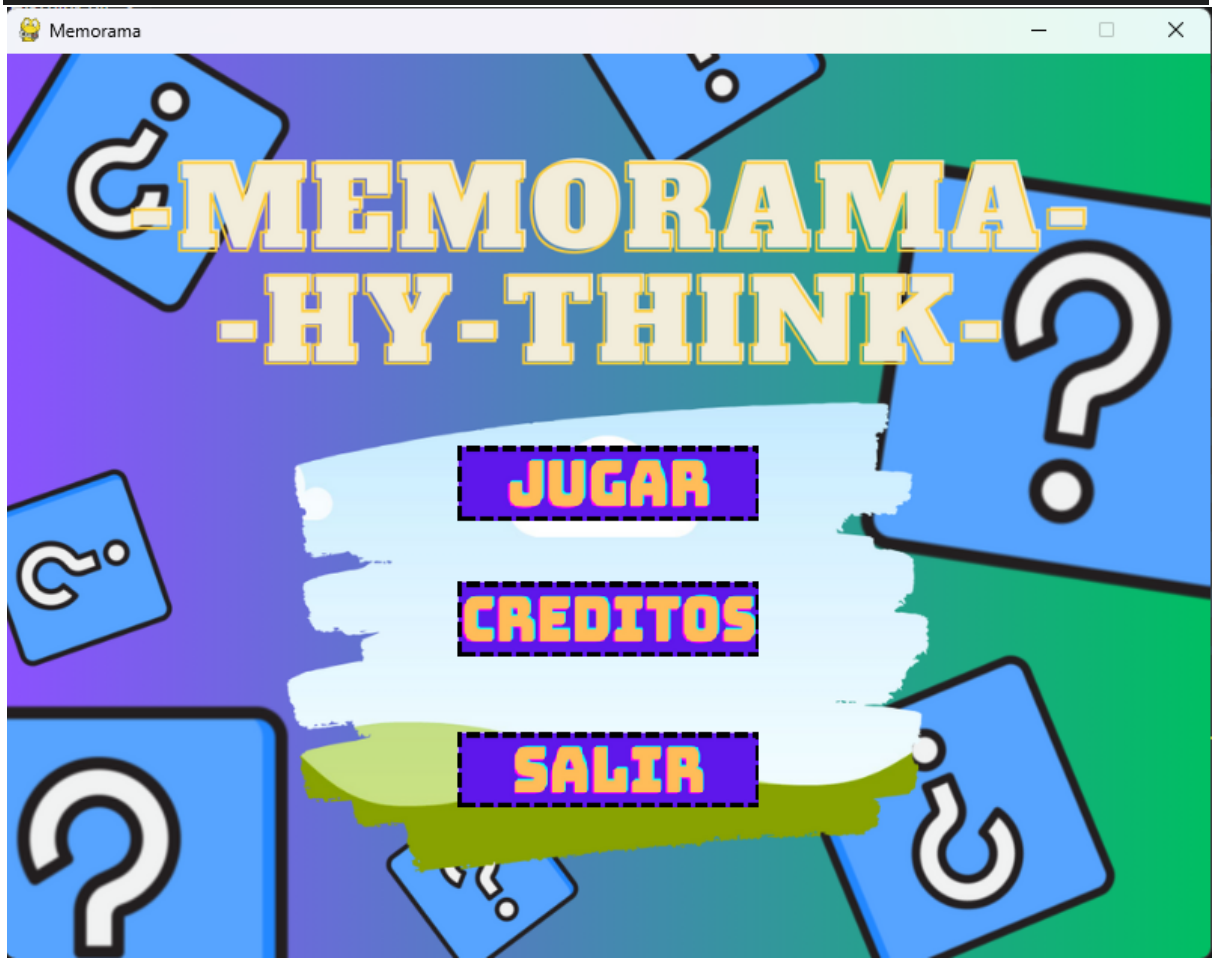
```
                # Mostrar botones en la imagen
                button(REINICIAR, 300, 330, 200, 50, reiniciar_juego)
                button(REGRESAR, 300, 410, 200, 50, regresar_menu)

                # Actualizar la pantalla completa
                pygame.display.flip()  # Muestra el mensaje de victoria en pantalla

            if len(cartas_volteadas) == 2:
                pygame.display.flip()
                time.sleep(.2)  # Pequeño retraso para mostrar las cartas

                # Obtener las imágenes de las dos cartas seleccionadas
                imagen_carta1 = tablero[cartas_volteadas[0][0]][cartas_volteadas[0][1]].imagen
                imagen_carta2 = tablero[cartas_volteadas[1][0]][cartas_volteadas[1][1]].imagen

                if imagen_carta1 == imagen_carta2:
                    # Si las imágenes son iguales, déjalas descubiertas
                    for fila, columna in cartas_volteadas:
                        tablero[fila][columna].descubierta = True
                    cartas_volteadas = []  # Reinicia la lista de cartas volteadas
```

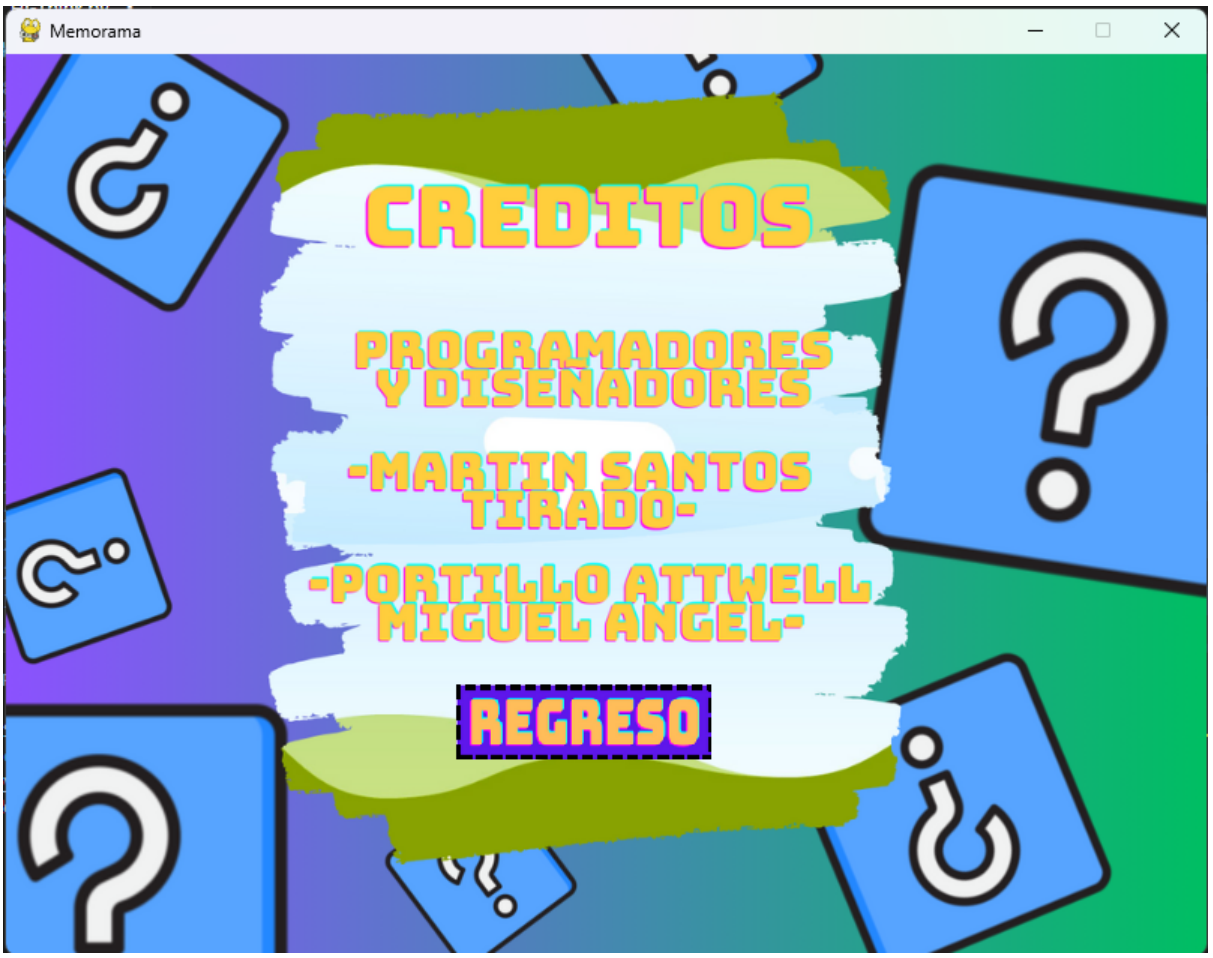
```
                else:
                    # Si no son iguales, voltea ambas cartas nuevamente después de un pequeño retraso
                    pygame.display.flip()
                    time.sleep(1)
                    for fila, columna in cartas_volteadas:
                        tablero[fila][columna].descubierta = False
                    cartas_volteadas = []  # Reinicia la lista de cartas volteadas
        pygame.display.flip()
# Función para mostrar el tablero con las cartas
def display_board(tablero):
    screen.blit(JUGAR, (0, 0))
    fila = 4
    columna = 4
    x, y = 50, 50  # Coordenadas iniciales
    gap = 20  # Espacio entre las cartas en píxeles

    for row in range(fila):
        for col in range(columna):
            carta_actual = tablero[row][col]
            if not carta_actual.descubierta:
                screen.blit(RN, (x, y))
            else:
```

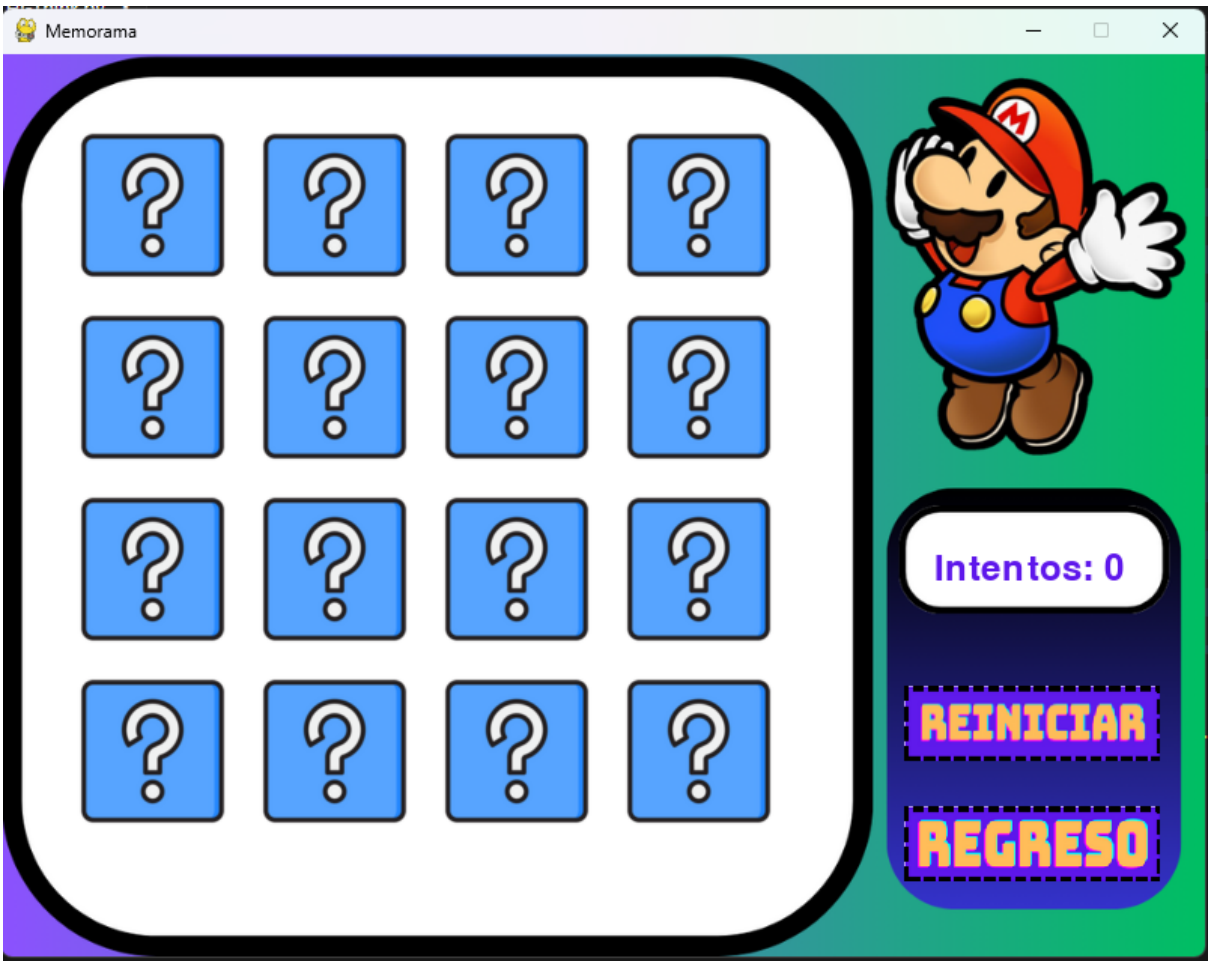
```
screen.blit(pygame.transform.scale(pygame.image.load(carta_actual.image
n), (CARD_WIDTH, CARD_HEIGHT)), (x, y))

            x += CARD_WIDTH + gap

        y += CARD_HEIGHT + gap

        x = 50

# Funcion para los creditos

def creditos():

    global estado

    estado_pila.append(estado)

    estado = "CREDITOS"

    while estado == "CREDITOS":

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                quit()

            screen.blit(CREDITOS, (0, 0))

            button(REGRESAR, 300, 420, 170, 50, regresar_creditos)

        pygame.display.update()

# Funcion para regresar al menu

def regresar_creditos():

    global estado

    limpiar_pantalla()  # Limpia la pantalla al regresar al menú
principal

    game_intro()
```

```
# Código a ejecutar cuando se hace clic en el botón
def mi_accion():
    pass
# Función para el menú principal
def game_intro():
    global estado
    estado = MENU_PRINCIPAL
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()

        screen.blit(MENU, (0, 0))

        button(PLAY, 300, 260, 200, 50, jugar)
        button(CREDITOS2, 300, 350, 200, 50, creditos)
        button(SALIR, 300, 450, 200, 50, pygame.quit)

        pygame.display.update()
# Funcion para jugar al juego
def jugar():
    global estado
    estado = EN_JUEGO
```

```
    memorama_game()

# Iniciar el menú principal

game_intro()
```




Memorama
CREDITOS
PROGRAMADORES
Y DISEÑADORES
-MARTIN SANTOS
TIRADO-
-PORTILLO ATTWELL
MIGUEL ANGEL-
REGRESO


Memorama
Intentos: 0
REINICIAR
REGRESO


Memorama
GANASTE
Total de intentos:15
REINICIAR
REGRESO
Intentos: 15
REINICIAR
REGRESO