

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO



MATERIA:
LENGUAJE DE PROGRAMACION PYTHON

PROFESOR:
PEDROS NUÑES YEPIZ

ALUMNO:
MIGUEL ANGEL PORTILLO ATTWELL
JOSE GAEI RODRIGUEZ RUIZ

GRUPO:
432

FECHA:
07 DE JUNIO DEL 2023

CODIGO – MEMORAMA

```
import pygame
import sys
import math
import time
import random

pygame.init()
pygame.font.init()
pygame.mixer.init()

altura_boton = 30 # El botón de abajo, para iniciar juego
medida_cuadro = 200 # Medida de la imagen en pixeles
# La parte trasera de cada tarjeta
nombre_imagen_oculta = ("img/oculto.png")
imagen_oculta = pygame.image.load(nombre_imagen_oculta)
segundos_mostrar_pieza = 2 # Segundos para ocultar la pieza si no es la correcta
imagen_fondo = pygame.image.load("img/fondo.png")

class Cuadro:
    def __init__(self, fuente_imagen):
        self.mostrar = True
        self.descubierto = False
        self.fuente_imagen = fuente_imagen
        self.imagen_real = pygame.image.load(fuente_imagen)

cuadros = [
    [Cuadro("img/kuupa_rojo.png"), Cuadro("img/kuupa_rojo.png"),
     Cuadro("img/kuupa_amarillo.png"), Cuadro("img/kuupa_amarillo.png")],
    [Cuadro("img/kuupa_verde.png"), Cuadro("img/kuupa_verde.png"),
     Cuadro("img/kuupa_azul.png"), Cuadro("img/kuupa_azul.png")],
    [Cuadro("img/flor_roja.png"), Cuadro("img/flor_roja.png"),
     Cuadro("img/flor_azul.png"), Cuadro("img/flor_azul.png")],
    [Cuadro("img/luigi.png"), Cuadro("img/luigi.png"),
     Cuadro("img/mario.png"), Cuadro("img/mario.png")],
]

# Colores
color_blanco = (255, 255, 255)
color_negro = (0, 0, 0)
color_gris = (206, 206, 206)
color_azul = (30, 136, 229)

# Los sonidos
sonido_fondo = pygame.mixer.Sound("sound/fondo.wav")
sonido_clic = pygame.mixer.Sound("sound/clic.wav")
sonido_exito = pygame.mixer.Sound("sound/ganador.wav")
sonido_fracaso = pygame.mixer.Sound("sound/incorrecta.wav")
sonido_voltar = pygame.mixer.Sound("sound/voltar.wav")
```

```
# Calculamos el tamaño de la pantalla en base al tamaño de los cuadrados
anchura_pantalla = 1200
altura_pantalla = 900
anchura_boton = anchura_pantalla

# La fuente que estará sobre el botón
tamanio_fuente = 20
fuente = pygame.font.SysFont("Arial", tamanio_fuente)
xFuente = int((anchura_boton / 2) - (tamanio_fuente / 2))
yFuente = int(altura_pantalla - altura_boton)

# El botón, que al final es un rectángulo
boton = pygame.Rect(0, altura_pantalla - altura_boton,
                    anchura_boton, altura_pantalla)

ultimos_segundos = None
puede_jugar = True
juego_iniciado = False

x1 = None
y1 = None
x2 = None
y2 = None

def ocultar_todos_los_cuadros():
    for fila in cuadros:
        for cuadro in fila:
            cuadro.mostrar = False
            cuadro.descubierto = False

def aleatorizar_cuadros():
    # Elegir X e Y aleatorios, intercambiar
    cantidad_filas = len(cuadros)
    cantidad_columnas = len(cuadros[0])
    for y in range(cantidad_filas):
        for x in range(cantidad_columnas):
            x_aleatorio = random.randint(0, cantidad_columnas - 1)
            y_aleatorio = random.randint(0, cantidad_filas - 1)
            cuadro_temporal = cuadros[y][x]
            cuadros[y][x] = cuadros[y_aleatorio][x_aleatorio]
            cuadros[y_aleatorio][x_aleatorio] = cuadro_temporal

def comprobar_si_gana():
    if gana():
        pygame.mixer.Sound.play(sonido_exito)
        reiniciar_juego()
```

```

# Regresa False si al menos un cuadro NO está descubierto. True en caso de que
absolutamente todos estén descubiertos
def gana():
    for fila in cuadros:
        for cuadro in fila:
            if not cuadro.descubierto:
                return False
    return True

def reiniciar_juego():
    global juego_iniciado
    juego_iniciado = False

def iniciar_juego():
    pygame.mixer.Sound.play(sonido_clic)

    global juego_iniciado
    # Aleatorizar 3 veces
    for i in range(3):
        aleatorizar_cuadros()
    ocultar_todos_los_cuadros()
    juego_iniciado = True

pantalla_juego = pygame.transform.scale(imagen_fondo, (anchura_pantalla,
altura_pantalla))
pantalla_juego = pygame.display.set_mode((anchura_pantalla, altura_pantalla))
pygame.display.set_caption('Memorama - By Miguel Portillo/Gael Rodriguez')
pygame.mixer.Sound.play(sonido_fondo, -1) # El -1 indica un loop infinito

# Ciclo infinito...
while True:
    # Escuchar eventos, pues estamos en un ciclo infinito que se repite varias
    veces por segundo
    for event in pygame.event.get():
        # Si quitan el juego, salimos
        if event.type == pygame.QUIT:
            sys.exit()
        # Si hicieron clic y el usuario puede jugar...
        elif event.type == pygame.MOUSEBUTTONDOWN and puede_jugar:
            # Si el click fue sobre el botón y el juego no se ha iniciado,
            entonces iniciamos el juego
            xAbsoluto, yAbsoluto = event.pos
            if boton.collidepoint(event.pos):
                if not juego_iniciado:
                    iniciar_juego()

        else:
            # Si no hay juego iniciado, ignoramos el clic
            if not juego_iniciado:

```

```

        continue
    x = math.floor(xAbsoluto / medida_cuadro)
    y = math.floor(yAbsoluto / medida_cuadro)
    # Primero lo primero. Si ya está mostrada o descubierta, no
hacemos nada

    cuadro = cuadros[y][x]
    if cuadro.mostrar or cuadro.descubierto:
        # continue ignora lo de abajo y deja que el ciclo siga
        continue
    # Si es la primera vez que tocan la imagen (es decir, no están
buscando el par de otra, sino apenas
    # están descubriendo la primera)
    if x1 is None and y1 is None:
        # Entonces la actual es en la que acaban de dar clic, la
mostramos

        x1 = x
        y1 = y
        cuadros[y1][x1].mostrar = True
        pygame.mixer.Sound.play(sonido_voltear)
    else:
        # En caso de que ya hubiera una clic anteriormente y estemos
buscando el par, comparamos lo siguiente
        x2 = x
        y2 = y
        cuadros[y2][x2].mostrar = True
        cuadro1 = cuadros[y1][x1]
        cuadro2 = cuadros[y2][x2]
        # Si coinciden, entonces a ambas las ponemos en descubiertas:
        if cuadro1.fuente_imagen == cuadro2.fuente_imagen:
            cuadros[y1][x1].descubierto = True
            cuadros[y2][x2].descubierto = True
            x1 = None
            x2 = None
            y1 = None
            y2 = None
            pygame.mixer.Sound.play(sonido_clic)
        else:
            pygame.mixer.Sound.play(sonido_fracaso)

            ultimos_segundos = int(time.time())
            puede_jugar = False
            comprobar_si_gana()

    ahora = int(time.time())
    if ultimos_segundos is not None and ahora - ultimos_segundos >=
segundos_mostrar_pieza:
        cuadros[y1][x1].mostrar = False
        cuadros[y2][x2].mostrar = False
        x1 = None
        y1 = None
        x2 = None
        y2 = None

```

```

    ultimos_segundos = None
    # En este momento el usuario ya puede hacer clic de nuevo pues las
    imágenes ya estarán ocultas
    puede_jugar = True

    pantalla_juego.blit(imagen_fondo, (0, 0))
    x = 0
    y = 0
    # Recorrer los cuadros
    for fila in cuadros:
        x = 0
        for cuadro in fila:
            if cuadro.descubierto or cuadro.mostrar:
                pantalla_juego.blit(cuadro.imagen_real, (x, y))
            else:
                pantalla_juego.blit(imagen_oculta, (x, y))
            x += medida_cuadro
        y += medida_cuadro

    # También dibujamos el botón
    if juego_iniciado:
        # Si está iniciado, entonces botón blanco con fuente gris para que
        parezca deshabilitado
        pygame.draw.rect(pantalla_juego, color_blanco, boton)
        pantalla_juego.blit(fuente.render(
            "Iniciar juego", True, color_gris), (xFuente, yFuente))
    else:
        pygame.draw.rect(pantalla_juego, color_azul, boton)
        pantalla_juego.blit(fuente.render(
            "Iniciar juego", True, color_blanco), (xFuente, yFuente))

    # Actualizamos la pantalla
    pygame.display.update()

```

