



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en computación

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Tony Ozuna Ceseña, Miguel Angel Monjaraz Ramirez

Matrícula: 372270, 372205

Maestro: Pedro Núñez Yépiz

Actividad No.: Trabajo final

Tema - Unidad :

Ensenada Baja California a 7 de Diciembre del 2023



Universidad Autónoma de Baja California

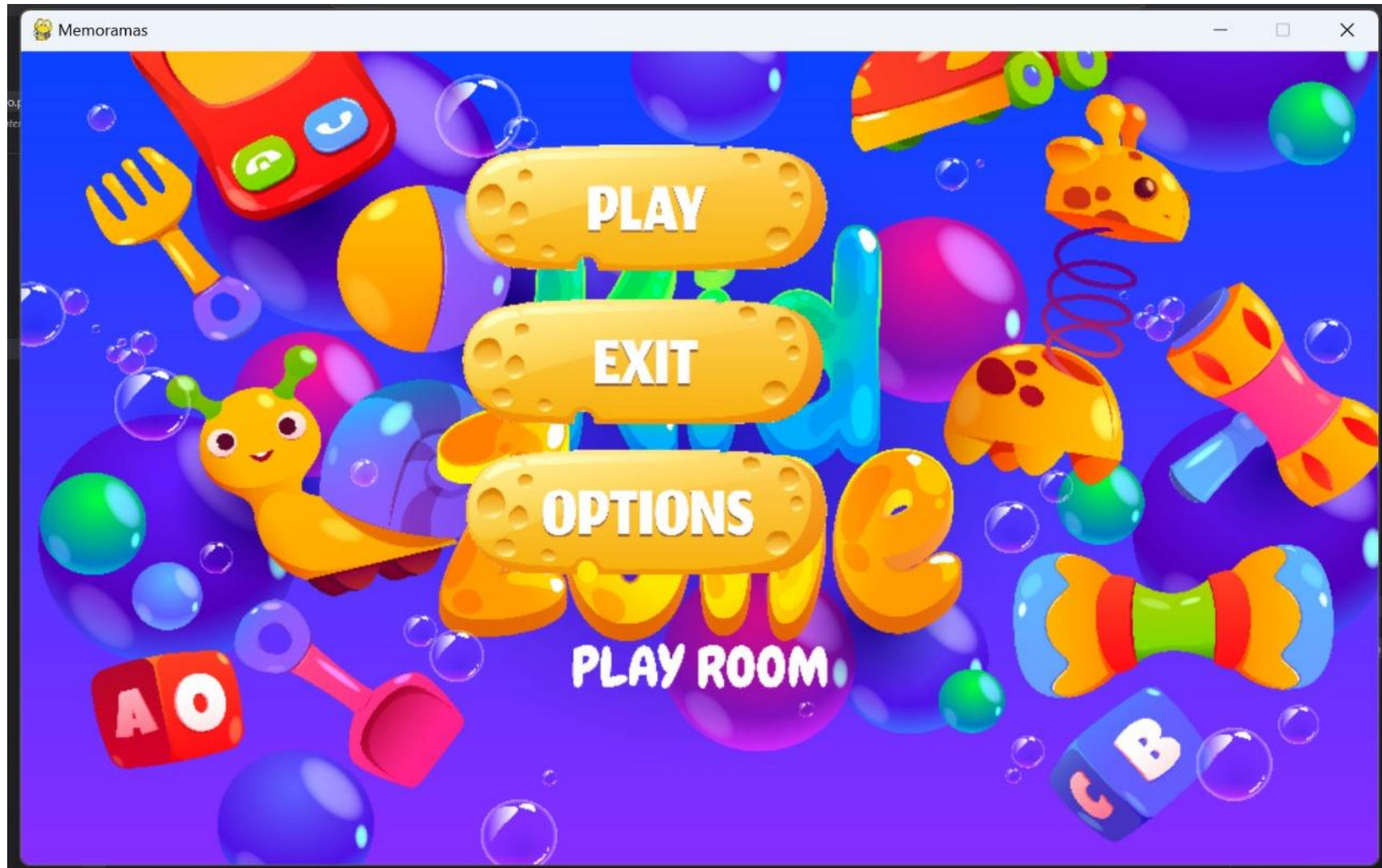
Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
python
pneubaby
p.py
FOC_calculadora_Act_13.py
CalculadoraBasica.py
juego.py
juego2.py

juego > juego > p.py > ...
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201

class:
    self.x2 = x
    self.y2 = y
    self.cuadros[self.y2][self.x2].mostrar = True
    pygame.mixer.Sound.play(sonido_voltear)

    cuadro1 = self.cuadros[self.y1][self.x1]
    cuadro2 = self.cuadros[self.y2][self.x2]

    # Verificar si las elecciones coinciden
    if cuadro1.identificador == cuadro2.identificador and cuadro1.es_operacion != cuadro2.es_operacion:
        self.cuadros[self.y1][self.x1].descubierto = True
        self.cuadros[self.y2][self.x2].descubierto = True
        pygame.mixer.Sound.play(sonido_correcto)
        self.x1, self.x2, self.y1, self.y2 = None, None, None, None
    else:
        self.ultimos_segundos = int(time.time())
        pygame.mixer.Sound.play(sonido_fallo)
        self.puede_jugar = False
    if self.gana():
        pygame.mixer.Sound.play(sonido_completado)

    self.comprobar_si_gana()

ahora = int(time.time())
if self.ultimos_segundos is not None and ahora - self.ultimos_segundos >= self.segundos mostrar.pieza:
    self.cuadros[self.y1][self.x1].mostrar = False
    self.cuadros[self.y2][self.x2].mostrar = False
    self.x1, self.y1, self.x2, self.y2 = None, None, None, None
    self.ultimos_segundos = None
    self.puede_jugar = True

# Dibujar el fondo antes de dibujar las imágenes de los cuadros
# pantalla_juego.blit(fondo, (0, 0))
pantalla_juego.blit(fondo, (0, 0))
self.boton.dibujar(pantalla_juego)
x, y = 300, 70
for fila in self.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(self.imagen_oculta, (x, y))
        x += self.medida_cuadro
    y += self.medida_cuadro

if self.juego_iniciado:
    pygame.draw.rect(pantalla_juego, self.color_blanco, self.boton)
    pantalla_juego.blit(self.fuente.render("INICIAR JUEGO", True, self.color_gris),
                        (self.xFuente, self.yFuente))
else:
    pygame.draw.rect(pantalla_juego, self.color_gris, self.boton)
    pantalla_juego.blit(self.fuente.render("INICIAR JUEGO", True, self.color_blanco),
                        (self.xFuente, self.yFuente))
pygame.display.update()

if __name__ == "__main__":
    juego = JuegoMemorama()
    juego.ejecutar()
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
juego > juego > .p.py > ...
85 def comprobar_si_gana(self):
86     if self.gana():
87         self.reiniciar_juego()
88
89 def gana(self):
90     for fila in self.cuadros:
91         for cuadro in fila:
92             if not cuadro.descubierto:
93                 return False
94     return True
95
96 def reiniciar_juego(self):
97     self.juego_iniciado = False
98
99 def iniciar_juego(self):
100     for _ in range(3):
101         self.aleatorizar_cuadros()
102     self.ocultar()
103     self.juego_iniciado = True
104
105 def ejecutar(self):
106     pantalla_juego = pygame.display.set_mode((self.anchura_pantalla, self.altura_pantalla))
107     width, height = 750, 500
108     pantalla_juego = pygame.display.set_mode((width, height))
109     pygame.display.set_caption("Memorama")
110     fondo = pygame.Surface(pantalla_juego.get_size())
111     fondo.fill((255, 255, 255)) # Rellena el fondo con el color blanco
112
113     sonido_voltar = pygame.mixer.Sound("musica/assets/voltar.wav")
114     sonido_correcto = pygame.mixer.Sound("musica/correct.mp3")
115     sonido_completado = pygame.mixer.Sound("musica/complet.mp3")
116     sonido_fallo = pygame.mixer.Sound("musica/fallo.mp3")
117
118     while not self.salir:
119         for event in pygame.event.get():
120             if event.type == pygame.QUIT:
121                 self.salir = True
122             elif event.type == pygame.MOUSEBUTTONDOWN and self.puede_jugar:
123                 xAbsoluto, yAbsoluto = event.pos
124                 if self.boton.colidepoint(event.pos):
125                     if not self.juego_iniciado:
126                         self.iniciar_juego()
127                     else:
128                         if not self.juego_iniciado:
129                             continue
130                         x = math.floor(xAbsoluto / self.medida_cuadro)
131                         y = math.floor(yAbsoluto / self.medida_cuadro)
132                         cuadro = self.cuadros[y][x]
133                         if cuadro.mostrar or cuadro.descubierto:
134                             continue
135                         if self.x1 is None and self.y1 is None:
136                             self.x1 = x
137                             self.y1 = y
138                         self.cuadros[self.y1][self.x1].mostrar = True
139                         pygame.mixer.Sound.play(sonido_voltar)
140                     else:
141                         self.x2 = x
142                         self.y2 = y
143                         self.cuadros[self.y2][self.x2].mostrar = True
144                         pygame.mixer.Sound.play(sonido_voltar)
145
146                 cuadro1 = self.cuadros[self.y1][self.x1]
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
pruebapy x IOC_calculadora_Act_13.py Calculadorabasic.py juego.py juego2.py
juego > juego > p.py > ~
1 import pygame
2 import sys
3 import math
4 import time
5 import random
6
7 from boton import Imagen
8
9 class JuegoMemorama:
10     def __init__(self):
11         pygame.init()
12         pygame.font.init()
13         pygame.mixer.init()
14
15         # Variables y configuraciones
16         self.altura_boton = 40
17         self.medida_cuadro = 100
18         self.nombre_imagen_oculta = "sumayresta/ATRAS.png"
19         self.imagen_oculta = pygame.image.load(self.nombre_imagen_oculta)
20         self.segundos_mostrar_pieza = 2
21
22         self.cuadros = [
23             [self.Cuadro(1, "sumayresta/op1.png", True), self.Cuadro(1, "sumayresta/resul1.png", False),
24              self.Cuadro(2, "sumayresta/op2.png", True), self.Cuadro(2, "sumayresta/resul2.png", False)],
25
26             [self.Cuadro(3, "sumayresta/op3.png", True), self.Cuadro(3, "sumayresta/resul3.png", False),
27              self.Cuadro(4, "sumayresta/op4.png", True), self.Cuadro(4, "sumayresta/resul4.png", False)],
28         ]
29
30         self.color_blanco = (255, 255, 255)
31         self.color_negro = (0, 0, 0)
32         self.color_gris = (255, 0, 255)
33         self.color_azul = (30, 136, 229)
34
35         self.anchura_pantalla = len(self.cuadros[0]) * self.medida_cuadro
36         self.altura_pantalla = (len(self.cuadros) * self.medida_cuadro) + self.altura_boton
37         self.anchura_boton = 800
38
39         self.tamano_fuente = 40
40         self.fuente = pygame.font.SysFont("Arial", self.tamano_fuente)
41         self.xFuente = int((self.anchura_boton / 2) - (self.tamano_fuente / 2))
42         self.yFuente = int(self.altura_pantalla - self.altura_boton)
43         self.play = Imagen(300, 40, 100, 50, "interfaz/play.png", "interfaz/play.png", 300, 150)
44         self.boton = pygame.Rect(0, self.altura_pantalla - self.altura_boton,
45                                  self.anchura_boton, self.altura_pantalla)
46
47 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
48 PS U:\Python> & C:\Users\Tony\AppData\Local\Programs\Python\Python312\python.exe u:/Python/juego/juego/p.py
49 pygame 2.5.2 (SDL 2.28.1, Python 3.12.0)
50 Hello from the pygame community. https://www.pygame.org/contribute.html
51 Traceback (most recent call last):
52   File "u:/Python/juego/juego/p.py", line 7, in <module>
53     from boton import Imagen
54   File "u:/Python/juego/juego/boton.py", line 2, in <module>
55     import imageio
56 ModuleNotFoundError: No module named 'imageio'
57 PS U:\Python>
```