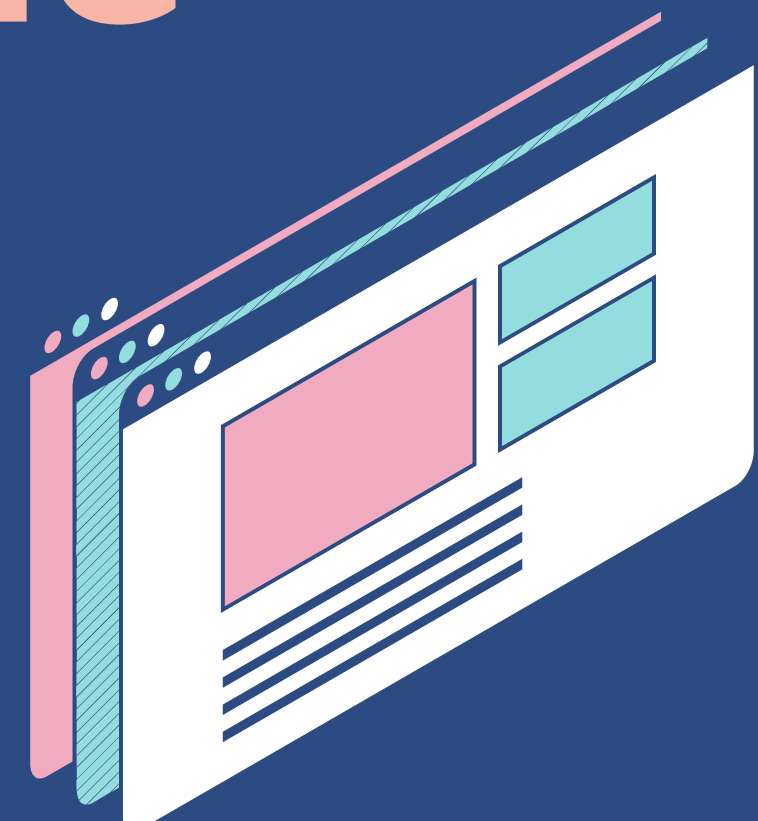


PRESENTACIÓN DEL JUEGO

PROYECTO ATARI PONG

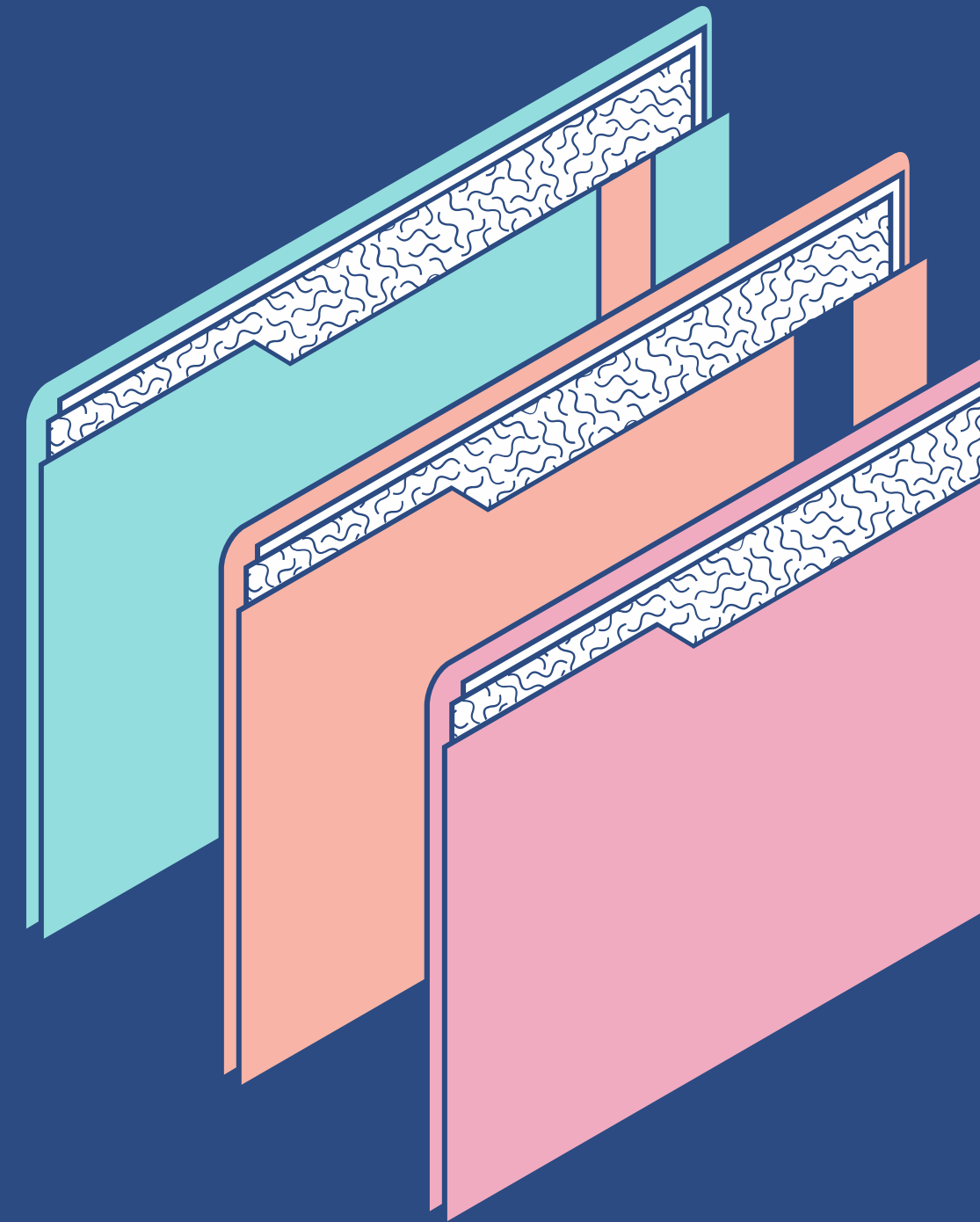
Integrantes:

- Jimmy Brito
- Luis Lutuala



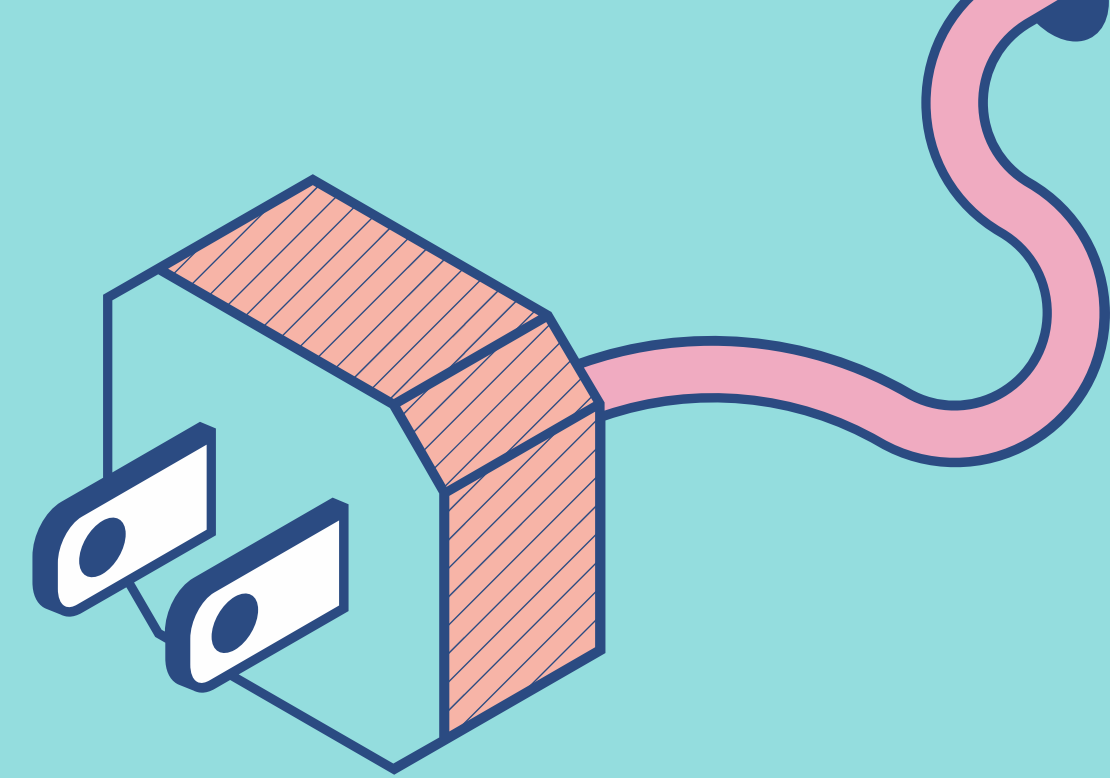
Configuración Inicial

```
1  import pygame
2  import sys
3
4  pygame.init()
5
6  # Aquí definimos los colores que se utilizarán en el juego
7  WHITE = (255, 255, 255)
8  BLACK = (0, 0, 0)
9  GRAY = (150, 150, 150)
10
11 # Las dimensiones y el nombre de la ventana
12 WIDTH, HEIGHT = 800, 600
13 screen = pygame.display.set_mode((WIDTH, HEIGHT))
14 pygame.display.set_caption('Atari Pong')
15 |
```



Primero, importamos las librerías necesarias, inicializamos Pygame y definimos los colores que utilizaremos en el juego. Luego, configuramos las dimensiones de la ventana del juego y establecemos su título que en este caso es 'Atari Pong'.

Configuración del Juego



```
16 # Se define el reloj y los FPS de nuestro juego
17 clock = pygame.time.Clock()
18 FPS = 60
19
20 # Definir las dimensiones y posiciones iniciales de las paletas y la pelota
21 PADDLE_WIDTH, PADDLE_HEIGHT = 10, 100
22 BALL_SIZE = 10
23
24 pos_jugador1 = [50, HEIGHT // 2 - PADDLE_HEIGHT // 2]
25 pos_jugador2 = [WIDTH - 50 - PADDLE_WIDTH, HEIGHT // 2 - PADDLE_HEIGHT // 2]
26 pos_bola = [WIDTH // 2, HEIGHT // 2]
27 vel_pel = [5, 5]
28
29 # Velocidad de las paletas
30 vel_paletas = 7
31
32 # Empiezan las puntuaciones
33 punt_jugador1 = 0
34 punt_jugador2 = 0
```

Aquí configuramos el reloj y los FPS del juego para que se ejecute de manera fluida. Definimos las dimensiones y posiciones iniciales de las paletas y la pelota, así como sus velocidades. También inicializamos las puntuaciones de ambos jugadores.

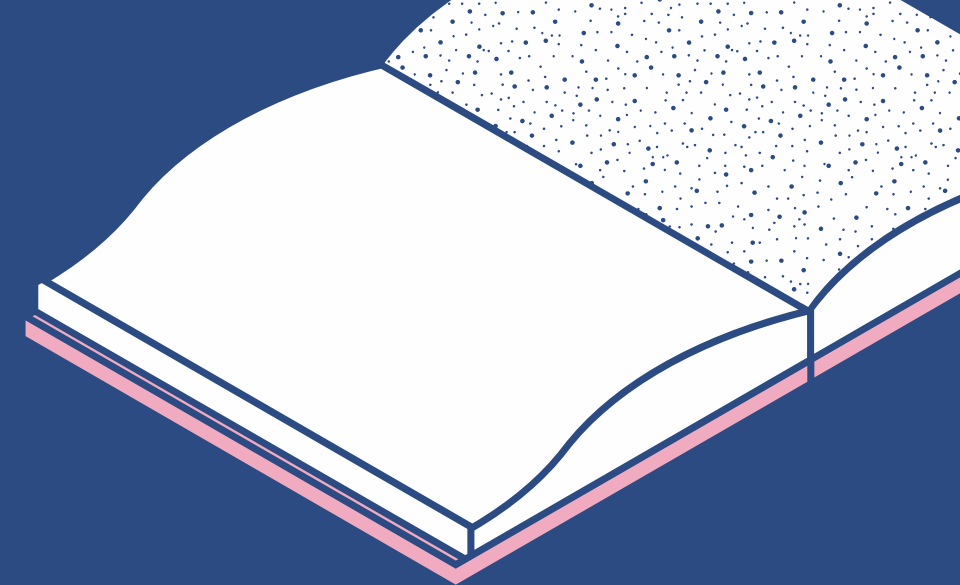
Dibujado de Objetos y Puntuaciones



```
52 # Dibujamos las paletas y la pelota que aparecerán en el juego
53 def draw_objects():
54     screen.fill(BLACK)
55     pygame.draw.rect(screen, WHITE, (pos_jugador1[0], pos_jugador1[1], PADDLE_WIDTH, PADDLE_HEIGHT))
56     pygame.draw.rect(screen, WHITE, (pos_jugador2[0], pos_jugador2[1], PADDLE_WIDTH, PADDLE_HEIGHT))
57     pygame.draw.ellipse(screen, WHITE, (pos_bola[0], pos_bola[1], BALL_SIZE, BALL_SIZE))
58     pygame.draw.aaline(screen, WHITE, (WIDTH // 2, 0), (WIDTH // 2, HEIGHT))
59     draw_scores()
60     pygame.display.flip()
61
62 def draw_scores():
63     # Dibujamos las puntuaciones de los jugadores en la pantalla.
64     font = pygame.font.Font(None, 74)
65     text = font.render(str(punt_jugador1), 1, WHITE)
66     screen.blit(text, (250, 10))
67     text = font.render(str(punt_jugador2), 1, WHITE)
68     screen.blit(text, (WIDTH - 250, 10))
```

Estas funciones son responsables de dibujar los objetos en la pantalla: las paletas, la pelota y la línea central. La función `draw_scores` es la responsable de mostrar las puntuaciones de los jugadores en la parte superior de la pantalla.

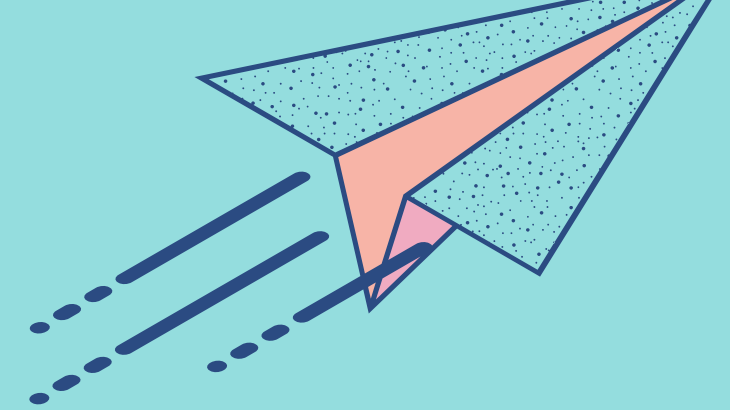
Movimiento de las paletas



```
71 # Aquí definimos las teclas de movimiento
72 def move_paddles(keys):
73     if keys[pygame.K_w] and pos_jugador1[1] > 0:
74         pos_jugador1[1] -= vel_paletas
75     if keys[pygame.K_s] and pos_jugador1[1] < HEIGHT - PADDLE_HEIGHT:
76         pos_jugador1[1] += vel_paletas
77     if keys[pygame.K_UP] and pos_jugador2[1] > 0:
78         pos_jugador2[1] -= vel_paletas
79     if keys[pygame.K_DOWN] and pos_jugador2[1] < HEIGHT - PADDLE_HEIGHT:
80         pos_jugador2[1] += vel_paletas
```

Esta función controla el movimiento de las paletas. Dependiendo de las teclas presionadas (W y S para el Jugador 1, y las flechas Arriba y Abajo para el Jugador 2), las paletas se mueven hacia arriba o hacia abajo dentro de los límites de la pantalla.

Movimiento de las pelotas

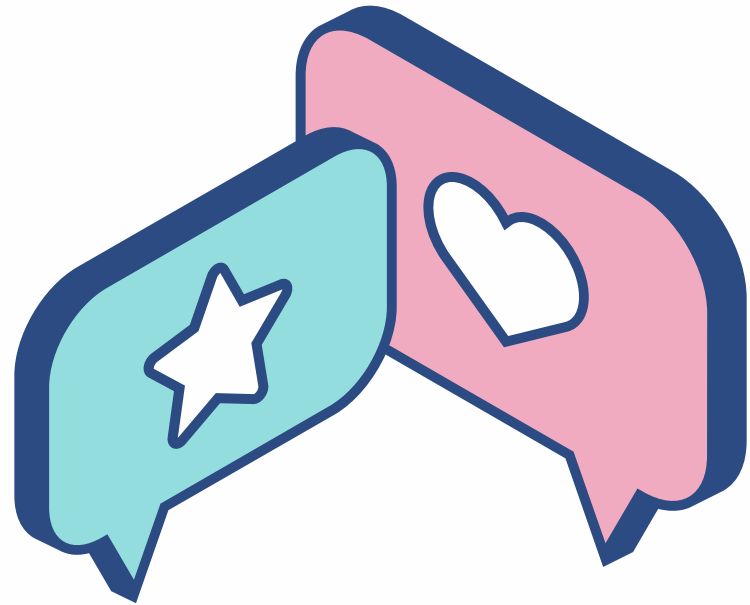


```
#Efecto de sonido colisión de la pelota
sonido_colision = pygame.mixer.Sound('efecto_bola.wav')
```

```
82 # Aquí nosotros definimos cómo se va a comportar la pelota
83 def move_ball():
84     global vel_pel, punt_jugador1, punt_jugador2
85     pos_bola[0] += vel_pel[0]
86     pos_bola[1] += vel_pel[1]
87
88     if pos_bola[1] <= 0 or pos_bola[1] >= HEIGHT - BALL_SIZE:
89         vel_pel[1] = -vel_pel[1]
90         sonido_colision.play()
91
92     if (pos_bola[0] <= pos_jugador1[0] + PADDLE_WIDTH and pos_jugador1[1] < pos_bola[1] < pos_jugador1[1] + PADDLE_HEIGHT) or \
93        (pos_bola[0] >= pos_jugador2[0] - BALL_SIZE and pos_jugador2[1] < pos_bola[1] < pos_jugador2[1] + PADDLE_HEIGHT):
94         vel_pel[0] = -vel_pel[0]
95         sonido_colision.play()
96
97     if pos_bola[0] <= 0:
98         punt_jugador2 += 1
99         reset_ball()
100     if pos_bola[0] >= WIDTH - BALL_SIZE:
101         punt_jugador1 += 1
102         reset_ball()
```

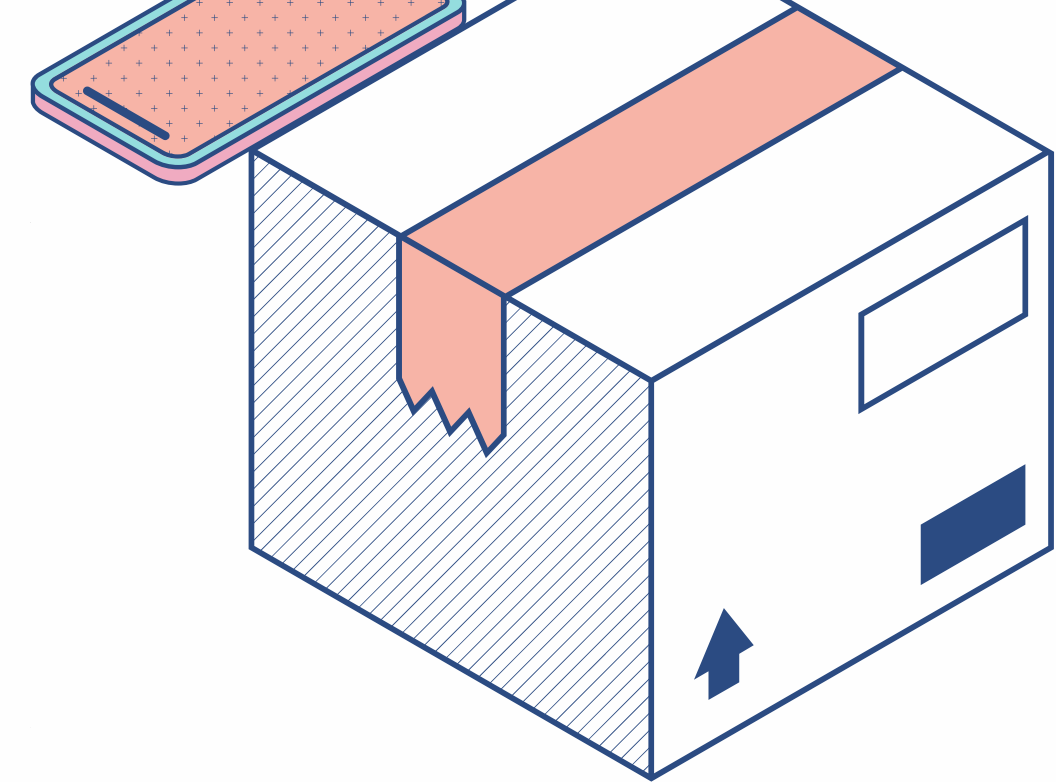
En la primera imagen se puede ver cómo implementamos un efecto de sonido cada vez que la pelota rebota con algo. La función `move_ball` se encarga de mover la pelota en la pantalla. La pelota rebota en las paredes superior e inferior, y también en las paletas de los jugadores. Si la pelota pasa una paleta, se incrementa la puntuación del jugador contrario y la pelota se reinicia en el centro de la pantalla.

Reinicio de la pelota



```
104 # Reinicia la posición y velocidad de la pelota
105 def reset_ball():
106     pos_bola[0], pos_bola[1] = WIDTH // 2, HEIGHT // 2
107     vel_pel[0] = -vel_pel[0]
```

Esta función reinicia la posición de la pelota en el centro de la pantalla y cambia su dirección para que el juego continúe de manera justa después de que se haya anotado un punto.



Menú de Pausa

```
109 # Esta función dibuja el menú de pausa
110 def draw_pause_menu():
111     font = pygame.font.Font(None, 74)
112     texto_reanudar = font.render("Reanudar", True, GRAY if opcion_seleccionada != 0 else WHITE)
113     texto_salir = font.render("Salir", True, GRAY if opcion_seleccionada != 1 else WHITE)
114
115     screen.blit(texto_reanudar, (WIDTH // 2 - texto_reanudar.get_width() // 2, HEIGHT // 2 - 100))
116     screen.blit(texto_salir, (WIDTH // 2 - texto_salir.get_width() // 2, HEIGHT // 2))
```

Estas funciones dibujan los menús de pausa e inicial. El menú de pausa se activa al presionar ENTER durante el juego, permitiendo reanudar o salir. El menú inicial aparece al inicio del juego con opciones para comenzar o salir, como se puede ver implementamos una imagen de fondo para el menú utilizando IA.

Menú Inicial

```
# Imagen de fondo del menú de pausa
fondo_menu_inicial = pygame.image.load('fondo_menu_resized.jpg')

120 # Esta función dibuja el menú principal del juego
121 def draw_initial_menu():
122     screen.blit(fondo_menu_inicial, (0, 0))
123     font = pygame.font.Font(None, 74)
124     texto_iniciar = font.render("Iniciar juego", True, GRAY if opcion_menu_inicial != 0 else WHITE)
125     texto_salir = font.render("Salir", True, GRAY if opcion_menu_inicial != 1 else WHITE)
126
127     screen.blit(texto_iniciar, (WIDTH // 2 - texto_iniciar.get_width() // 2, HEIGHT // 2 - 100))
128     screen.blit(texto_salir, (WIDTH // 2 - texto_salir.get_width() // 2, HEIGHT // 2))
129
130     pygame.display.flip()
```


Mensaje de victoria

```
132 #Aquí se dibuja el mensaje del ganador
133 def draw_winner_message(winner):
134     font = pygame.font.Font(None, 74)
135     texto_ganador = font.render(f"{winner} Ganó", True, WHITE)
136
137     screen.fill(BLACK)
138     screen.blit(texto_ganador, (WIDTH // 2 - texto_ganador.get_width() // 2, HEIGHT // 2))
139     pygame.display.flip()
```

Esta función muestra un mensaje de victoria en la pantalla cuando un jugador alcanza 10 puntos. El texto indica cuál jugador ganó, y el fondo de la pantalla se cambia a negro.

Función Principal

```
141 # Esta es la función principal que va a ejecutar el juego
142 def main():
143     global juego_pausa, opcion_seleccionada, menu_inicial, opcion_menu_inicial, punt_jugador1, punt_jugador2
144
145     while True:
146         for event in pygame.event.get():
147             if event.type == pygame.QUIT:
148                 pygame.quit()
149                 sys.exit()
150
151             if event.type == pygame.KEYDOWN:
152                 if menu_inicial:
153                     if event.key == pygame.K_RETURN:
154                         if opcion_menu_inicial == 0:
155                             menu_inicial = False
156                         elif opcion_menu_inicial == 1:
157                             pygame.quit()
158                             sys.exit()
159                     if event.key == pygame.K_UP:
160                         opcion_menu_inicial = (opcion_menu_inicial - 1) % 2
161                     if event.key == pygame.K_DOWN:
162                         opcion_menu_inicial = (opcion_menu_inicial + 1) % 2
163                 elif juego_pausa:
164                     if event.key == pygame.K_RETURN:
165                         if opcion_seleccionada == 0:
166                             juego_pausa = False
167                         elif opcion_seleccionada == 1:
168                             pygame.quit()
169                             sys.exit()
```

La función main es como el corazón del juego. No alcanzó toda la función en esta diapositiva, pero esta función maneja la lógica del menú inicial, el menú de pausa y el bucle principal del juego. En esta función también verificamos si alguno de los jugadores ha ganado al alcanzar 10 puntos, y en ese caso, mostramos el mensaje de victoria y esperamos a que se presione Enter para cerrar el juego.

Entonces para resumir, hemos desarrollado un juego inspirado en el clásico Pong utilizando Pygame. Hemos cubierto la configuración inicial, el manejo de los controles, la lógica del juego, y cómo mostrar mensajes de victoria. Ha sido todo un desafío implementar más y más funciones para el juego, pero espero que nuestro proyecto haya sido de su agrado. Muchas gracias.

MUCHAS GRACIAS

HECHO POR: JIMMY Y LUIS

