

Documentación Juego Pac-Man

Parcial Corte I

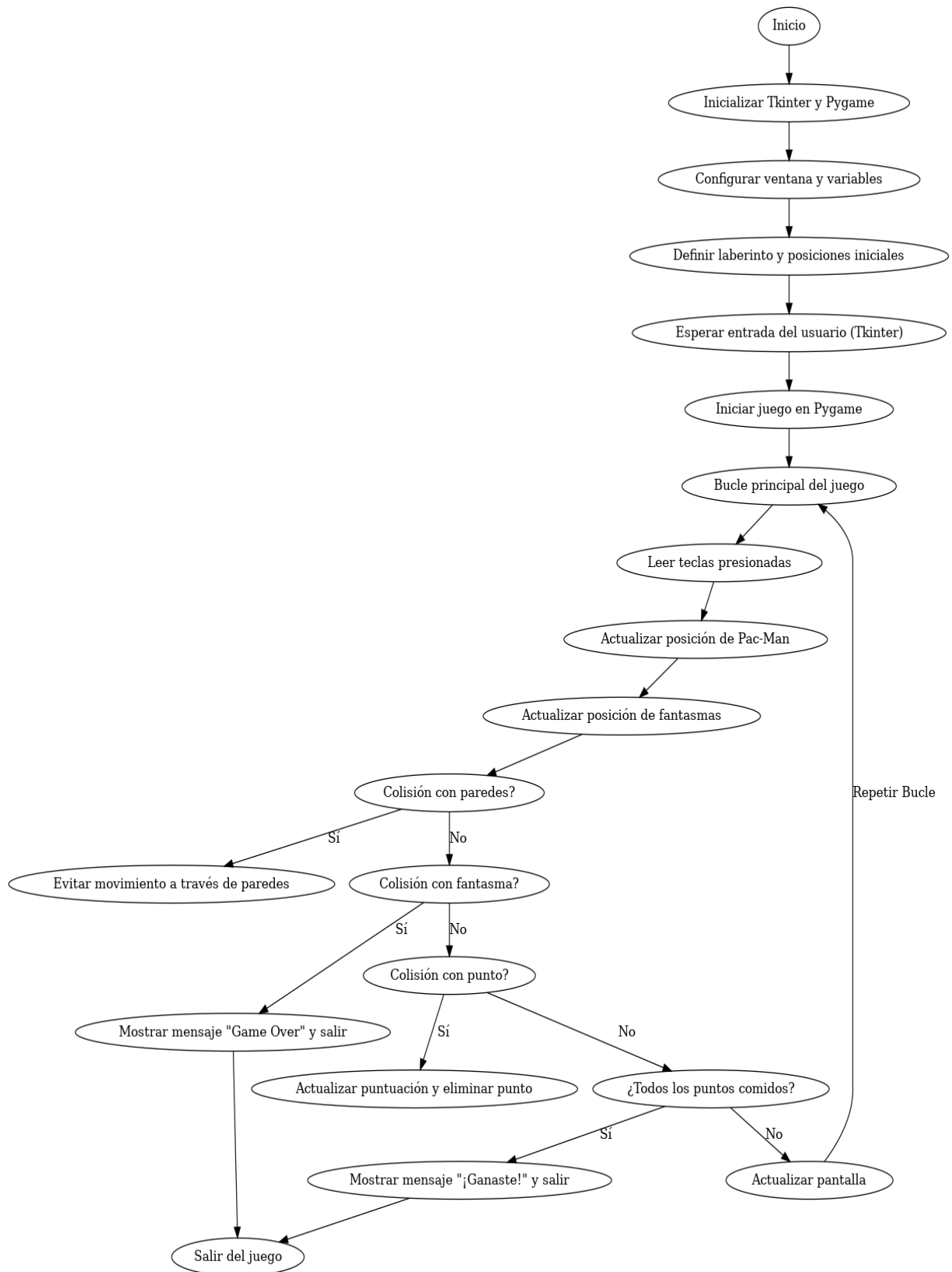
Programación Orientada A Objetos

Brahian Alexis Lozano Gonzalez (409214)

Nikolas Valencia Arteaga (409500)

Universidad Unicatólica Lumen Gentium

Fecha: 22 De Marzo Del 2025



Este documento describe en detalle el funcionamiento del juego de Pac-Man desarrollado en Python, utilizando las librerías pygame para la jugabilidad y tkinter para la interfaz gráfica. El juego cuenta con un laberinto, fantasmas que se mueven aleatoriamente y puntos que Pac-Man debe recolectar para ganar la partida.

Requisitos

Para ejecutar este juego, es necesario tener instaladas las siguientes librerías:

- pygame: para la mecánica del juego.
- tkinter: para la interfaz gráfica de inicio.

```
PS C:\Users\gcbra> pip install pygame
```

Explicación del Código

1. Importación de Librerías

El código importa las librerías necesarias:

```
1  import pygame
2  import random
3  import sys
4  import tkinter as tk
5  from tkinter import messagebox
```

2. Configuración de colores

Se definen los colores en formato RGB

```
10 # Colores
11 NEGRO = (0, 0, 0)
12 AMARILLO = (255, 255, 0)
13 AZUL = (0, 0, 255)
14 ROJO = (255, 0, 0)
15 BLANCO = (255, 255, 255)
```

3. Configuración de la Ventana y tkinter

Se configura la ventana principal de tkinter y se define una variable de puntuación:

```
30 # Ventana de Tkinter
31 root = tk.Tk()
32 root.title("Pac-Man")
```

Se crean botones y etiquetas para la interfaz gráfica:

```
128 # Botones en la ventana de Tkinter
129 tk.Label(root, text="Pac-Man", font=("Arial", 14)).pack()
130 tk.Button(root, text="Iniciar Juego", command=iniciar_juego).pack()
131 tk.Label(root, textvariable=puntos_comidos, font=("Arial", 12)).pack()
```

4. Definición del Laberinto

Se crean las paredes del laberinto usando pygame.Rect.

```
20 # Laberinto con más muros
21 paredes = [
22     pygame.Rect(0, 0, 600, 10), pygame.Rect(0, 390, 600, 10),
23     pygame.Rect(0, 0, 10, 400), pygame.Rect(590, 0, 10, 400),
24     pygame.Rect(50, 50, 500, 10), pygame.Rect(50, 340, 500, 10),
25     pygame.Rect(50, 50, 10, 300), pygame.Rect(540, 50, 10, 300),
26     pygame.Rect(150, 100, 300, 10), pygame.Rect(150, 300, 300, 10),
27     pygame.Rect(100, 200, 400, 10), pygame.Rect(250, 100, 10, 200)
28 ]
```

5. Configuración de Pac-Man

Se define su posición inicial, velocidad y tamaño:

```
49 pacman_pos = [100, 100]
50 pacman_vel = 4
51 pacman_radio = 15
```

6. Configuración de los Fantasmas

Se crean los fantasmas con posiciones iniciales y velocidades aleatorias:

```
53 fantasmas = [pygame.Rect(450, 100, 20, 20), pygame.Rect(450, 250, 20, 20)]
54 fantasmas_vel = 2
55 fantasmas_direcciones = [random.choice([(fantasmas_vel, 0), (-fantasmas_vel, 0), (0, fantasmas_vel), (0, -fantasmas_vel)]) for _ in fantasmas]
```

7. Generación de Puntos

Los puntos se generan en ubicaciones donde no haya muros:

```
57 puntos = [pygame.Rect(x, y, 10, 10) for x in range(60, 540, 40) for y in range(60, 340, 40) if not any(pygame.Rect(x, y, 10, 10).colliderect(p) for p in paredes)]
```

8. Movimiento de Fantasmas

Los fantasmas cambian de dirección si chocan con una pared:

```
62 def mover_fantasma(idx):
63     fantasma = fantasmas[idx]
64     dx, dy = fantasmas_direcciones[idx]
65     nueva_pos = fantasma.move(dx, dy)
66     if any(nueva_pos.colliderect(pared) for pared in paredes):
67         nuevas_direcciones = [(fantasmas_vel, 0), (-fantasmas_vel, 0), (0, fantasmas_vel), (0, -fantasmas_vel)]
68         random.shuffle(nuevas_direcciones)
69         for nueva_dx, nueva_dy in nuevas_direcciones:
70             nueva_pos = fantasma.move(nueva_dx, nueva_dy)
71             if not any(nueva_pos.colliderect(pared) for pared in paredes):
72                 fantasmas_direcciones[idx] = (nueva_dx, nueva_dy)
73                 break
74     fantasma.x += fantasmas_direcciones[idx][0]
75     fantasma.y += fantasmas_direcciones[idx][1]
```

9. Lógica de Juego

El juego se ejecuta en un bucle que maneja eventos de teclado y actualiza posiciones:

```
77 while ejecutando:
78     for evento in pygame.event.get():
79         if evento.type == pygame.QUIT:
80             pygame.quit()
81             return
```

10. Colisiones y Puntuación

Se detectan colisiones con puntos y fantasmas:

```
104         for punto in puntos[:]:
105             if pacman_rect.colliderect(punto):
106                 puntos.remove(punto)
107                 puntos_comidos.set(puntos_comidos.get() + 1)
```

Si Pac-Man es atrapado:

```
100         if pacman_rect.colliderect(fantasmas[i]):
101             messagebox.showinfo("Game Over", "¡Pac-Man ha sido atrapado!")
102             return
```

Si gana el juego:

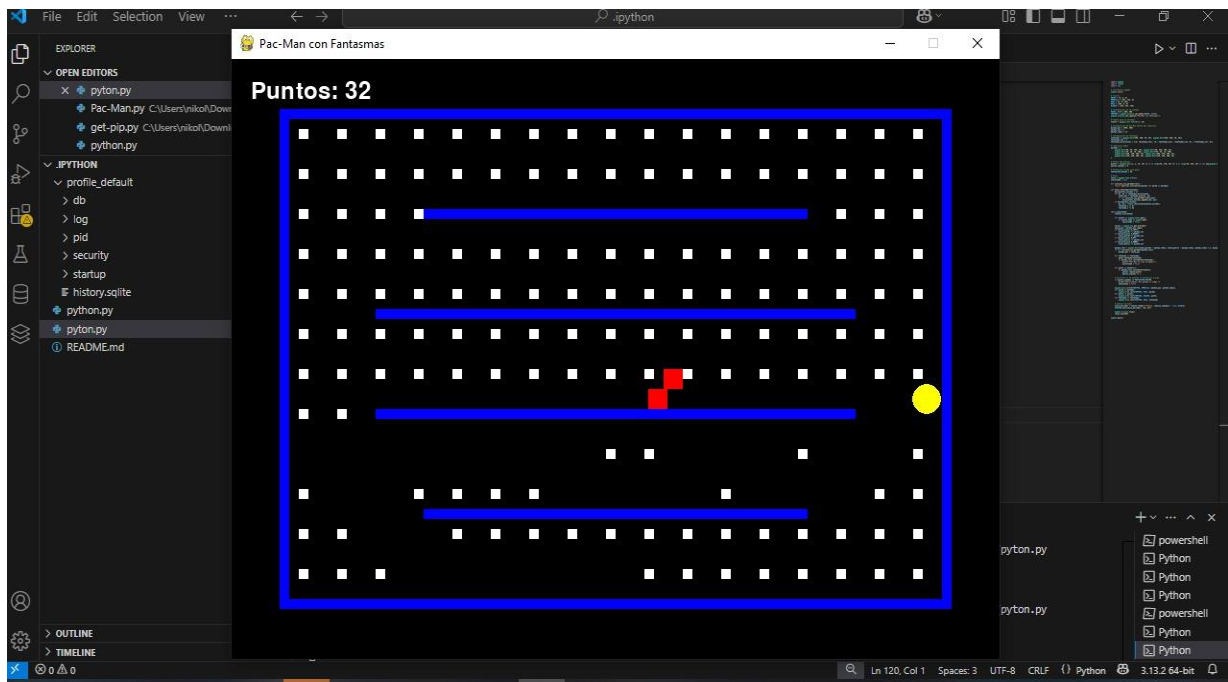
```
109         if not puntos:
110             messagebox.showinfo("¡Ganaste!", "Has comido todos los puntos.")
111             return
```

Conclusión

Este juego de Pac-Man implementa mecánicas básicas como movimiento, colisiones y un sistema de puntuación. tkinter se utiliza para gestionar la interfaz inicial, permitiendo iniciar una nueva partida y mostrar la puntuación en tiempo real.

Errores

1. Error de paredes



```

29 # Laberinto ampliado estilo clásico
30 paredes = [
31     pygame.Rect(50, 50, 500, 10), pygame.Rect(50, 340, 500, 10),
32     pygame.Rect(50, 50, 10, 300), pygame.Rect(540, 50, 10, 300),
33     pygame.Rect(200, 100, 200, 10), pygame.Rect(200, 250, 200, 10),
34     pygame.Rect(300, 150, 10, 100), pygame.Rect(100, 150, 10, 100),
35     pygame.Rect(490, 150, 10, 100), pygame.Rect(150, 200, 300, 10)
36 ]

```

2. Error movimiento de fantasmas

```

46 def colision_con_paredes(rect):
47     """Verifica si un rectángulo colisiona con alguna pared."""
48     return any(rect.colliderect(pared) for pared in paredes)
49
50 def mover_fantasma(fantasma):
51     """ Movimiento del fantasma evitando paredes """
52     direcciones = [(fantasmas_vel, 0), (-fantasmas_vel, 0), (0, fantasmas_vel), (0, -fantasmas_vel)]
53     random.shuffle(direcciones) # Para que no siempre siga el mismo patrón
54     for dx, dy in direcciones:
55         nueva_pos = fantasma.move(dx, dy)
56         if not colision_con_paredes(nueva_pos):
57             fantasma.x, fantasma.y = nueva_pos.x, nueva_pos.y
58             break

```

