Documento parcial 1

Nombres

Santiago Jembuel Calambaz

Daniel Becerra Solis

1. Introducción

Este presente informe nos describe un desarrollo de un juego más conocido como pac-man, donde utilizamos la biblioteca pygame en Python, se implemente una cuadrícula o cuadro que es el laberinto del juego, en el cual el usuario puede controlar a pac-man y poder que pac-man recolecte los rompe cocos sin que los fantasmas lo atrapen.

Este código representa la variación de los laberintos, los movimientos aleatorios programados por los fantasmas y la implementación de un sistema de puntuación y la detención de una victoria y una derrota.

Estructura del juego.

Este código se divide en varias secciones importantes.

1. Inicialización de Pygame y Configuración de la Pantalla
2. Definición de Colores y Fuente para la Interfaz
3. Generación Aleatoria del Laberinto
4. Ubicación Inicial de Pac-Man y los Fantasmas
5. Dibujo del Laberinto en la Pantalla
6. Movimiento del Jugador y de los Fantasmas
7. Condiciones de Victoria o Derrota
8. Bucle Principal del Juego

```
pygame.init()


TAMANO_CELDA = 30


FILAS, COLUMNAS = 10, 10


ANCHO, ALTO = COLUMNAS * TAMANO_CELDA, FILAS * TAMANO_CELDA + 50


pantalla = pygame.display.set_mode((ANCHO, ALTO))
```

pygame.display.set_caption("Pac-Man").

Se inicia el pygame.

1. Se define una cuadrícula de 10x10 celdas con tamaño de 30 píxeles cada una.
2. Se crea una ventana con dimensiones proporcionales al tamaño del laberinto
3. Se deja un espacio adicional de 50 píxeles en la parte inferior para mostrar la puntuación

Definición y colores de la fuente.

NEGRO = (0, 0, 0)

BLANCO = (255, 255, 255)

ROJO = (255, 0, 0)

AMARILLO = (255, 255, 0)

AZUL = (0, 0, 255)

fuente = pygame.font.Font(None, 36)

Se asignan estos colores en formato RGB para diferentes elementos del juego.

1. Se define una fuente de diseño de texto para la puntuación

Generación aleatoria del laberinto.

```
def generar_laberinto():

laberinto = [[0 for _ in range(COLUMNAS)] for _ in range(FILAS)]

for i in range(FILAS):
  for j in range(COLUMNAS):
    if random.random() < 0.2:
        laberinto[i][j] = 1
 laberinto[1][1] = 3

laberinto[FILAS-2][COLUMNAS-2] = 4

 laberinto[FILAS-2][1] = 4

for i in range(FILAS):
   for j in range(COLUMNAS):
      if laberinto[i][j] == 0 and random.random() < 0.5:
```

```
                laberinto[i][j] = 2

    return laberinto
```

Creamos una matriz de 10x10 en donde se muestra estas siguientes partes:

- **0:** Espacios libres
- **1:** Paredes (que tienen una posibilidad de 20%)
- **2:** Rompe cocos (que tiene un 50% de probabilidad en espacios libres)
- **3:** Pac-man (que está en una posición de (1,1))
- **4:** Fantasmas (ellos están en dos posiciones aleatoriamente)

Detención de las posiciones iniciales.

```
def encontrar_posiciones():

    global posicion_jugador, posiciones_fantasmas

    posiciones_fantasmas = []

for i in range(FILAS):

    for j in range(COLUMNAS):

        if laberinto_matriz[i][j] == 3:

            posicion_jugador = (i, j)

elif laberinto_matriz[i][j] == 4:

    posiciones_fantasmas.append((i, j))
```

Se recorre la matriz del laberinto para saber donde esta la posición de pac-man

```
def dibujar_laberinto():
```

```
    pantalla.fill(NEGRO)
for i in range(FILAS):
    for j in range(COLUMNAS):
x, y = j * TAMANO_CELDA, i * TAMANO_CELDA
if laberinto_matriz[i][j] == 1:
     pygame.draw.rect(pantalla, BLANCO, (x, y, TAMANO_CELDA,
TAMANO_CELDA))
            elif laberinto_matriz[i][j] == 3:
                pygame.draw.circle(pantalla, AMARILLO, (x + TAMANO_CELDA //
2, y + TAMANO_CELDA // 2), TAMANO_CELDA // 2)
              elif laberinto_matriz[i][j] == 4:
                   pygame.draw.circle(pantalla, ROJO, (x + TAMANO_CELDA // 2,
y + TAMANO_CELDA // 2), TAMANO_CELDA // 2)
                elif laberinto_matriz[i][j] == 2:
                     pygame.draw.circle(pantalla, AZUL, (x + TAMANO_CELDA //
2, y + TAMANO_CELDA // 2), TAMANO_CELDA // 6)


texto_puntaje = fuente.render(f'Puntaje: {puntaje}', True, BLANCO)
pantalla.blit(texto_puntaje, (10, FILAS * TAMANO_CELDA + 10))
pygame.display.flip()
```

Se dibuja el laberinto y se agrega en la pantalla los colores definidos.

- Se muestra la puntuación en la parte inferior de la pantalla.

Movimientos de Pac-man.

```
def mover_jugador(dx, dy):
    global posicion_jugador, puntaje
    x, y = posicion_jugador
    nueva_x, nueva_y = x + dx, y + dy
```

```
    if 0 <= nueva_x < FILAS and 0 <= nueva_y < COLUMNAS:
        if laberinto_matriz[nueva_x][nueva_y] != 1:
            if laberinto_matriz[nueva_x][nueva_y] == 2:
                puntaje += 10
    elif laberinto_matriz[nueva_x][nueva_y] == 4:
        game_over()
   laberinto_matriz[x][y] = 0
  laberinto_matriz[nueva_x][nueva_y] = 3
  posicion_jugador = (nueva_x, nueva_y)
 verificar_victoria()
```

- Se mueve a pac-man solo cuando haya una casilla que no sea una pared
- Si pac-man encuentra un rompe cocos, su puntuación sumara a 10 puntos
- Si pac-man toca un fantasma pac-man será eliminado y pierdes el juego.

Movimientos de los fantasmas

```
def mover_fantasmas():
      global posiciones_fantasmas
       nuevas_posiciones = []
       for x, y in posiciones_fantasmas:
       direcciones = [(0,1), (0,-1), (1,0), (-1,0)]
       random.shuffle(direcciones)
   for dx, dy in direcciones:
        nueva_x, nueva_y = x + dx, y + dy
        if 0 <= nueva_x < FILAS and 0 <= nueva_y < COLUMNAS and
laberinto_matriz[nueva_x][nueva_y] in [0, 2, 3]:
            if laberinto_matriz[nueva_x][nueva_y] == 3:
               game_over()
            laberinto_matriz[x][y] = 0
```

laberinto_matriz[nueva_x][nueva_y] = 4

nuevas_posiciones.append((nueva_x, nueva_y))

break

posiciones_fantasmas = nuevas_posiciones

- Los fantasmas se mueven aleatoriamente en el laberinto
- Si uno de los fantasmas toca a pac-man, automáticamente el jugador pierde

Conclusión.

Este código implementa a pac-man sencillo con generación de laberintos, movimientos y un sistema de puntuación, Esto puede generar o mejorar agregando más inteligencia en los fantasmas, que sean más rápidos y tengan patrones diferentes, y también se puede agregar mas niveles difíciles para que el jugador desarrolle sus capacidades en el juego.