



Powered by
Arizona State University

LOGICA DE PROGRAMACION 1
SNAKE-GAME - PROYECTO FINAL

Nombre: Edison Enriquez

Docente: Monica Salazar

Quito. 17 Diciembre 2024



OBJETIVO

El objetivo de este proyecto es desarrollar el juego clásico de la serpiente utilizando Python y la librería Pygame. El programa integra las funcionalidades requeridas, aplicando los conocimientos adquiridos en las 4 unidades del curso.

APLICACION DE LOS TEMAS DEL CURSO

Unidad 1: Resolución de problemas y manejo de datos.

- Variables como snake, food, score.

Unidad 2: Algoritmos y diagramas de flujo.

- Diseño del flujo lógico del juego.

Unidad 3: Lógica de programación.

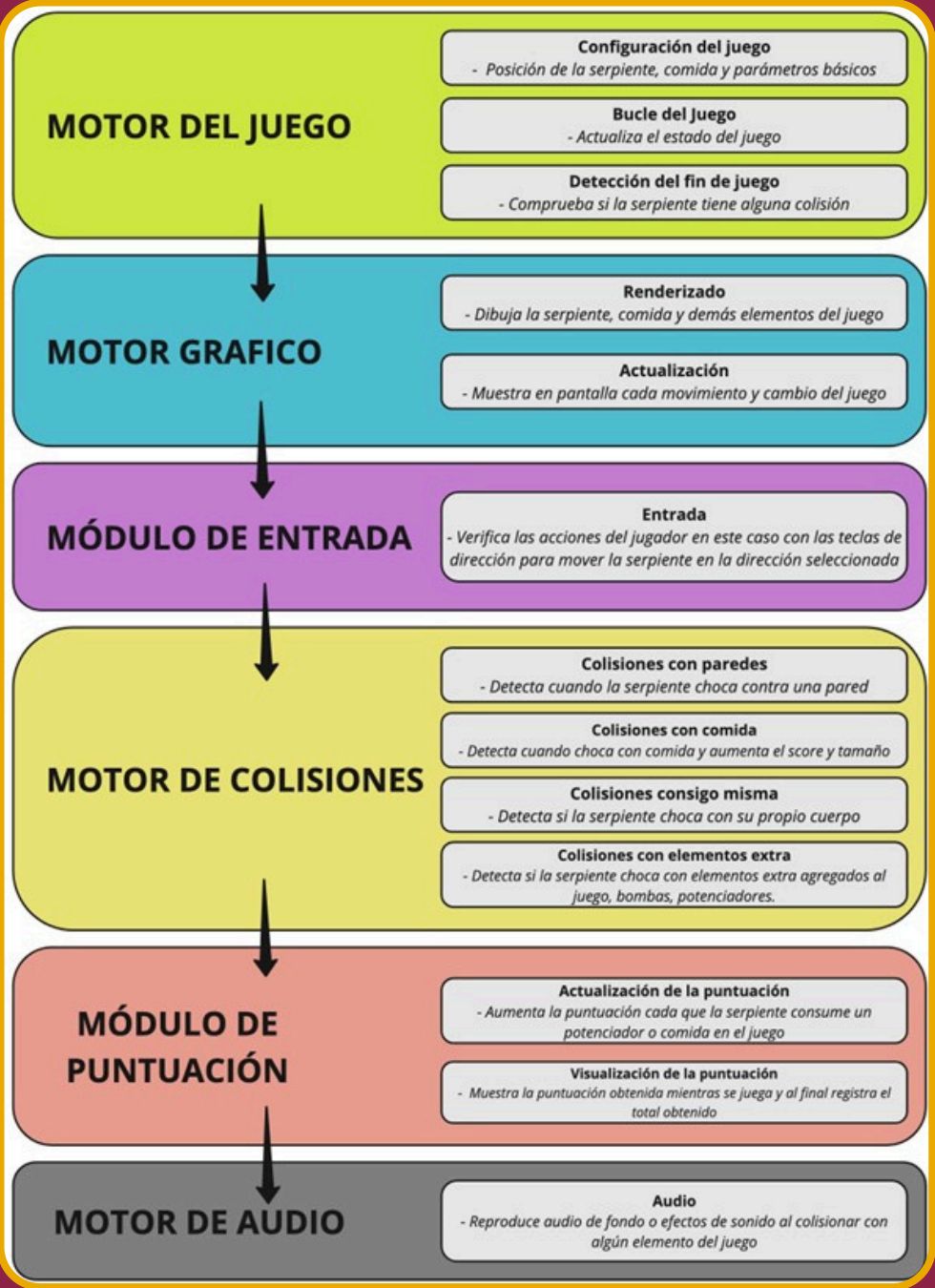
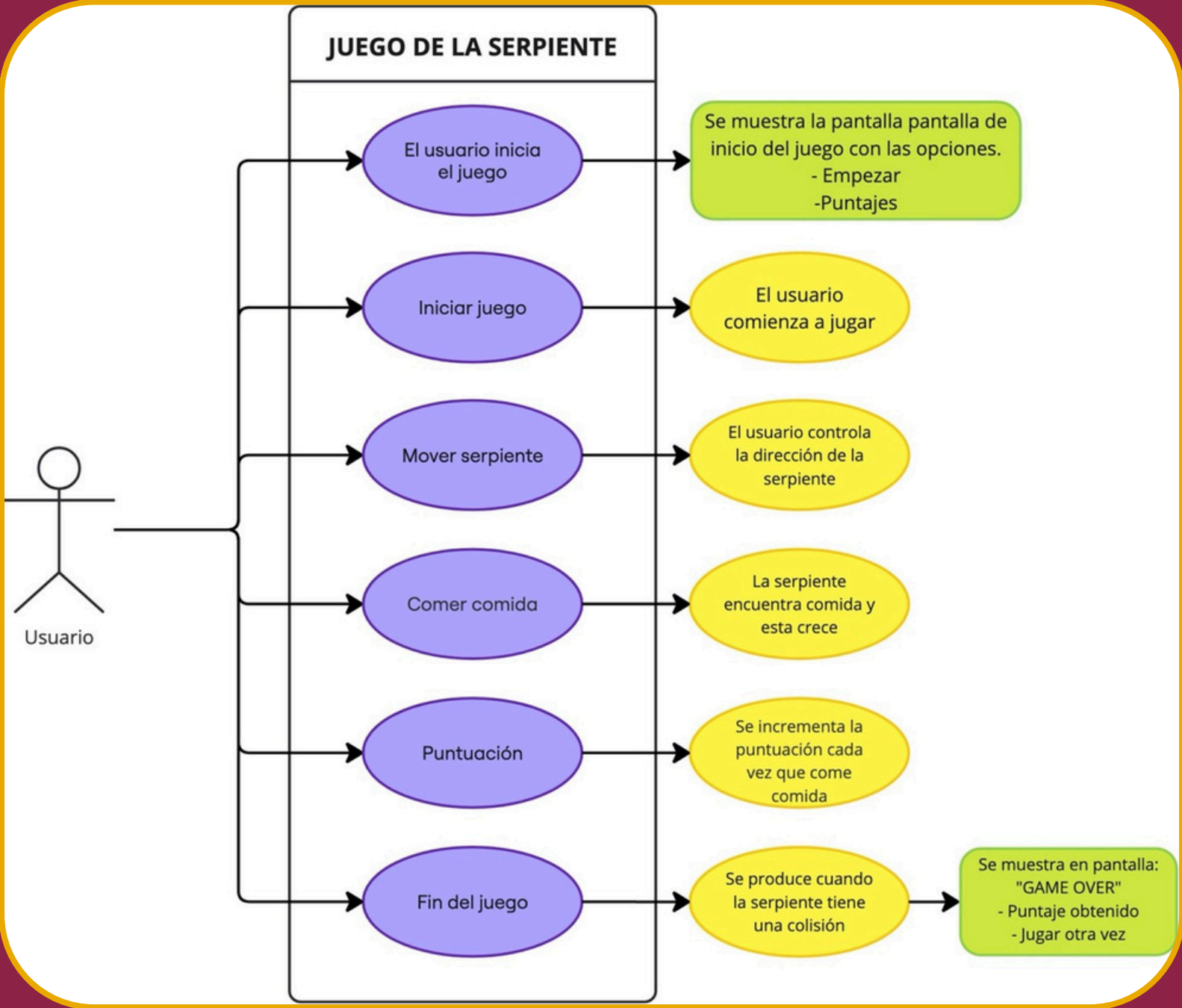
- Uso de condicionales (if) y bucles (while y for).

Unidad 4: Estructuras de datos y funciones.

- Uso de listas y tuplas.
- Modularización del programa con funciones.

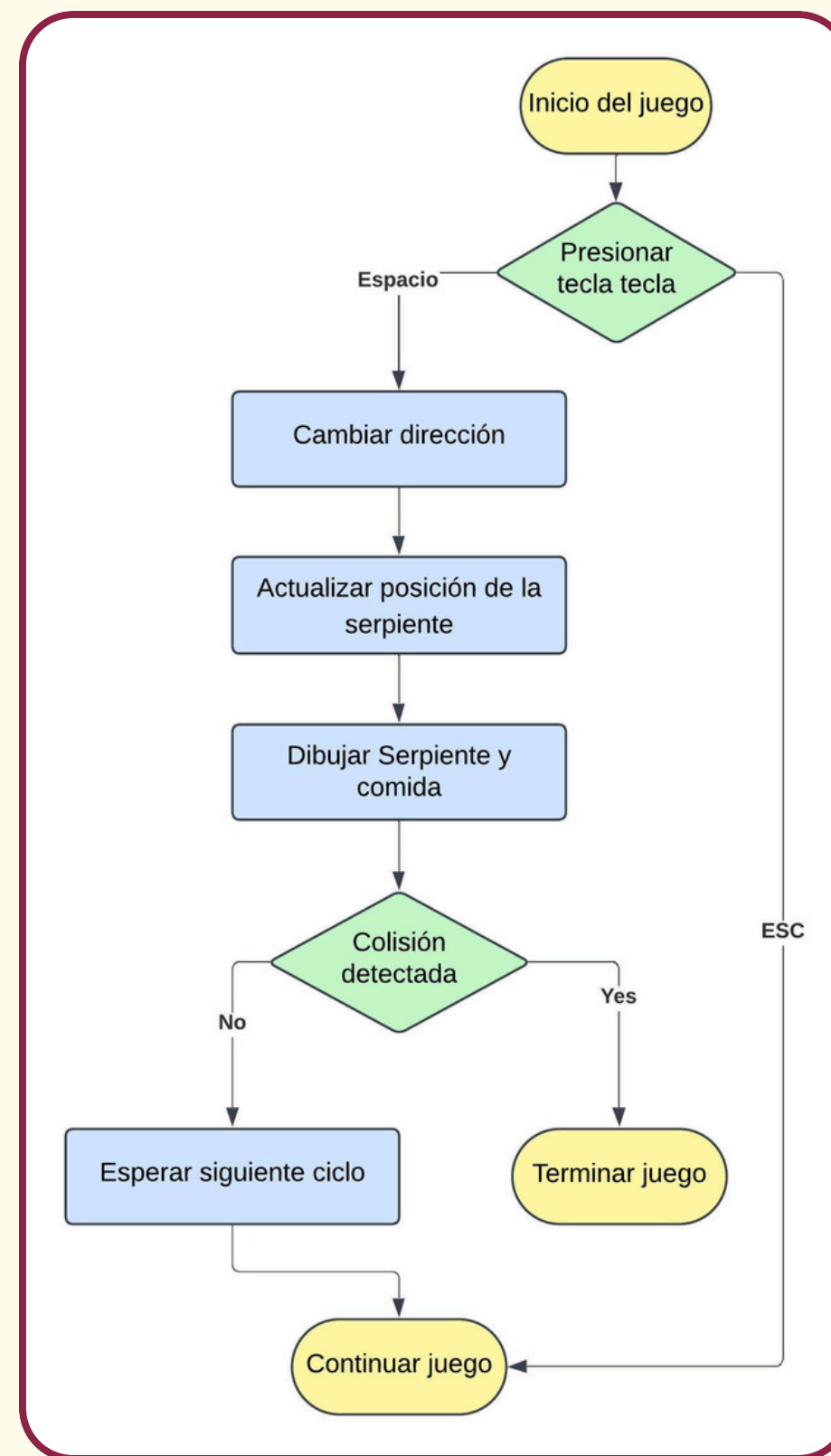
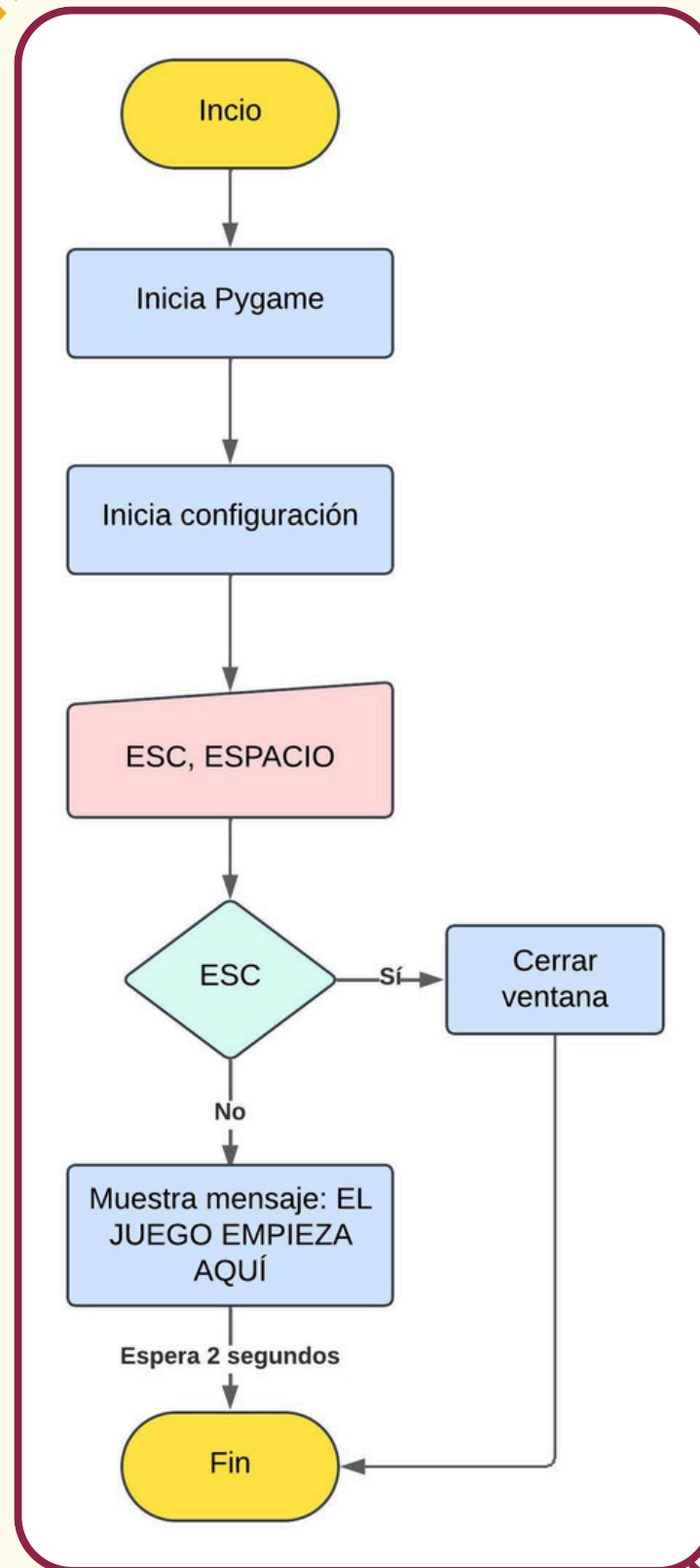


DIAGRAMAS DE FLUJO



En estos diagramas podemos observar el funcionamiento básico del juego de la serpiente y los módulos que se usan para el funcionamiento del mismo.

DIAGRAMAS DE FLUJO



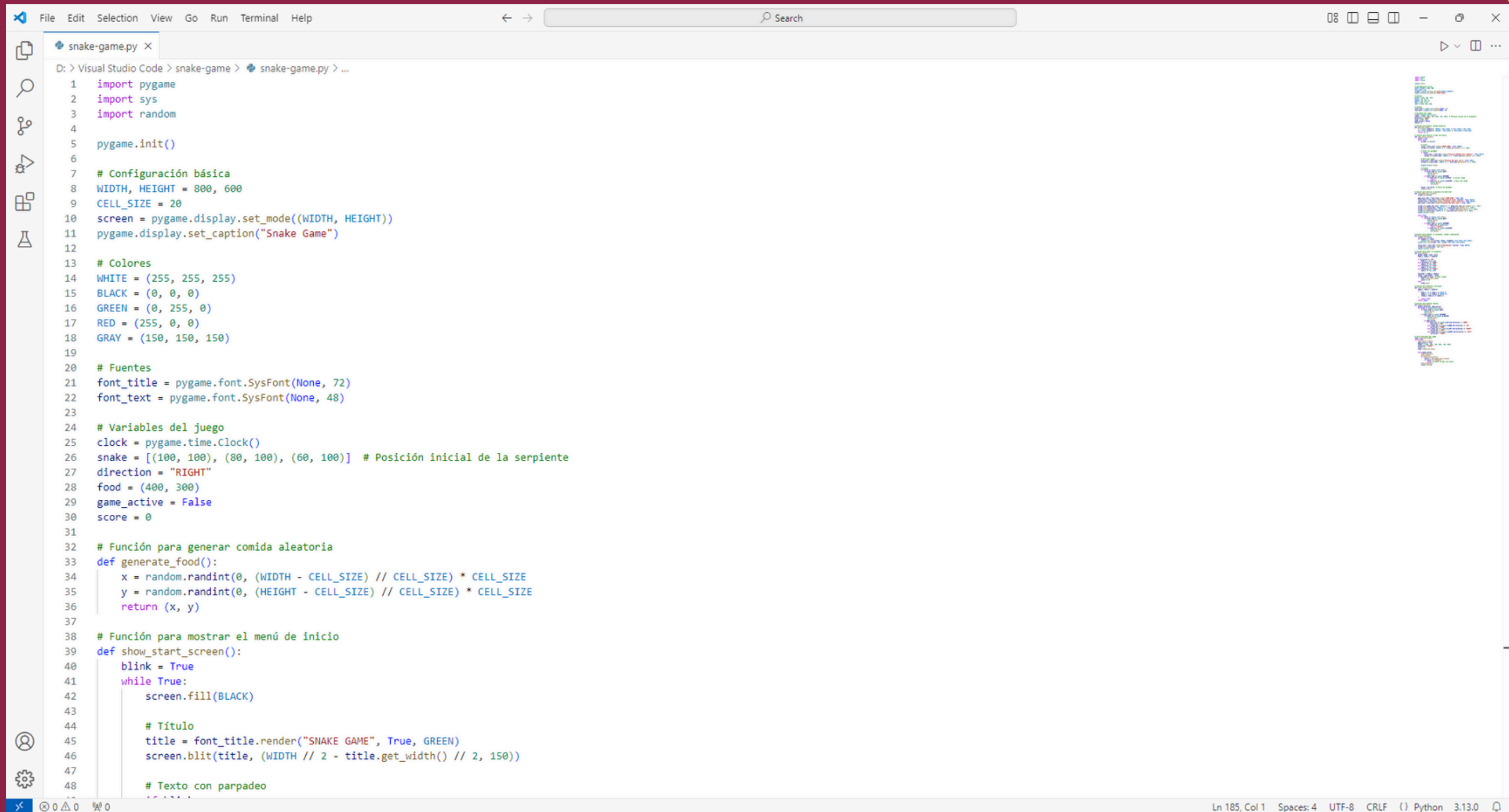
Podemos observar como funciona el juego de la serpiente dentro del código, comenzando por la pantalla de inicio y continuando con el movimiento de la serpiente.

FUNCIONALIDADES PRINCIPALES

- 01. Pantalla de Inicio:** Menú inicial con opciones para jugar o salir.
- 02. Movimiento de la Serpiente:** Controlado con las flechas del teclado.
- 03. Crecimiento al Comer:** La serpiente crece al comer comida generada aleatoriamente.
- 04. Detección de Colisiones:** El juego termina si la serpiente choca con los bordes o consigo misma.
- 05. Pantalla de Game Over:** Muestra la puntuación final y opciones para reiniciar o salir.

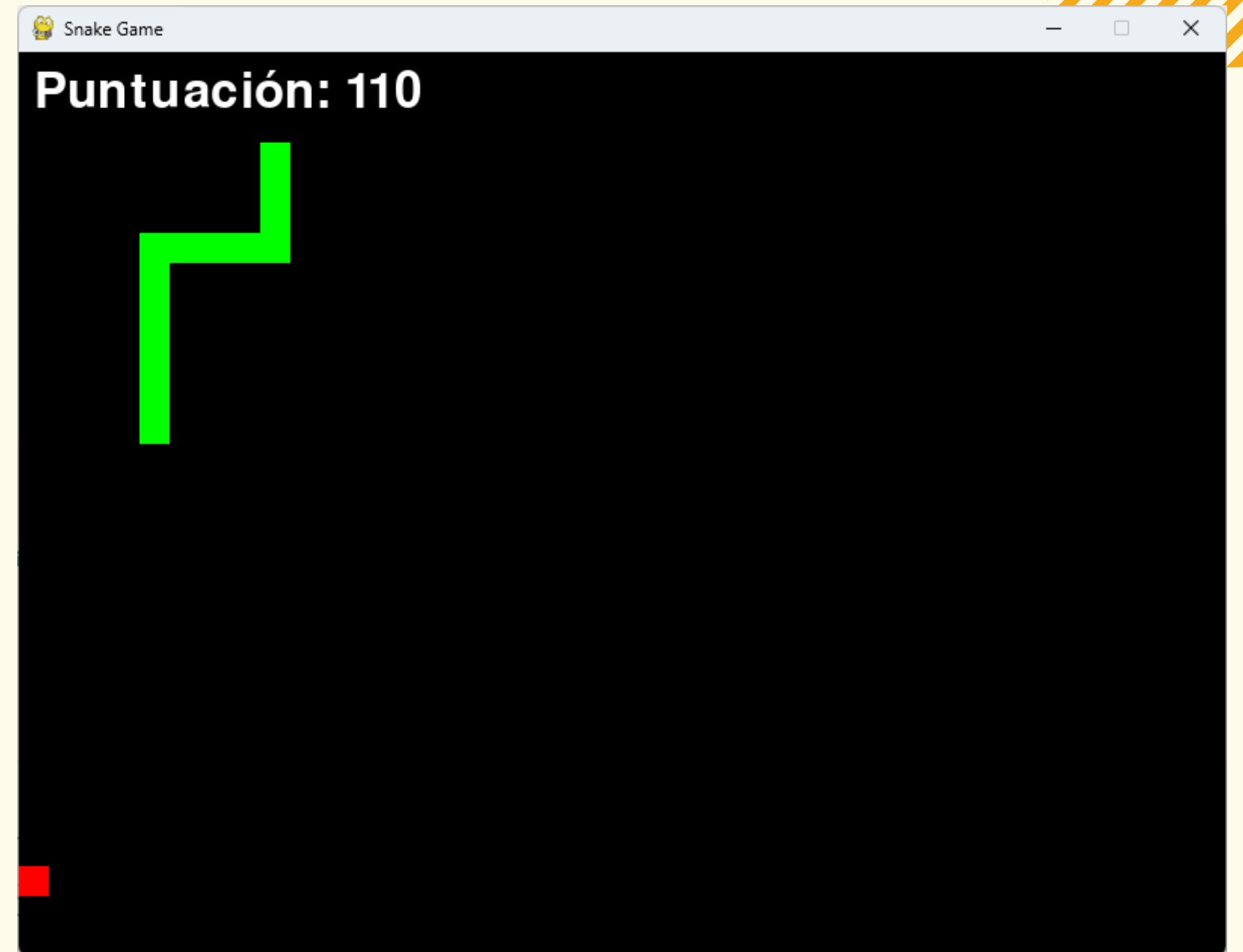
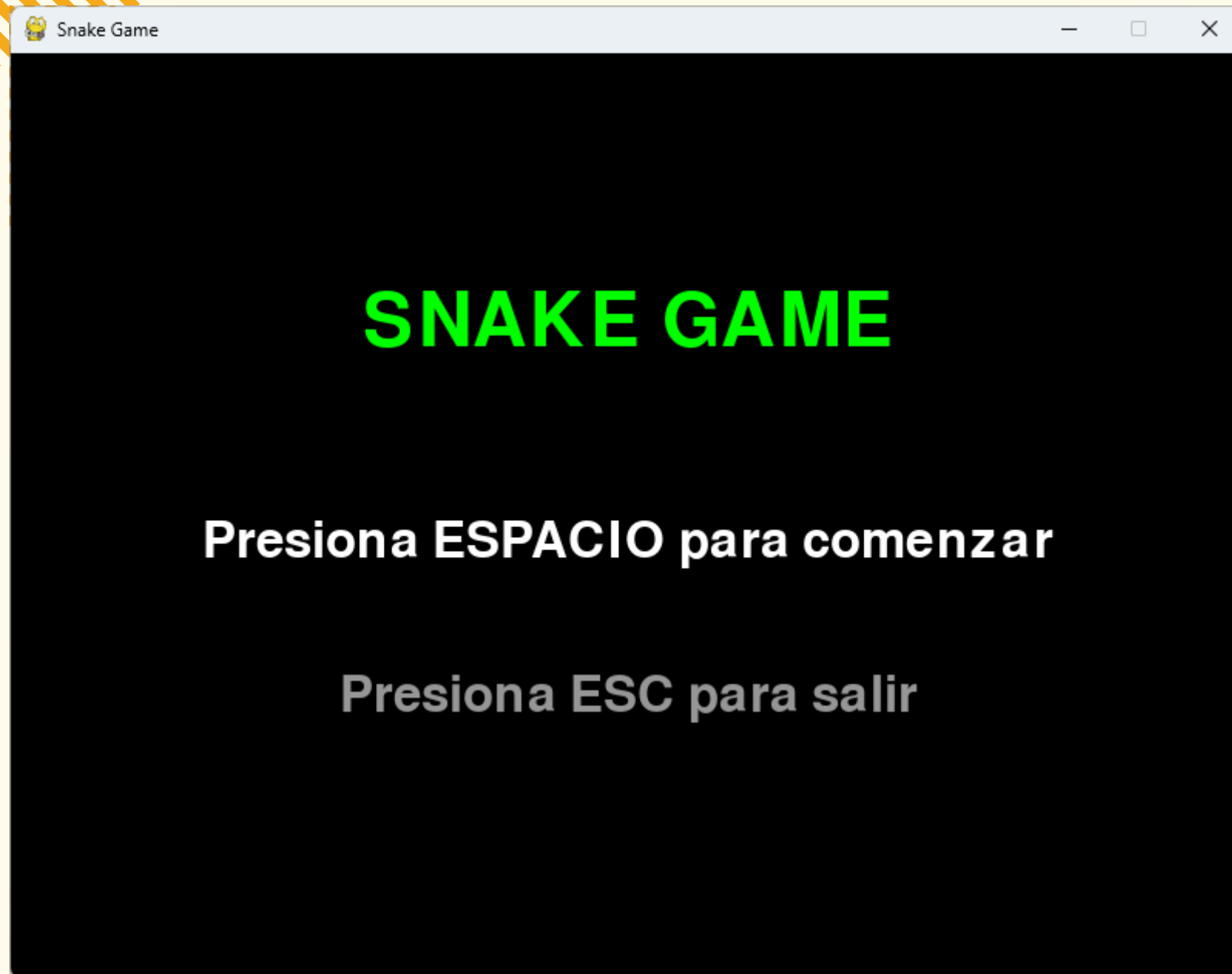


CAPTURAS



```
1 import pygame
2 import sys
3 import random
4
5 pygame.init()
6
7 # Configuración básica
8 WIDTH, HEIGHT = 800, 600
9 CELL_SIZE = 20
10 screen = pygame.display.set_mode((WIDTH, HEIGHT))
11 pygame.display.set_caption("Snake Game")
12
13 # Colores
14 WHITE = (255, 255, 255)
15 BLACK = (0, 0, 0)
16 GREEN = (0, 255, 0)
17 RED = (255, 0, 0)
18 GRAY = (150, 150, 150)
19
20 # Fuentes
21 font_title = pygame.font.SysFont(None, 72)
22 font_text = pygame.font.SysFont(None, 48)
23
24 # Variables del juego
25 clock = pygame.time.Clock()
26 snake = [(100, 100), (80, 100), (60, 100)] # Posición inicial de la serpiente
27 direction = "RIGHT"
28 food = (400, 300)
29 game_active = False
30 score = 0
31
32 # Función para generar comida aleatoria
33 def generate_food():
34     x = random.randint(0, (WIDTH - CELL_SIZE) // CELL_SIZE * CELL_SIZE
35     y = random.randint(0, (HEIGHT - CELL_SIZE) // CELL_SIZE * CELL_SIZE
36     return (x, y)
37
38 # Función para mostrar el menú de inicio
39 def show_start_screen():
40     blink = True
41     while True:
42         screen.fill(BLACK)
43
44         # Título
45         title = font_title.render("SNAKE GAME", True, GREEN)
46         screen.blit(title, (WIDTH // 2 - title.get_width() // 2, 150))
47
48         # Texto con parpadeo
49         text = font_text.render("PULSA ENTER PARA COMENZAR", True, WHITE)
50         screen.blit(text, (WIDTH // 2 - text.get_width() // 2, 300))
51
52         if blink:
53             blink = False
54         else:
55             blink = True
56
57         pygame.display.flip()
58         pygame.time.wait(1000)
59
60     return True
61
62 if __name__ == "__main__":
63     show_start_screen()
64     game_active = True
65     while game_active:
66         for event in pygame.event.get():
67             if event.type == pygame.QUIT:
68                 sys.exit()
69
70             if event.type == pygame.KEYDOWN:
71                 if event.key == pygame.K_UP:
72                     direction = "UP"
73                 elif event.key == pygame.K_DOWN:
74                     direction = "DOWN"
75                 elif event.key == pygame.K_LEFT:
76                     direction = "LEFT"
77                 elif event.key == pygame.K_RIGHT:
78                     direction = "RIGHT"
79
80         # Mover la serpiente
81         x, y = snake[-1]
82         x += direction[0] * CELL_SIZE
83         y += direction[1] * CELL_SIZE
84         snake.append((x, y))
85
86         # Comprobar si se ha comido la comida
87         if x == food[0] and y == food[1]:
88             score += 1
89             food = generate_food()
90             snake.pop()
91
92         # Comprobar si se ha chocado contra las paredes
93         if x < 0 or x > WIDTH - CELL_SIZE or y < 0 or y > HEIGHT - CELL_SIZE:
94             game_active = False
95
96         # Dibujar la serpiente y la comida
97         for i in range(len(snake)):
98             pygame.draw.rect(screen, WHITE, (snake[i][0], snake[i][1], CELL_SIZE, CELL_SIZE))
99
100         pygame.draw.rect(screen, RED, (food[0], food[1], CELL_SIZE, CELL_SIZE))
101
102         # Mostrar el puntaje
103         text = font_text.render("Puntuación: " + str(score), True, WHITE)
104         screen.blit(text, (10, 10))
105
106         pygame.display.flip()
107         clock.tick(10)
```

CAPTURAS



CAPTURAS

```
File Edit Selection View Go Run Terminal Help
snake-game.py snake-game-basic.py X
D:\> Visual Studio Code > snake-game > snake-game-basic.py > ...
1 #Snake-Game-Basic
2
3 import os
4 import time
5
6 # Dimensiones del tablero
7 ancho, alto = 30, 12
8
9 # Posición inicial de la serpiente
10 serpiente = [(10, 15)]
11
12 # Dirección inicial de la serpiente
13 direccion = (0, 1) # Derecha
14
15 # Posición inicial de la comida
16 comida = (3, 3)
17
18 # Puntuación inicial
19 puntos = 0
20
21 # Función para dibujar el tablero
22 def dibujar():
23     # Limpiar la pantalla
24     os.system('cls' if os.name == 'nt' else 'clear')
25     # Dibujar el tablero fila por fila
26     for y in range(alto):
27         for x in range(ancho):
28             if (y, x) in serpiente: # Dibujar la serpiente
29                 print('O', end='')
30             elif (y, x) == comida: # Dibujar la comida
31                 print('X', end='')
32             else: # Espacios vacíos
33                 print('.', end='')
34         print()
35     # Mostrar la puntuación actual
36     print(f"Puntos: {puntos}")
37
38 # Función para mover la serpiente
39 def mover():
40     global comida, puntos
41     # Calcula la nueva posición de la cabeza de la serpiente
42     cabeza = (serpiente[0][0] + direccion[0], serpiente[0][1] + direccion[1])
43     if cabeza == comida: # Si la cabeza de la serpiente alcanza la comida
44         comida = (cabeza[0] + 2, cabeza[1] + 2) # Genera nueva posición de la comida
45         puntos += 1 # Incrementa la puntuación
46     else:
47         serpiente.pop() # Elimina el último segmento de la cola si no comió
48     # Verificar colisión (paredes o consigo misma)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
.....
.....
.....X.....
.....
.....
.....
.....
.....
.....
.....O.....
.....
Puntos: 0
Mover (w/a/s/d):
```

An illustration of a man with dark hair, wearing a red t-shirt and a lanyard with an ID badge, sitting at a dark brown desk and working on a light blue laptop. The background is a light cream color with several decorative circles: a small orange circle with diagonal lines in the top left, a medium purple circle with diagonal lines in the top center, and a large orange circle with diagonal lines in the top right.

CONCLUSION

El proyecto permitió desarrollar un juego funcional aplicando lógica de programación, estructuras de datos y algoritmos. Aprendí a dividir el código en funciones y a crear un flujo lógico usando diagramas.



LINKS



GitHub



Canva

Video Presentación



Video Explicativo

Video Demostrativo

**GRACIAS POR SU
ATENCION. ESPERO QUE
ESTE PROYECTO CUMPLA
CON LAS EXPECTATIVAS.**

