

# Desarrollo del Snake Game usando Pygame

Naydelin Hernández Vargas

Julio 2024

## 1. Introducción

El juego de Snake, o serpiente, es uno de los juegos más icónicos y simples en la historia de los videojuegos.

En 1976, Gremlin Industries desarrolló un juego llamado Blockade, la dinámica de este consistía en la competencia de dos participantes, donde se debía manejar una flecha utilizando cuatro botones de dirección para crear una ruta firme. La meta era sobrevivir al crear la estela más larga sin colisionar con los bordes o con las estelas del oponente. Si un jugador lograba seis victorias de bloqueo, el juego terminaba. (Ruiz Alcántara, 2019)

A partir de Blockade surge la inspiración para crear 'Snake Game', el cual consiste en un juego de un solo jugador que debe controlar la dirección de una serpiente en constante movimiento. El jugador necesita hacer que la serpiente se alimente de trozos de comida que van apareciendo en la pantalla, lo que hará que su tamaño aumente. A medida que la serpiente crece, la dificultad del juego se incrementa.

Snake Game volvió a tomar gran relevancia años después en parte por la implementación de este, a partir de 1998, en los telefonos Nokia. (Ruiz Alcántara, 2019)

Este proyecto tiene como objetivo recrear el clásico juego de Snake utilizando la biblioteca Pygame en Python y a su vez crear una versión personalizada del juego ya existente, ofreciendo una oportunidad de aprender y aplicar conceptos fundamentales de programación y diseño de videojuegos.

## 2. Metodología

La implementación del Snake Game se ha realizado utilizando la biblioteca Pygame, diseñada específicamente para el desarrollo de videojuegos en Python (Zorrilla, 2000)

A continuación, se describen las ideas principales del código y la estructura general del programa:

Si desea ver el código en detalle, puede acceder al repositorio en GitHub buscando el siguiente enlace: <https://github.com/nay0221/Prueba.git>

### 2.1. Librerías Usadas

Para el desarrollo del proyecto se ha utilizado principalmente Pygame, la cual facilita la creación de gráficos y la gestión de eventos. Además, se han usado otras librerías estándar de Python como time y random.

### 2.2. Estructura del Programa

El programa se estructura en varias secciones clave:

- Inicialización de Pygame y configuración de la ventana de juego.
- Definición de la serpiente y su movimiento.
- Generación aleatoria de comida para la serpiente.
- Detección de colisiones y condiciones de fin de juego.
- Bucle principal del juego, donde se gestionan los eventos y se actualiza la pantalla.

## 3. Desarrollo del Código

### 3.1. Inicialización y Configuración

La primera parte del código se encarga de inicializar Pygame y configurar la ventana del juego, definiendo el tamaño de la pantalla y los colores que se usarán. Se toma de (Python, 2023) el código base, donde se implementarán las nuevas ideas y modificaciones necesarias.

### **3.2. Definición de la Serpiente**

En esta versión del juego se implementan dos serpientes, cada una controlada por un jugador diferente. Las serpientes se representan como listas de segmentos, donde cada segmento es una posición  $(x, y)$  en la pantalla. El movimiento de las serpientes se maneja actualizando las posiciones de estos segmentos de acuerdo con las entradas de los jugadores.

### **3.3. Generación de Comida**

La posición de la comida se genera de manera aleatoria en la pantalla. Cada vez que la serpiente come una manzana, su longitud aumenta y se genera una nueva pieza de comida en una posición diferente. Para mejorar la estética del juego se opta por cargar imágenes de manzanas, las cuales fueron cargadas gracias a la función : `pygame.image.load("filename")`. El jugador 1, crece solo si come manzanas rojas y el jugador 2 solo si come manzanas verdes.

### **3.4. Detección de Colisiones**

La detección de colisiones es una parte fundamental del juego, ya que determina cuándo termina el juego y si una serpiente ha chocado con un obstáculo o consigo misma. En esta versión del juego, se implementan varios tipos de detección de colisiones:

Cada vez que una serpiente se mueve, se verifica si la nueva posición de su cabeza está fuera de los límites de la pantalla. Si la cabeza de la serpiente está fuera de estos límites, se considera una colisión con la pared y el juego termina para esa serpiente. Además, para detectar si una serpiente ha chocado consigo misma, se compara la posición de su cabeza con las posiciones de todos sus segmentos corporales. Si la posición de la cabeza coincide con la de algún segmento del cuerpo, se considera una colisión y el juego termina.

### **3.5. Bucle Principal del Juego**

El bucle principal del juego maneja la entrada del usuario (control de la serpiente), actualiza la posición de la serpiente, verifica colisiones y actualiza la pantalla.

## 4. Resultados

El juego resultante es una versión funcional del clásico Snake Game. A continuación se presenta una capturas de pantalla del juego en acción.



Figura 1: Captura de pantalla del juego Snake en acción.

Durante las pruebas, se ha verificado que el juego responde adecuadamente a los controles del usuario y que la detección de colisiones funciona correctamente.

## 5. Recomendaciones y Mejoras Futuras

Algunas posibles mejoras y nuevas características que se podrían agregar incluyen:

- Implementar niveles de dificultad ajustables.
- Incorporar gráficos y animaciones más avanzadas.
- Incorporar gráficos y animaciones más avanzadas.
- Mostrar exactamente el jugador que ganó y añadir una tabla de puntuaciones para registrar las mejores marcas de los jugadores.

## 6. Conclusión

Este proyecto ha permitido la recreación del clásico juego de Snake utilizando Pygame, ofreciendo una experiencia práctica en la programación de videojuegos. A través del desarrollo de este juego, se han aplicado conceptos importantes de programación orientada a objetos, manejo de eventos y diseño de juegos, resultando en un producto funcional y entretenido, en cual se logró implementar aportes personalizados.

## Referencias

- Python, C. (2023). *Juego de la serpiente en python - copiar y pegar*.
- Ruiz Alcántara, M. d. M. (2019). Snake game.
- Zorrilla, G. A. (2000). Pygame 1.5.5 reference manual [Manual de software informático].