

VIRAL HUNT

- **Descripción**

El proyecto consiste en el desarrollo de un videojuego en Python en el que un jugador controla un nanobot que recorre distintas zonas del cuerpo humano, esquivando obstáculos, con el objetivo de eliminar virus y bacterias

- **Desarrollo**

Este proyecto consta de los siguientes elementos de desarrollo de software:

- 893 líneas de código.
- 10 módulos.
- 27 funciones.
- 24 recursos (imágenes, sonidos, fuentes)
- 3 ficheros de datos
- 3 excepciones
- Librerías:
 - Pygame, random, sys

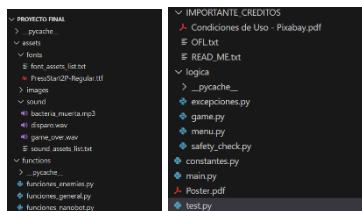


Figura 1. Estructura de módulos

- **Resultados**

Al ejecutar el programa se presenta un menú al usuario desde el cual el jugador haciendo clic puede iniciar la partida, acceder a las instrucciones o salir del juego.



Figura 1. Menú principal y menú de instrucciones

Pantalla que se muestra al terminar la partida y muestra la puntuación obtenida (score) y la distancia obtenida (distance). Además, permite al jugador iniciar una partida nueva.



Figura 2. Game Over

Los errores se muestran en la consola del usuario y se impide la ejecución del programa.

```
Pygame 2.6.1 (SDL 2.28.4, Python 3.13.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
No se ha podido encontrar la imagen: Nanobot.png
El programa no se puede ejecutar debido a que no tiene todos recursos debidos
```

Figura 3. Muestra de errores al usuario

Durante la partida se actualiza de forma continua y se muestra en pantalla la puntuación obtenida y la distancia recorrida.



Figura 4. Sistema de puntuación y progreso

El videojuego incluye niveles ambientados en el cerebro, el estómago, los pulmones y el torrente sanguíneo, los cuales se actualizan en función de la puntuación y la distancia recorrida. Cada uno cuenta con enemigos y obstáculos propios.



Figura 5. Sistema de niveles y enemigos

El juego genera proyectiles cuando el jugador dispara. Si una bala colisiona con un enemigo este se elimina y se incrementa la puntuación, mientras que una colisión del nanobot con las paredes, un obstáculo o un enemigo finaliza la partida.

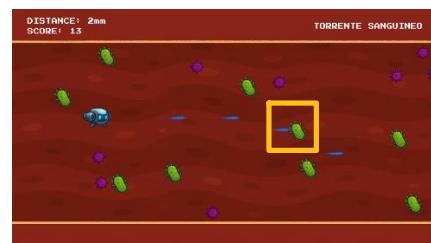


Figura 6. Sistema de colisión y generación de proyectiles