

# **Introducción a los videojuegos con Pygame**

Nuestro objetivo es resolver el desafío de crear un videojuego interactivo desde cero, utilizando Programación Orientada a Objetos (POO) y conceptos básicos de diseño de juegos.

La problemática central es cómo implementar un juego 2D simple, que incluya:

- Movimiento del jugador.
- Disparos y colisiones.
- Enemigos en pantalla.
- Interacción en tiempo real con el teclado.

## **Tecnologías utilizadas:**

- Visual Studio Code: Editor de código
- Python 3: Lenguaje de programación elegido
- Pygame: Librería utilizada para crear el videojuego

## **¿Qué es un Videojuego?**

Un videojuego es una aplicación interactiva orientada al entretenimiento que ejecuta una serie de reglas en tiempo real y permite la interacción entre un jugador (o varios) y un entorno gráfico, generalmente visualizado en una pantalla.

## **¿Qué es un Pygame?**

**Pygame** es una biblioteca libre y multiplataforma escrita en **Python** que se utiliza para desarrollar videojuegos. Proporciona funcionalidades para controlar gráficos, sonidos y entradas del teclado o mouse.

**(Leer documentación de Pygame)**

## **Modelar el juego:**

Lo primero es pensar la estructura del juego. Para esto hay que responder las siguientes preguntas:

### **1)¿De qué género será el juego?**

En nuestro caso queremos hacer un videojuego **shooter**. Más concretamente **Shoot 'em up** como Space invaders por ejemplo.

### **2)¿Cómo será la Jugabilidad?**

La Jugabilidad será bastante simple:

El jugador podrá mover la nave y disparar a los enemigos para eliminarlos. Los enemigos vendrán en patrones aleatorios y se detectará cualquier colisión entre los disparos y el enemigo o entre los enemigos y el jugador. Si algo golpea al jugador perderá una cantidad de vida

### **3)¿Qué objetos importantes va a tener el Videojuego y Cómo interactúan entre sí?**

En nuestro caso:

- **Jugador:** Nave que se mueve en las 4 direcciones(←↑↓→)
- **Enemigo:** Estos vendrán de arriba hacia abajo en patrones aleatorios

- **Disparos:** El ataque de la Nave para golpear enemigos
- **Pantalla:** Espacio para mostrar las interacción del juego

#### 4)¿Qué lógica tendrá mi juego?

- Un bucle principal que actualiza la pantalla 60 veces por segundo.
- Eventos de teclado para mover al jugador y disparar.
- Detección de colisiones con `colliderect()` para saber si el disparo acierta.
- Reaparición del enemigo cuando es destruido en un nuevo patrón aleatorio