

En este proyecto inspirado en el juego del dinosaurio de Google, hemos aplicado los conceptos de Singleton y Object Pooling para mejorar la eficiencia y la organización del código.

1. Singleton Pattern:

El patrón Singleton se utilizó en el script ObjectPool, que maneja la creación y gestión de los objetos reutilizables, como los obstáculos y el suelo. Este patrón garantiza que solo exista una instancia de la clase ObjectPool en toda la ejecución del juego, facilitando el acceso global y evitando que se creen múltiples instancias innecesarias. En este caso, el Singleton asegura que la gestión del ciclo de vida de los objetos sea centralizada, lo que mejora el rendimiento y la consistencia del sistema. Esto es especialmente útil para objetos que se crean y destruyen continuamente, como los obstáculos.

2. Object Pooling:

El Object Pooling se implementó para optimizar el manejo de los obstáculos y el suelo, que se generan de manera repetitiva. En lugar de instanciar y destruir objetos constantemente, lo que puede ser costoso en términos de rendimiento, se crea un grupo o "piscina" de objetos que se reutilizan.

Cuando un obstáculo sale de la pantalla, en lugar de destruirlo, simplemente se desactiva y se coloca de nuevo en la piscina de objetos. Cuando es necesario generar un nuevo obstáculo, el ObjectPool lo activa y lo reposiciona en la escena. Esto reduce el consumo de memoria y mejora el rendimiento general del juego.

Beneficios:

- **Mejora de rendimiento:** Al evitar la creación y destrucción constante de objetos, reducimos el uso innecesario de recursos y las pausas que podrían ocurrir por la recolección de basura (Garbage Collection).
- **Organización:** El uso de Singleton facilita la organización del código, ya que centraliza la gestión de los objetos, haciéndolo accesible desde cualquier otra parte del código sin la necesidad de múltiples referencias.