Práctica 3: Patrones de diseño

Objetivo: Hacer uso de patrones de diseño para resolver el problema que nos propone el cliente.

Descripción: Estamos desarrollando un videojuego y nuestro cliente nos ha pedido añadir una nueva funcionalidad:

Estimado desarrollador,

Te escribo con relación a la última reunión que tuvimos con el equipo artístico, hemos decidido añadir obstáculos dinámicos. Una idea fantástica de nuestro equipo que le añadirá valor a nuestro videojuego.

Nuestra idea es que mientras el jugador está realizando un trayecto con el taxi, vayan apareciendo obstáculos, lo cual añadirá una jugabilidad apabullante para el jugador, generándole una sensación continua de tensión por las trampas que puedan aparecer.

Por ahora hemos considerado los siguientes obstáculos, aunque queremos tener la posibilidad de añadir nuevos en el futuro:

- *Coche de policía. Queremos que aparezcan nuevos coches de policía ya que consideramos que es demasiado fácil escapar de ellos.*
- *Valla de obra. Estas vallas de obra pueden aparecer en cualquier momento, así el taxi lo tendrá complicado para no chocar con una de ellas y así perder vida fácilmente.*
- *Debilitador. A modo de moneda que en caso de que el Taxi la coja, este perderá velocidad, perdiendo tiempo y teniéndolo más difícil para escapar de la policía.*

Para equilibrar la mecánica del juego, cuando haya una interacción con los obstáculos, los efectos de cada obstáculo tendrán efecto sobre el Taxi durante un tiempo específico.

Miguel A.P.

CTO of TaxiDrive the game ©

Requisitos:

1. Los obstáculos comparten los siguientes atributos:
    - Puede o no perseguir al Taxi.
    - Es o no un objeto sólido.
    - Puntos de vida que quita al Taxi en caso de colisión
        - Coche de policía: -30
        - Valla: -10
        - Debilitador: 0
    - Multiplicador de velocidad del Taxi en caso de colisión

        - Coche de policía: x0.8
        - Valla: x0.8
        - Debilitador: x0.5
    - Tiempo durante el que hace efecto el multiplicador de velocidad:
        - Coche de policía: 1 sec
        - Valla: 1 sec
        - Debilitador: 30 sec
2. Los obstáculos deben surgir efecto sobre el Taxi.
3. Existencia de **un único Taxi**, con los siguientes atributos:
    - Vida, con un valor inicial de 100.
    - Velocidad, con un valor inicial de 1.

    Para comprobar el efecto de los obstáculos sobre el Taxi, cada vez que alguno de sus atributos cambie de valor, se imprimirá por pantalla un mensaje con los valores de **vida y velocidad** (para realizar esto también se necesitarán los atributos *lastLifeValue* y *lastSpeedValue*)**. Únicamente está permitido imprimir por pantalla desde Taxi** (a excepción del mensaje inicial del punto 4)**.**
4. Realizar un programa de consola con un buble infinito, cada bucle se ejecutará cada 20 ms sin bloquear la ejecución del programa. Dentro de este bucle, podremos crear obstáculos al pulsar una tecla del teclado de la siguiente manera:

```
Press a key to create an obstacle:
    A -> PoliceCar
    S -> ConstructionFence
    D -> SpeedDebuff
```

5. El contador de tiempo debe estar en el buble principal.
6. Si hace falta utilizar el patrón de diseño *Singleton*, implementar como una clase genérica.
7. Prestar especial atención de para **cumplir con los principios SOLID y los principios de diseño software**.

Información:

En refactoring.guru podréis entender y obtener un punto de partida para poder implementar patrones de diseño.