

# DOCUMENTACIÓN HERO CITY

Hero city es un juego en el que eres un superhéroe que vive en una ciudad en la que aparecen villanos y tienes que ir a combatir con ellos. El juego es automático así que una vez introducida la dificultad, elegido el héroe e introducidos los días que se desean jugar el juego empezará a ejecutarse y al pasar los días finalizará automáticamente.

Para hacerlo el juego tiene diferentes clases:

- **TimeManager:**  
Esta clase contiene los datos que tienen que ver con el flujo del juego. Tiene una función que devuelve true o false se llama al final de cada frame para saber si hay que seguir jugando o se ha llegado ya al último día de juego, también gestiona el día, el día de la semana y la hora actual.
- **Game:**  
Esta clase gestiona todo el bucle y el flujo del juego. Para hacerlo tiene diferentes variables y funciones.  
Contiene los diferentes superhéroes que se pueden ser, una instancia de la clase World, una bool para saber si el juego tiene que seguir ejecutándose una instancia del time para coger los datos del tiempo, una variable int que hace referencia al nivel de dificultad. Funciones:
  - **Update:** Crea un bucle y ejecuta la función RunGame tantas veces por segundo como este especificado en la instancia del Timemanager.
  - **RunGame:** Controla cada frame del juego segun el estado del superHeroe, Se encarga de Spawnear enemigos cuando deba hacerlo, gestionar los combates y algunos prints en pantalla.
  - **TrySpawn** es una funcion que devuelve true o false, y depende de lo que devuelva el Update hará un print u otro. Gestiona que enemigo se spawneara y algunas estadísticas de vitalidad y fuerza de cada uno.
- **World:**  
Esta case esta contenida dentro de la clase game. Contiene un puntero a la clase superhéroe y otro a la clase villano y una bool de si existe un villano.
- **State:**  
Es una Interfaz con las funciones OnEnter, Update y OnExit.
- **StateMachine:**  
Esta clase tiene un instancia de un enum GameState.
- **Character:**  
Es la clase abstracta de donde heredaran todos los villanos y superhéroes.

Contiene variables que hacen referencia a la fuerza, la vida, la vida máxima, al nombre, al estado y a la posición.

Tiene varias funciones virtuales para atacar, recibir daño y las funciones que hereda de State.

- **SuperHero:**  
Es una clase abstracta que hereda de Character y tiene las variables que hacen referencia a la capacidad de regeneracion, una booleana que hace referencia a si ha perdido un combate, la posicion de su casa y la posicion a la que debera dirigirse cuando aparezca un enemigo. Tiene una función virtual OnWin que se llama cada vez que vence a un villano.
- **HealthyHero:**  
Hereda de Superhero y tiene una variable shield que cuando recibe daño se gasta antes que la vida, cada vez que gana una pelea recupera una cantidad de escudo.
- **LastHitHero:**  
Hereda de SuperHero y contiene una bool para saber si ya ha usado su habilidad. Cuando le atacan y le dejan a 0 vida, la primera vez ataca al villano y se queda a uno de vida.
- **LethalHero:**  
Hereda de SuperHero y tiene un int que cuenta ataques y cada dos hace un doble ataque y cada tres hace un ataque un poco menos potente.
- **Villain:**  
Hereda de Character y tiene una funcion para setear sus habilidades cada vez que spawnear. Hace override de las funciones ataque y daño, esta clase hace de villano básico cuando se instancia.
- **HealthyVillain:**  
Herda de Villain y hace override de a funcion ataque, cada vez que ataca recupera vida.
- **IncrementalForceVillain:**  
Hereda de Villain y cada vez que recibe daño su fuerza incrementa.
- **MultiAttackVillain:**  
Hereda de Villain y tiene un contador de ataques, en su función ataque cada 3 ataques hace un ataque extra.

Creo que este ejercicio va genial para aprender C++, he aprendido mucho sobre el manejo de clases y punteros. En verano cuando tenga mas tiempo convertiré el mundo en un lugar 2D y mecánicas para que el combate no sea automático.