# Introducción a la clase

Ejercicio: lampara con un switch.

Al tocar el switch se enciende la lámpara. Ambos componentes en la misma blueprint

Ejercicio 2: tengo dos actores y ambos tienen que interactuar entre sí. Visir con el trigger. Cuando el player vaya al trigger queremos cambiar la variable de estado del visir.

Queremos entonces comunicar a dos actores, se necesita **COMUNICACIÓN**.

Hoy vamos a hacer el SuperMario64.

* **Contador de monedas (coin)**: queremos imprimir por pantalla cuantas tenemos
* **PointManager**: objeto del mundo y cuando las monedas cuando se cogen avisan al PointManager que es quien tiene el contador.

Coin con cilindro y en el onclick que rote

Controlador:

* blueprint que no se ve. Es como el player Start (es algo logico)
* Vamos a hacer el CoinController. Creamos la blueprint y le añadimos una variable
* Control (1) y hay Coins (N).
  + El coin NO ve la variable contador (le da igual)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEs un private Esto es un public

Captura de pantalla de un celular

Descripción generada automáticamente

Podemos hacerlo de dos formas:

O que Coin sea miembro de CoinController y la otra es un paso de mensajes.

* **Primera forma**: cada coin tendrá un coinController.

Interfaz de usuario gráfica

Descripción generada automáticamente Arrastro la variable, hago un get, llamo a la función INC que incrementa ubno. Y acabo haciendo un set.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* **Segunda forma**: d

getActor, permite coger una escena y coger el actor de ahí. Nos devuelve el primer Actor que encuentra, (vamos a llamar a CoinController) como solo hay uno, pues nos devuelve el que hay.

Vamos a borrar la variable

Vamos a hacer una cosa intermedia. Usar el getActorOfClass al inicializar el objeto. En el primer frame solo.

Ahora, realmente en vez de tener este contador en este actor, podemos usar el Player.

En la carpeta Content

* Characters: mallas
* ThirdPerson:
  + Blueprints: hay 2 blueprints muy importantes que vamos a tocar
    - Character: yo, en el juego
    - Gamemode: reglas de la sesión.
  + Maps

Ahora vamos a recoser el código, vamos a ventilarnos el objeto para que el counter pase a ser una variable del Player.

**Ejercicio para nosotros**:

Player: debe tener 2 variables

* **Counter**: va sumando con las monedas
* **Life**: va restando con unas bombas. Empieza a 100 y cada bomba resta 25 de vida.

Creamos la blueprint bomba, que tenga un indicador de quien es el player y cada vez que se destruyan las bombas se reste vida.

Para poner la vida a 100.

* Cogemos la variable en la derecha pone las propiedas, en donde dice default le ponemos el 100

# Viewport, Widget y

Es la pantalla

Aquí puedo poner pegatinas (elementos de interface). Estos elementos son los widget

Vamos a poner un LIFE = x

Boton drcho. > user interface > widget blueptint > User widget

El widget es una composición de elementos. Todoo es un canvas

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Ahora vamos a

Content > thir person > blueoprints > gamemode (para las cosas de la sesiojn y volcarlo en Canvas)

Queremos volcar el widget al viewportr (en la sesion)

Creamos evento, y creamos el widget score

Y luego al add to viewport

Una pantalla de una computadora

Descripción generada automáticamente con confianza baja

Binding (coser/enganchar): coge una variable y la engancha al widget, para que no haya un contenido estático.

En vez de text pogress bar

El bindgin va en el percent

Como hacer el float / int

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego hacemos el binding también con el color Fill Color de la barra, de manera que si es mayor que 50 devolvemos verde, si no rojo

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja