

# Puerta cooperativa

## Introducción

En esta actividad expondremos cómo, partiendo del proyecto *MPShooter* trabajado en clase, se ha creado un Blueprint de una puerta “cooperativa” que se abre cuando más de un jugador se acerca a la misma.

## Blueprint de puerta

Lo primero que haremos será crear un Blueprint para la puerta. Este BP incluirá un volumen de tipo trigger que detectará las colisiones con los jugadores, un *SceneComponent* que actuará como bisagra para la puerta y la mesh de la puerta.

A parte, crearemos un material para la puerta al que le podamos pasar un color, que será rojo cuando la puerta esté cerrada (o cerrándose) y verde cuando esté abierta (o abriéndose). Posteriormente, desde el *BeginPlay* de la puerta, crearemos una instancia de material dinámica hija de dicho material, se lo asignaremos a la mesh de la puerta y nos guardaremos la referencia.

En cuanto a variables, utilizaremos:

- **MinPlayersCountToOpen**: un integer de tipo read-only para definir la cantidad mínima de jugadores que hacen falta para abrir la puerta.
- **PlayersCount**: un integer para llevar la cuenta de los personajes que hay simultáneamente cerca de la puerta.
- **IsOpen**: un booleano que nos ayuda a saber si la puerta está abierta o cerrada y si hay que abrirla o cerrarla en el momento en que aparece o desaparece un jugador.
- **DoorMaterial**: referencia a la instancia de material dinámica que nos ayudará a modificar el color de la puerta.

Una vez todo configurado, sobreescribiremos los métodos *OnComponentBeginOverlap* y *OnComponentEndOverlap* del volumen de tipo trigger. Dentro de ellos, lo primero que haremos será comprobar que seamos autoridad y, en caso afirmativo, ajustar el contador de jugadores si es necesario. En caso de que la puerta esté cerrada y se den las condiciones para abrirla o viceversa, llamaremos a los métodos de tipo **Multicast**: *OpenDoor* y *CloseDoor*, respectivamente.

Lo que haremos dentro de los métodos que acabamos de mencionar será cambiar el color de la puerta para que sea verde cuando esté abierta y roja cuando esté cerrada; además de ajustar el yaw de la bisagra de la puerta entre 0 (cerrada) y 90 (abierta) mediante un Timeline.

## Enlace al repositorio

Este es el enlace al repositorio de nuestro proyecto, que incluye también una demo:

<https://github.com/FranMorenoULL/TATV-23/tree/main/MPShooter>