# Introducción

Vamos a ver un spawner (tamb por EQS) y también guardar partida

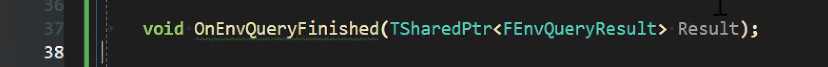
# Spawner

Trigger en el actor y cuando entre trigger lo activamos y cada x tiempo spawneamos. Con una EQS.

MCVGame > Actors > MCVSpawnerEQS.h

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Visual Assist. Click derecho generar implementacion para los métodos.

El cpp

Text

Description automatically generated

On begin overlap



On end overlap

Text

Description automatically generated

**Cuando se spawnea (LO UNICO NUEVO ESS ESTO. Crear la Request por código y lanzarla)**

Text

Description automatically generated

Cuando se realice la query se lanza el OnEvnQueryFinished

Text

Description automatically generated

Se puede hacer luego un Test de distancia para que no aparezcan cerca entre sí

A screenshot of a video game

Description automatically generated with medium confidence

O uno de overlap

Graphical user interface, application

Description automatically generated

Y le podemos poner que no colisione con Pawns en una caja de 40x40x100

Graphical user interface

Description automatically generated

El bool match tiene que estar en TRUE, para que no haya Pawns

# Guardar partida (Save Game) (me he perdido)

USaveGame: clase de la que heredamos para poder guardar la partida. Lo serializa a disco si ponemos las cosas en UPROPERTY

Usamos la GameInstance.

Funciones para gestionar

Text

Description automatically generated

Creamos /SaveData/MCV\_SaveGame.h

C++ game heredamos de SaveGame

Text

Description automatically generated

Cuando arranco el juego se crea la GameInstance por lo que es el lugar idóneo para instanciar un objeto de tipo SaveGame.

En el Player State, begin Play cojo la gameinstance y le añado la experiencia del SaveGame

En la GameInstance tenemos estas funciones

Text

Description automatically generated

Menciona no se que de Broadcast