**GameState**: info global del juego

**PlayerController**: hace de interfaz entre el input y el personaje. Un player controller puede controlar varios personajes.

Diagrama

Descripción generada automáticamente

**Pawn**: avioneta, coche, personaje

**Character**: personajes humanoides (heredan de pawn)

# Estados del juego (GameInstance/GameMode/GameState)

**GameInstance**: es nuestra clase Singleton

* Init: al iniciar el juego
* Shutdown: cuando se cierra

**Gamemode**: gestiona las reglas (al entrar un jugador, spawnea un personaje, o me espero a que estemos todos y luego spawneo a todos a la vez).

**GameState**: es parecido, está mas orientado a datos. En singleplayer lo podríamos meter todo en gamemode, pero es interesante usarlo bien. Podemos guardar puntuación o estado de los cofres si está abierto o cerrado. Cosas así las puede controlar el gameState.

* GameState datos relativos a la partida (cofres, puntuaciones de equipo)
* Gamemode lógica a mas bajo nivel (spawn de players)

Puntuación del equipo en un juego online por ejemplo

# Estados del jugador (PlayerState/PlayerController)

**PlayerState**: experiencia del jugador y cosas así.

**Player** **Controller**: mas de mecánicas, input,

Al cargar el nivel se van a perder datos. SI queremos conservarlos lo guardamos en gameinstance

# Class Specifiers

**Blueprintable: puedes heredar y modificarlo desde blueprint**

**BlueprintType**: puedes acceder desde Blueprint

## Properties Specifiers

**EditDefaultsOnly**: solo podemos editar en blueprints (si lo ponemos en el nivel no)

**EditAnywhere**: todos lados

# Levels

Texto

Descripción generada automáticamenteCreamos ULevel en la GameInstance

A screen shot of a computer program

Description automatically generated with low confidence

(MAL, no son ULevel SON UWORLD)

A screen shot of a computer program

Description automatically generated with low confidence

(Tenemos cargados los niveles, si pongo un TSoftObjectPtr no lo tengo siempre en memoria).

Creamos la clase

Interfaz de usuario gráfica

Descripción generada automáticamente

MyGameInstance y le hemos puesto las 2 referencias a niveles, por lo que en los detalles vemos esto.

Texto

Descripción generada automáticamente

Llamar siempre al Super a no ser que estemos seguros de que mejor no hacerlo.

## Load New Level

A picture containing screenshot, text, line, font

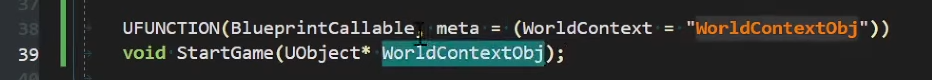
Description automatically generated

Whole Tomato Visual Assist

Visual Assits

## Meta

Le ponemos meta para que sepa que el parámetro que enviamos es un WorldContext ¿¿??



UFUNCTION sirve para que se pueda llamar desde Blueprint.

COLISIONES: HABLA A LAS 20:03 o así

2 pawns nunca harán overlap.

## LineTraces a partir del World

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

To bind Actions we can do it from here: SetUpPlayerInputComponent

Texto

Descripción generada automáticamente

## GameModes

We have

* **MCVGameGameMode:** game
* **MCVMenuGameMode**: we clone the MCVGameGameMode and we delete the code for spawning the player because we dont need it in the menu.

By default the gamemode creates a Pawn. But we want to create another gamemode for the menu.

Each one for each level.

**Windows > World Settings**: GameMode Override: we specify the gamemode of each Level (we change it manually in each level).

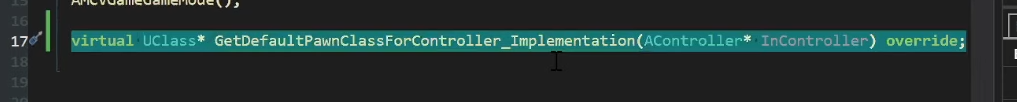
### Virtual functions Gamemode

#### GetDefaultPawnClassForController (gamemode):

We can add logic to override, so the first character is X and the second another one.

**BlueprintNativeEvent**:

* can be overriden in BP or C++ (Blueprintable only in BP)
* Needs a **GetDefaultPawnClassForController\_Implementation:** lo hace solo el preprocesador de unreal. (EN C++ el nombre necesita el \_Implementation)



Texto

Descripción generada automáticamente

This is a function we can override

#### HandleStartingNewPlayer

While the player is waiting for joining the room, we can handle it (with loading screen, cinematics or something with this function).



The implementation of the timer and function not even the professor knows:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente