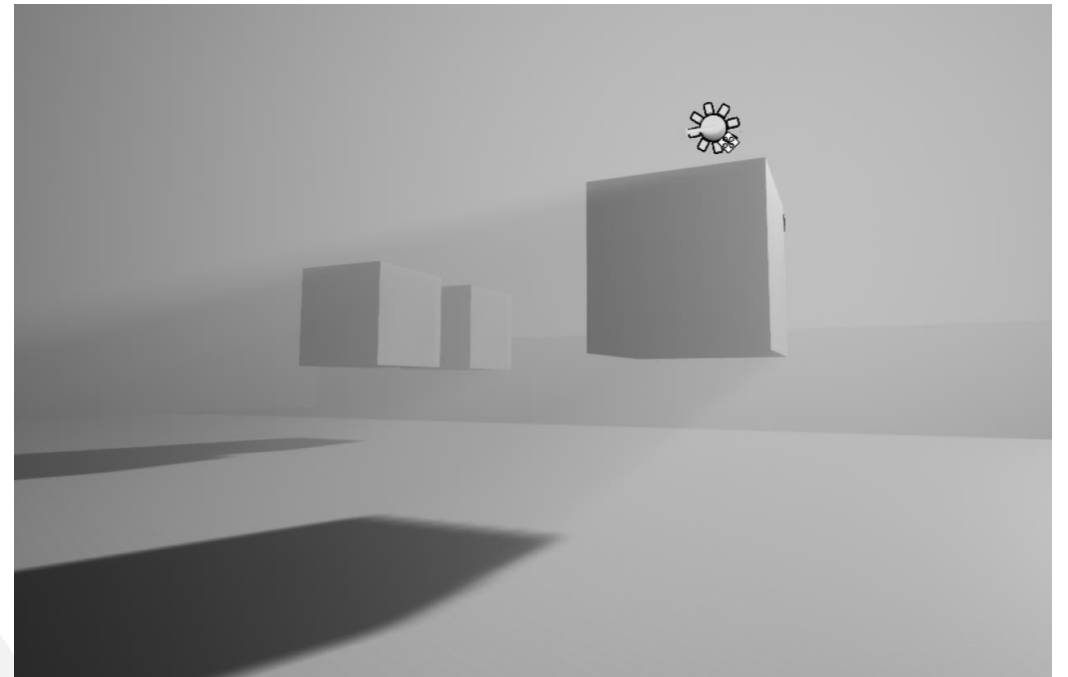


# Blueprints 1

Por David García y Arturo Escamilla

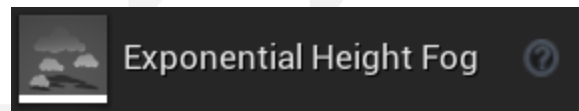
# Luces volumétricas

- |

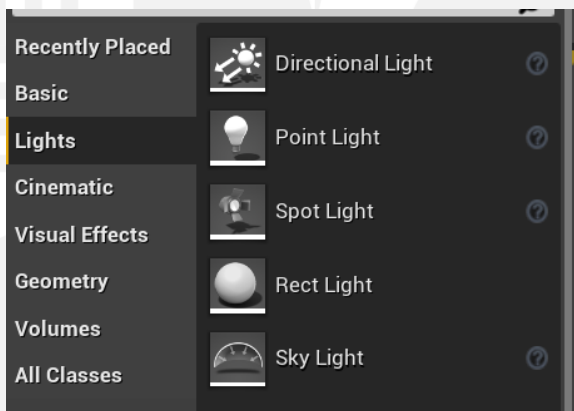


# Necesitamos...

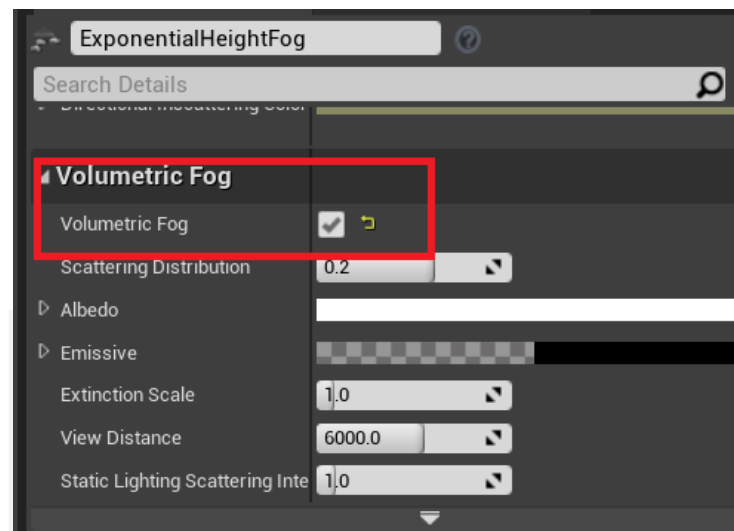
1.



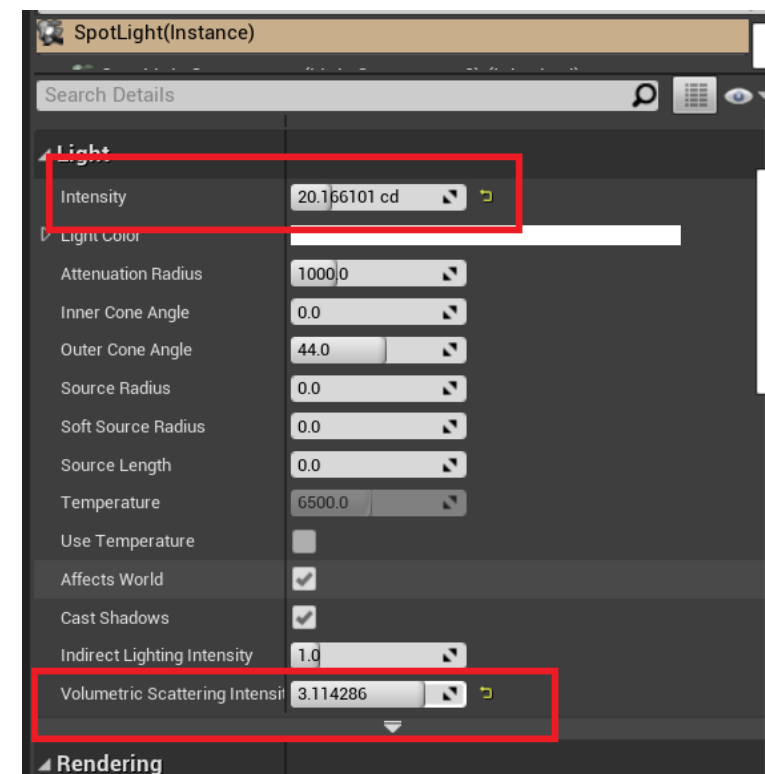
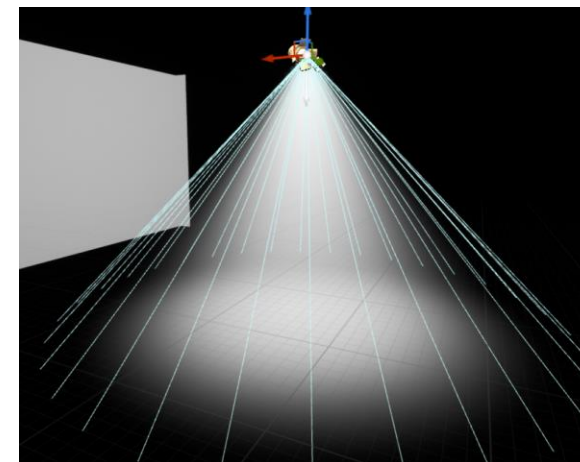
## Luces



2.



3.



# Agenda

- Introducción al GameMode
- Elementos del GameMode
- Jerarquía en Blueprints.
- Character Blueprint.
- Player Controller/IA Controller
- Repasando Inputs.

*Objetivo:*

*Entender conceptos basicos: "GameMode"  
Character, Pawn etc. Y ponerlos en práctica.*



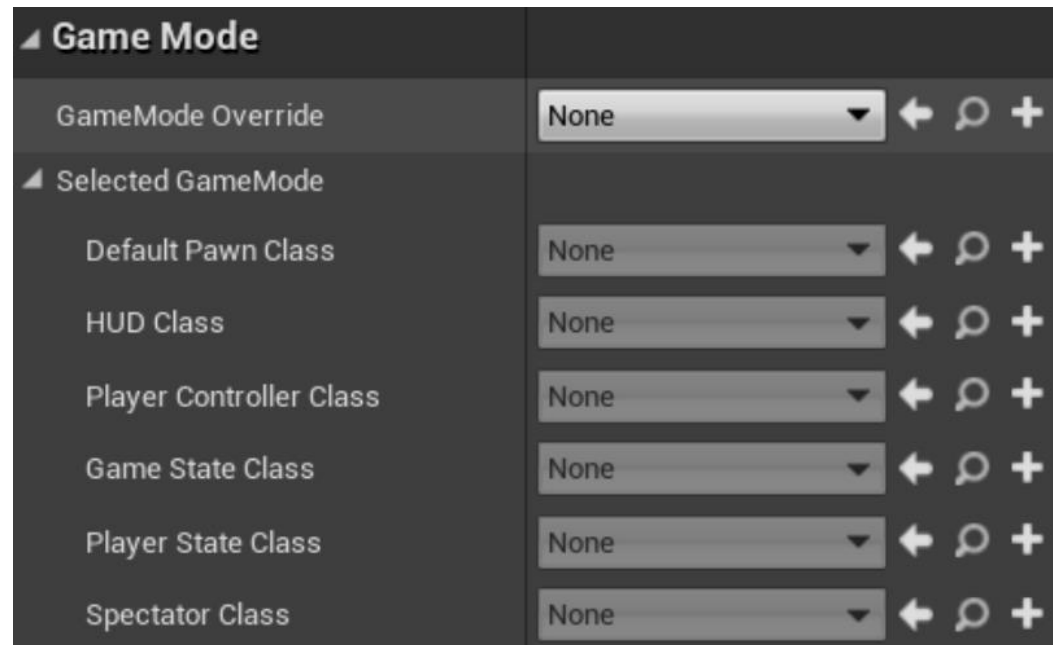
# Sección de preguntas



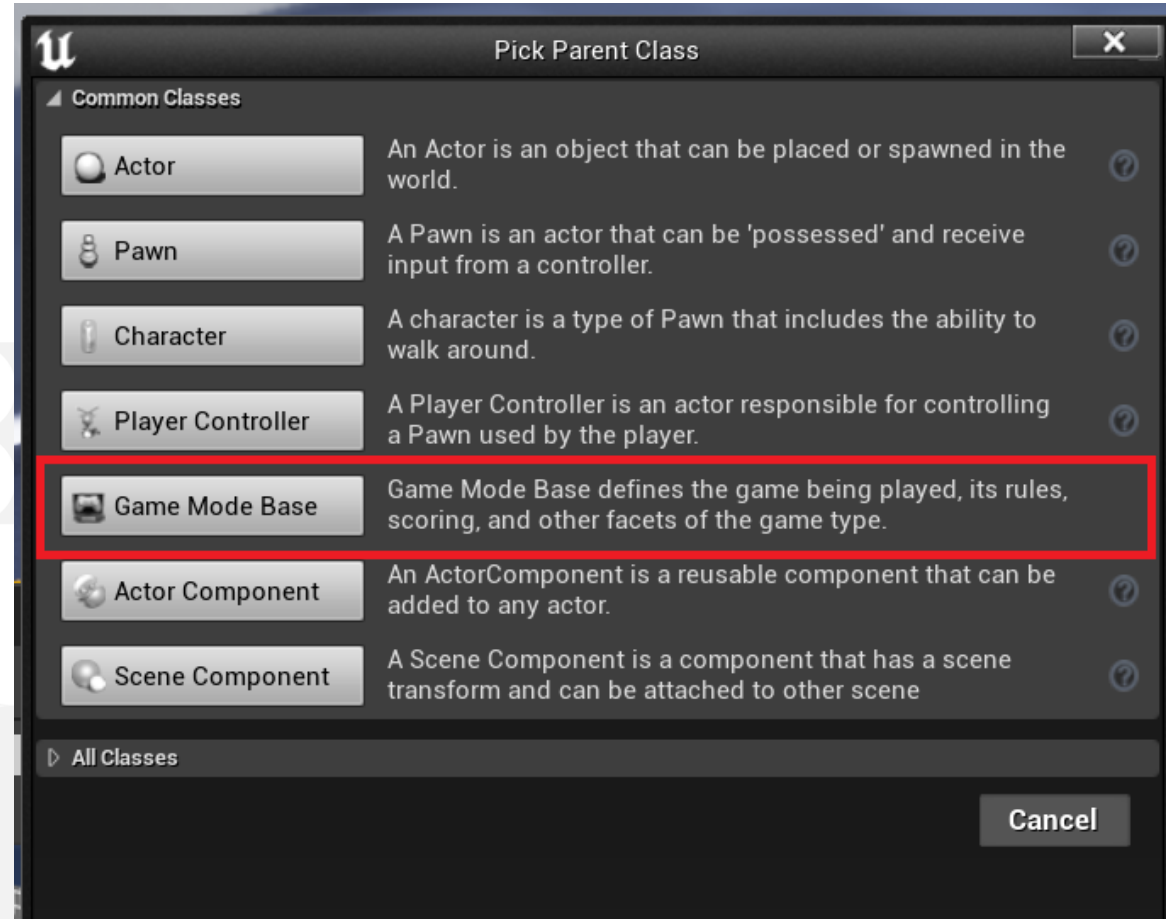
# Introducción al "GameMode"

Aquí definimos:

- El inicio del juego
- Las reglas del juego
- Puntajes
- Cantidad de jugadores



# Donde crearlo



File Edit Asset View Debug Window Help

Components

+ Add Component Search

Example2GM(self)

DefaultSceneRoot

My Blueprint

+ Add New Search

Graphs

- EventGraph
  - Event BeginPlay
  - Event Tick

Functions (30 Overridable)

- ConstructionScript

Macros

Variables

Components

- DefaultSceneRoot

Event Dispatchers

Compile Save Browse Find Hide Unrelated Class Settings Class Defaults Simulation Play No debug object selected Debug Filter

Viewport Construction Scrip Event Graph

Example2GM > Event Graph Zoom 1:1

Right-Click to Create New Nodes.

This node is disabled and will not be called. Drag off pins to build functionality.

Event BeginPlay

This node is disabled and will not be called. Drag off pins to build functionality.

Event Tick

Delta Seconds

BLUEPRINT

Compiler Results

Clear

Details

Search Details

Parent class: Game Mode Base

Actor Tick

- Start with Tick Enabled ☒
- Tick Interval (secs) 0.0
- Allow Tick Before Begin Play ☒

Classes

- Game Session Class GameSession
- Game State Class GameStateBase
- Player Controller Class PlayerController
- Player State Class PlayerState
- HUD Class HUD
- Default Pawn Class DefaultPawn
- Spectator Class SpectatorPawn
- Replay Spectator Player Controller PlayerController
- Server Stat Replicator Class ServerStatReplicator

Game

Default Player Name

Game Mode

- Use Seamless Travel ☐
- Start Players as Spectators ☐
- Pauseable ☒

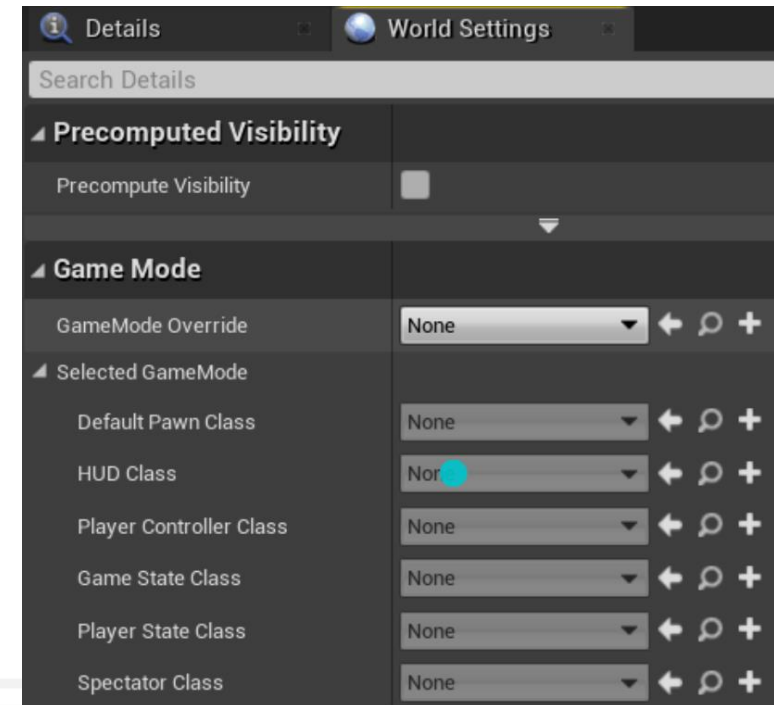
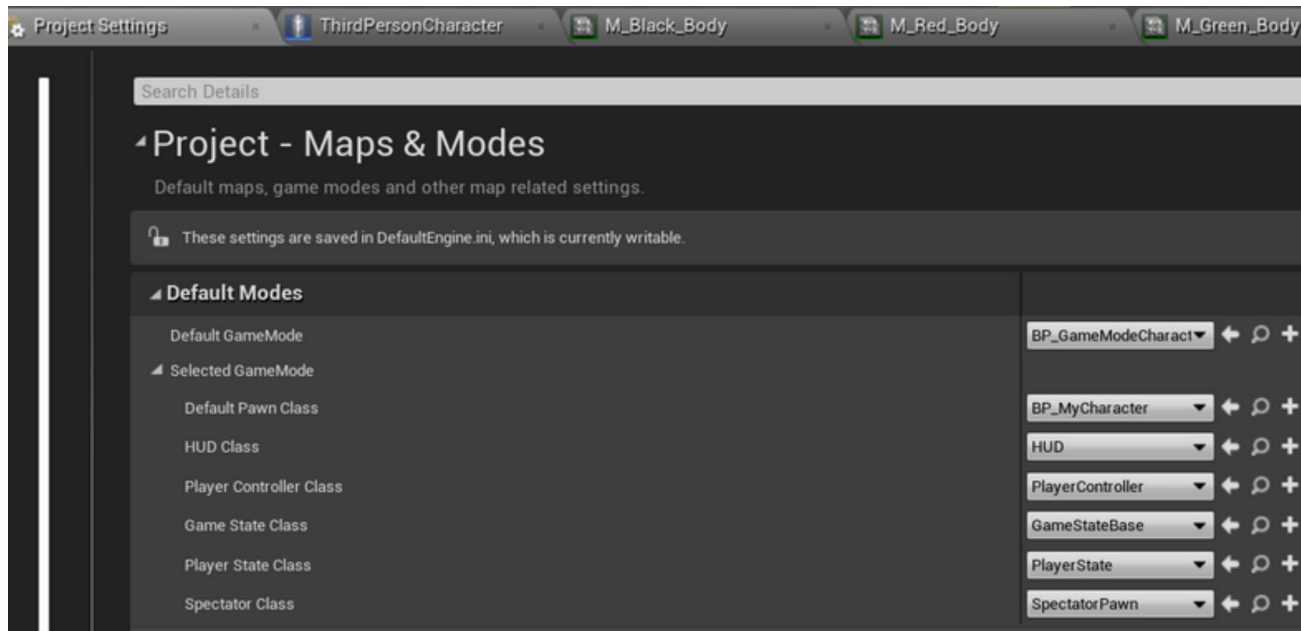
LOD

Cooking



# Lugar donde ponemos el GameMode

- Project settings y World Settings



- Sobrescribirá el de "project"

# Introducción a los elementos del GameMode



# Default Pawn

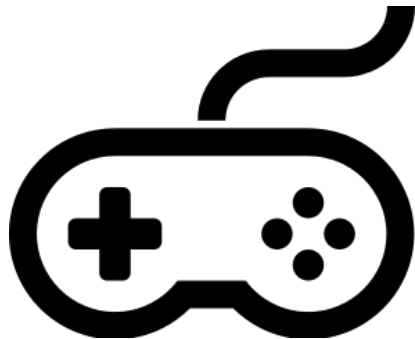
- Aquí colocamos el personaje u objeto que invocaremos



# Player Controller

Es el controlador designado al Pawn.

- Recibe las entradas (del dispositivo/ control/ teclado y mouse)



Posesión.

Player Controller

Pawn



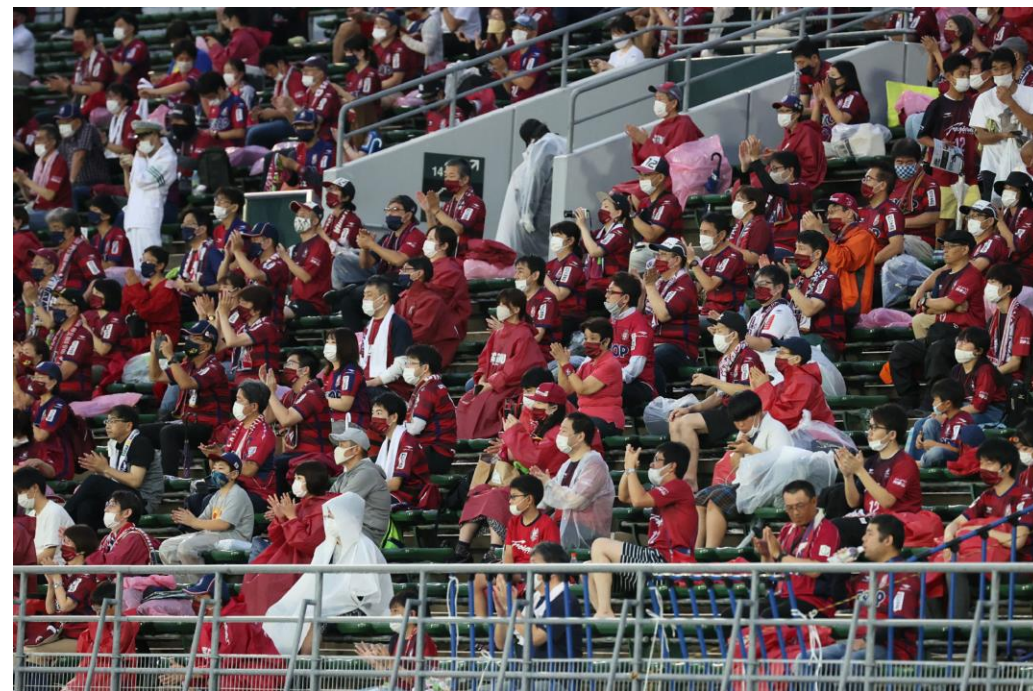
# HUD

- Head-Up Display (HUD).





# Spectator



# Game Session

## Game Session Class

Lo usamos para juegos online y donde definimos opciones ejemplo:

- Máximo de jugadores
- Máximo de espectadores

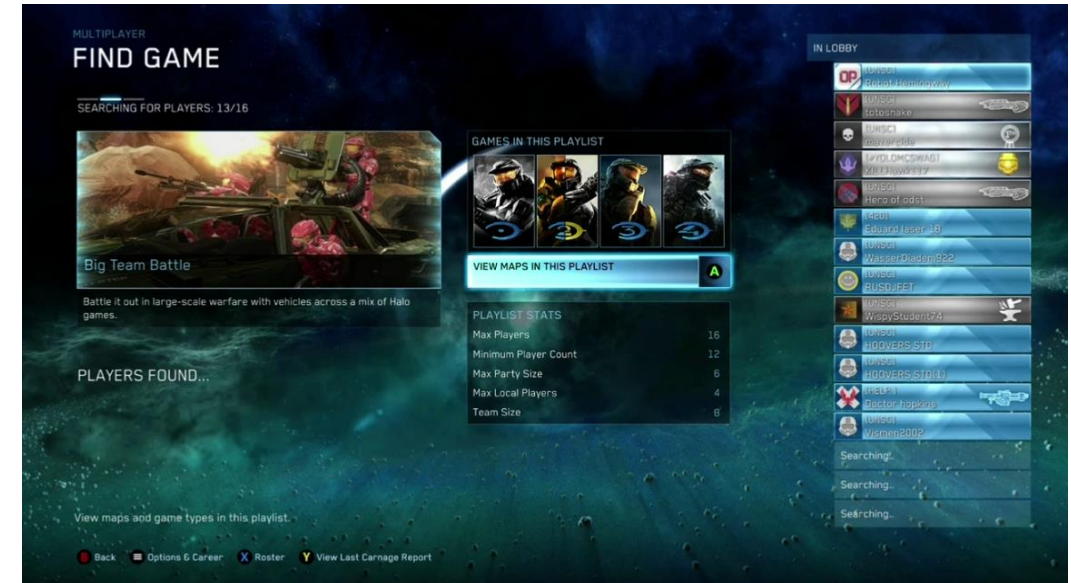




# Game state

Es el responsable de mostrar:

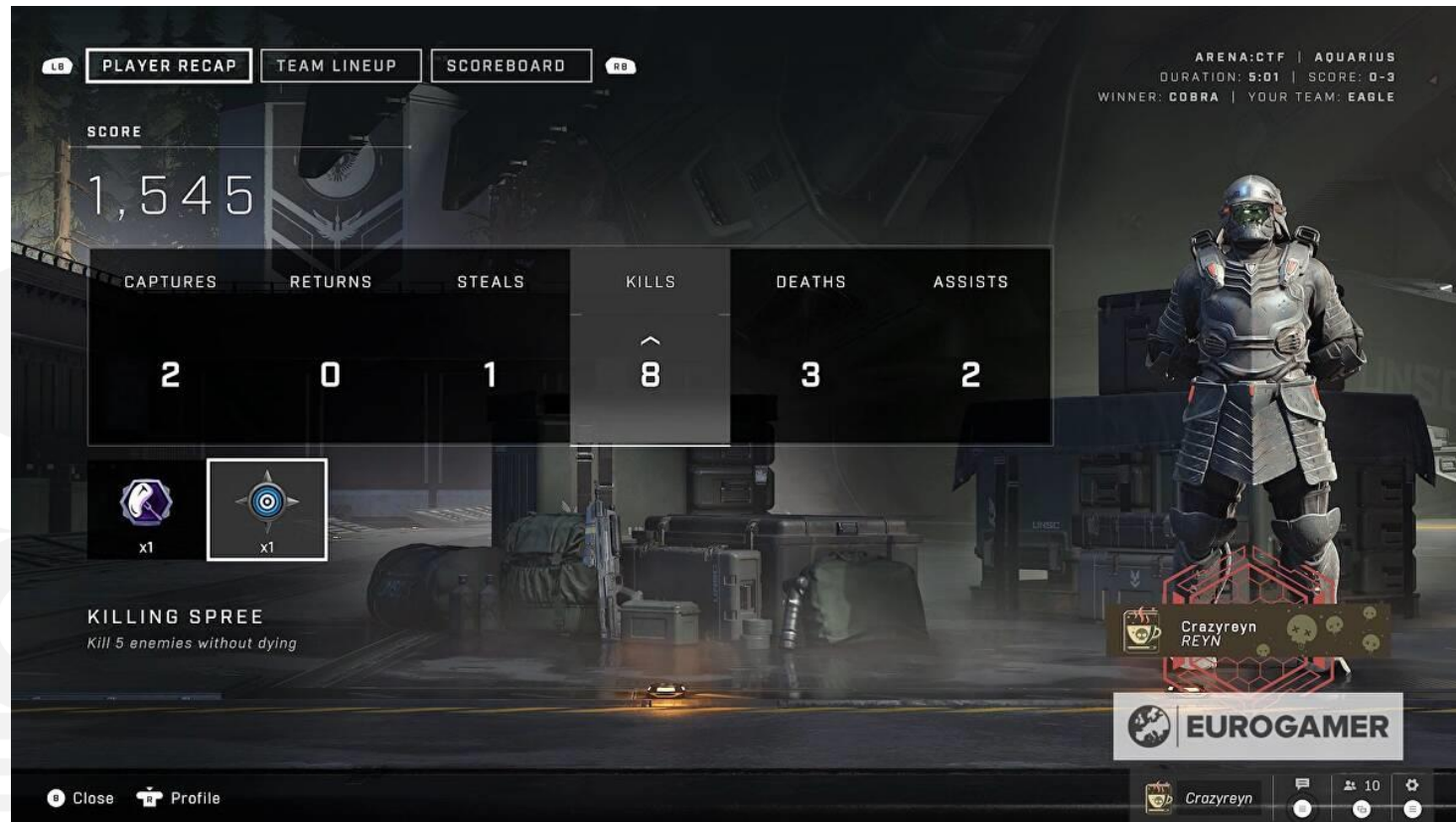
- La lista de jugadores conectados
- El puntaje del juego
- Las misiones que tienes que completar



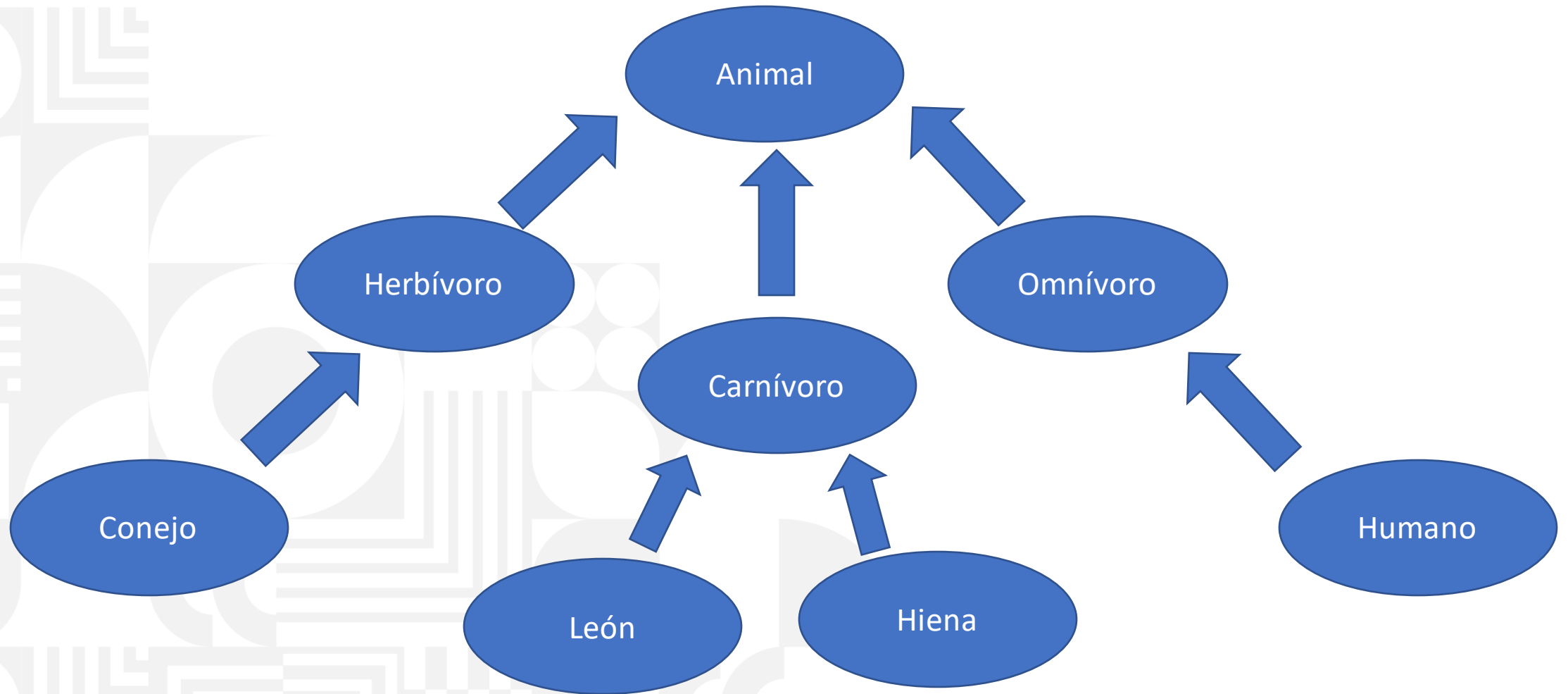


# Player State

El "Playerstate" es creado para cada jugador y es el lugar donde tenemos la información relevante del jugador (Nombre del jugador, puntajes).



# Concepto "Herencia"



# Jerarquía en Blueprints

Object > Actor > Pawn > Character > Nuestro Personaje



# Character Blueprint

- Versión extendida de un Pawn que incluye Capsule Component, Character Movement y Skeletal Mesh.
- Estos Pawns nos permiten agregar personajes con animaciones.



Character

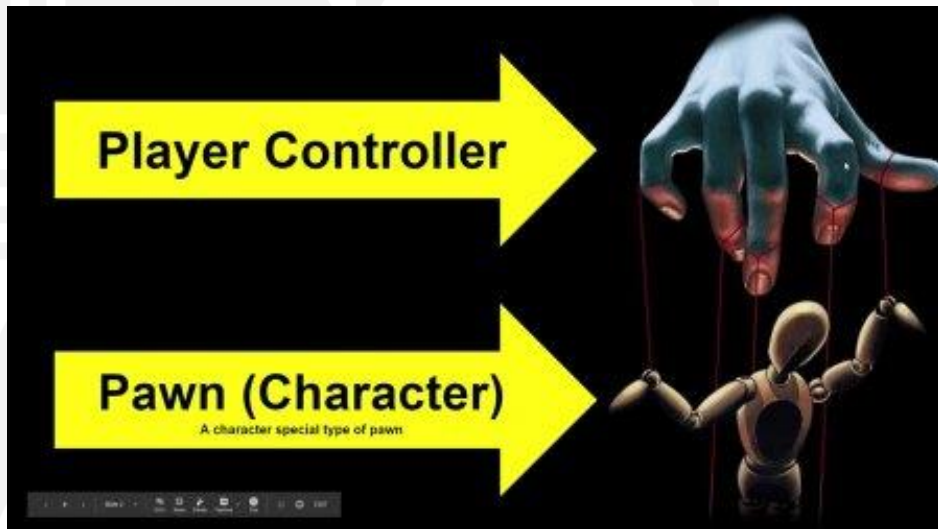


Pawn



# Los pawns necesitan un Player Controller/AI Controller para interactuar.

- Es la interfaz entre un pawn y un jugador humano. Nos permite recibir los inputs del teclado/control para que llegue a los pawns y puedan reaccionar.
- AI Controller no lee inputs pues no será controlado por un humano.



Player

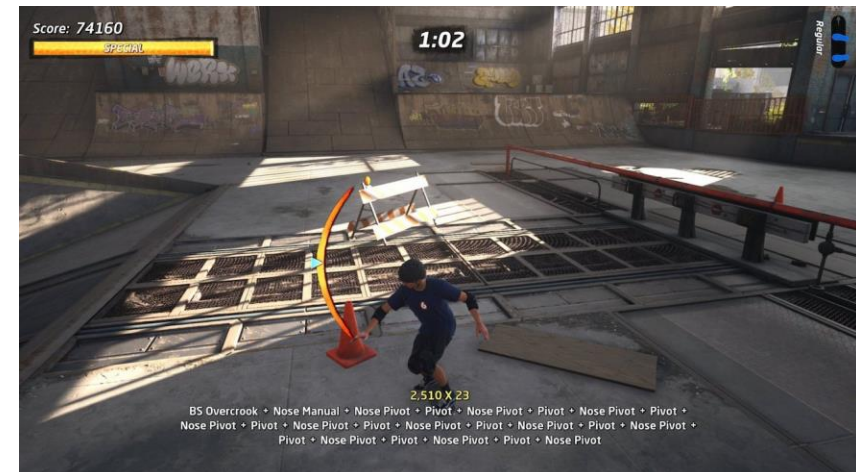


AI



# Repasando Inputs

- Recordando: Los Inputs es la forma en que traducimos las entradas del jugador (con un teclado o control) y las convertimos en acciones para nuestros actores.



## Hagamos un Character Blueprint con el señor maniquí!



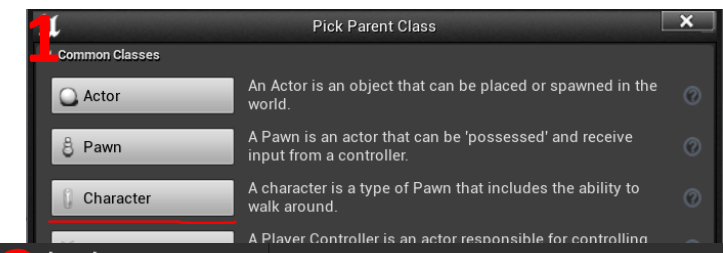
Moveremos nuestro personaje con Inputs: Correr y Saltar.



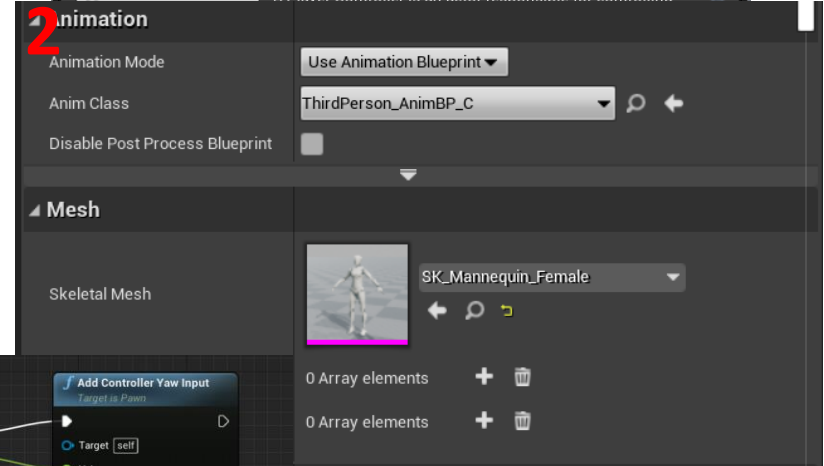
Controlaremos la cámara y el personaje se orientará según el movimiento de la perspectiva.



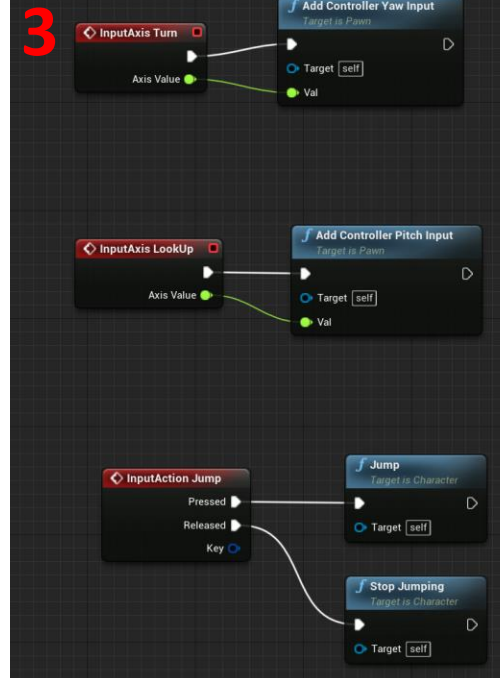
## Paso 1: Crear el Character Blueprint



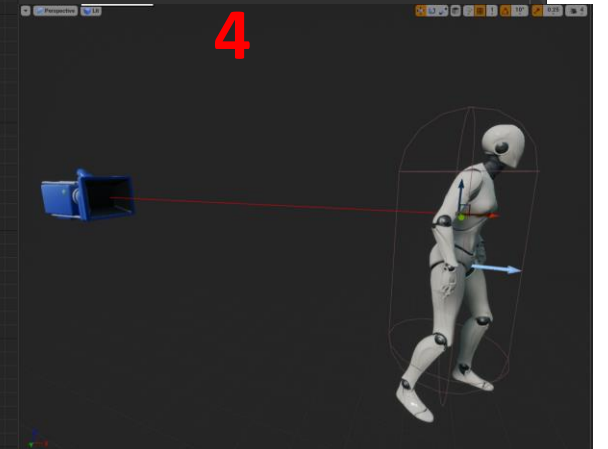
## Paso 2: Agregar el Skeletal Mesh a nuestro Character



## Paso 3: Agregar la lógica de input a nuestro blueprint



## Paso 4: Agregar el SpringArm con su cámara y ajustar el movimiento de nuestro Character





# Conclusiones

- **Aprendimos varios conceptos básicos (GameMode, PlayerController, Character, Pawn, HUD, etc).**
- **Vimos como configurar el GameMode.**
- **Aprendimos la diferencia entre Pawn y Character y cómo darle inputs a nuestro personaje y que se necesita un Player Controller para obedecernos.**
- **También vimos que se necesita una cámara en nuestro Character para visualizar correctamente el juego.**



# Tarea de la semana

- Crea 2 "GameMode" que invoquen diferentes "pawns" o "characters"



# Gracias

Correos:

- David García

[dgarcia@amberstudio.com](mailto:dgarcia@amberstudio.com)

- Arturo Escamilla

[arturo.garcia@amberstudio.com](mailto:arturo.garcia@amberstudio.com)



Arden Asociación Civil



arden\_asociacion



Classroom - código de clase plow4fn



Discord -

<https://discord.gg/K6PHCgz8Mb>

**AMERIKE**

INSTITUTO UNIVERSITARIO

AMBER



ARDEN



Agencia para el Desarrollo  
de Industrias Creativas y  
Digitales de Jalisco



CIUDAD  
CREATIVA  
DIGITAL  
GUADALAJARA

