VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Práctica 2 Unreal: “The Escapist”

Animación Digital

Grado en Ingeniería Informática

Luis Blázquez Miñambres, 70910465Q

Contenido

1. Introducción

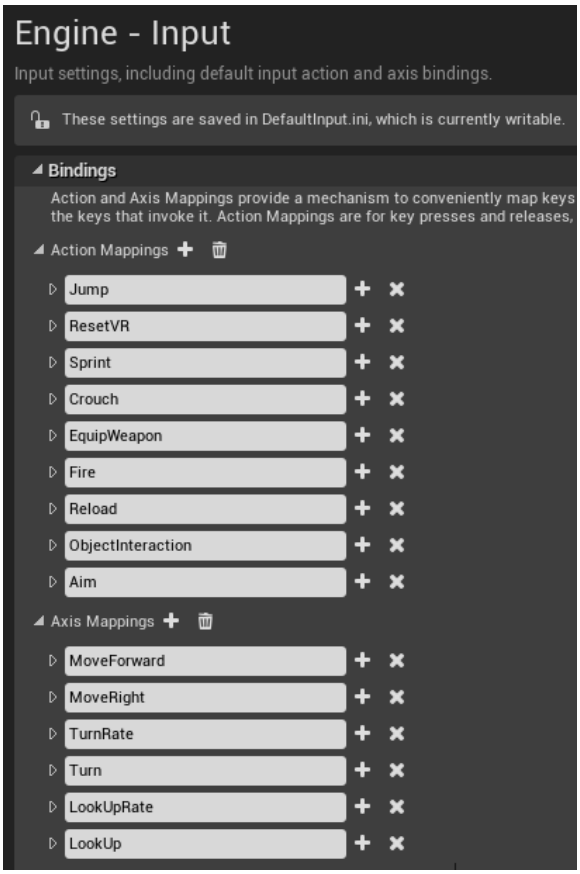
El juego que he realizado es un *shooter* en tercera persona en la que el jugador principal debe encontrar una manera de escapar de una prisión combatiendo a varios enemigos por el camino con un arma de fuego.

El aspecto de la interfaz principal quedaría de la siguiente manera, donde la cámara que sigue al personaje está un poco movida a la derecha de este debido al sistema de disparos que explicaremos más adelante.

Imagen 1.1: Interfaz principal del juego

Podemos observar otros detalles, que ya explicaré más adelante, como la cruz del centro cuando el arma esta equipada o la cantidad de munición que nos queda y tenemos en cada instante, así como la barra de salud del personaje.

Otro detalle como introducción al resto de apartados es indicar las entradas de teclado y ratón que he añadido al juego para simplificar la tarea a la hora de crear los blueprints con eventos de teclado y ratón.



2. Manual de Juego

Aunque los controles del juego se pueden observar en el menú del juego, aquí los dejaré reflejados. En concreto:

Disparar arma: Botón Izquierdo del ratón

Apuntar con arma: Botón derecho del ratón

Mover al personaje: Teclas W, A, S, D

Interactuar (Pulsar un botón, recoger objeto, …): Tecla E

Equipar o desequipar el arma: Tecla F

Saltar: Barra espacidora

Recargar munición: Tecla R

Sprint: Tecla Mayus Izquierda

3. Desarrollo del Juego

A continuación, explicaré brevemente el desarrollo del juego desde las fases más primerizas hasta llegar al juego final.

a. Importación de personaje y movimiento

Tanto para la importación del aspecto del personaje principal y de los enemigos, así como sus respectivas animaciones, las importé desde la página web "*Mixamo"* ya que intenté importarlas desde otros portales, pero fue imposible debido a la capacidad de almacenamiento que suponían.

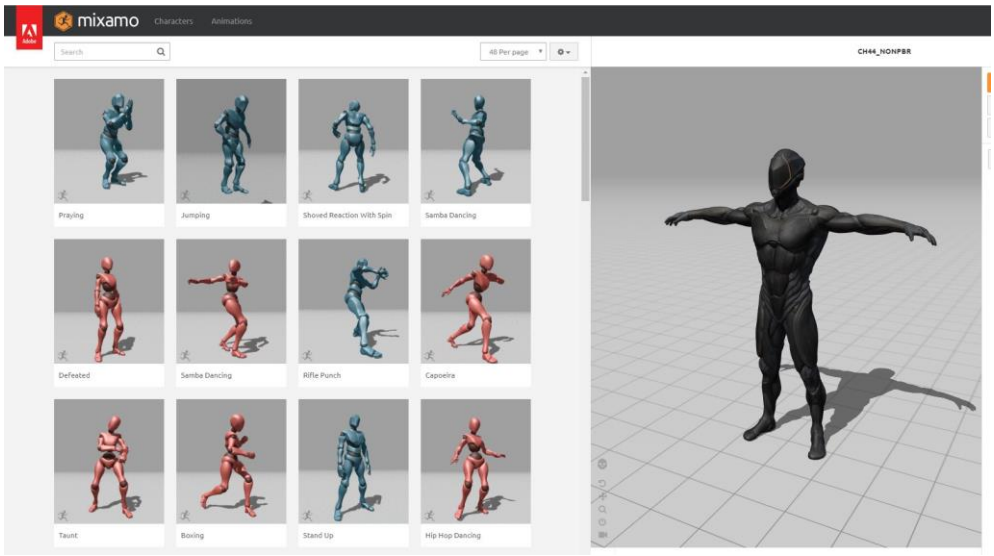


Imagen 3.a.1: Aspecto de la página web Mixamo

En primer lugar, cree un proyecto vacío con los ajustes por defecto para un juego en tercera persona que trae, entre otras cosas, un personaje y un escenario por defecto como el que se puede ver en la imagen *3.a.2*, dejando los blueprints que venían por defecto con respecto a las animaciones (ajuste de la velocidad, del salto, etc). A partir de aquí cogí al personaje del escenario y le cambié el *Skeletal Mesh* al de mi personaje que había importado.

Imagen 3.a.2: Personaje principal con skeletal mesh del personaje de Mixamo

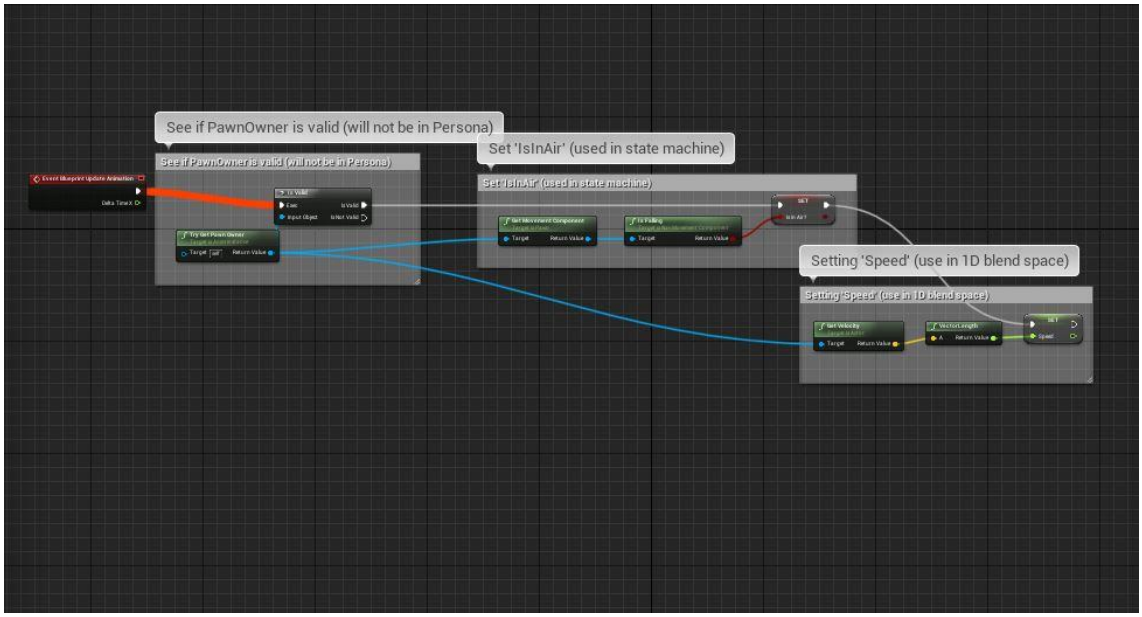


Imagen 3.a.3: Blueprints del AnimGraph del personaje por defecto en tercena persona

A continuación, ajuste una de las propiedades del personaje, el movimiento, a través de una variable en el *EventGraph* de mi personaje. Además de ajustar la acción de Sprint (mientras se mantenga la tecla Mayus Izquierda), aumentando el valor de la propiedad ya intrínseca en el personaje (*Max Walk Speed)* del *ThirdPersonCharacter* consiguiendo aumentar la velocidad del personaje e ir cambiando entre las acciones de caminar y correr.

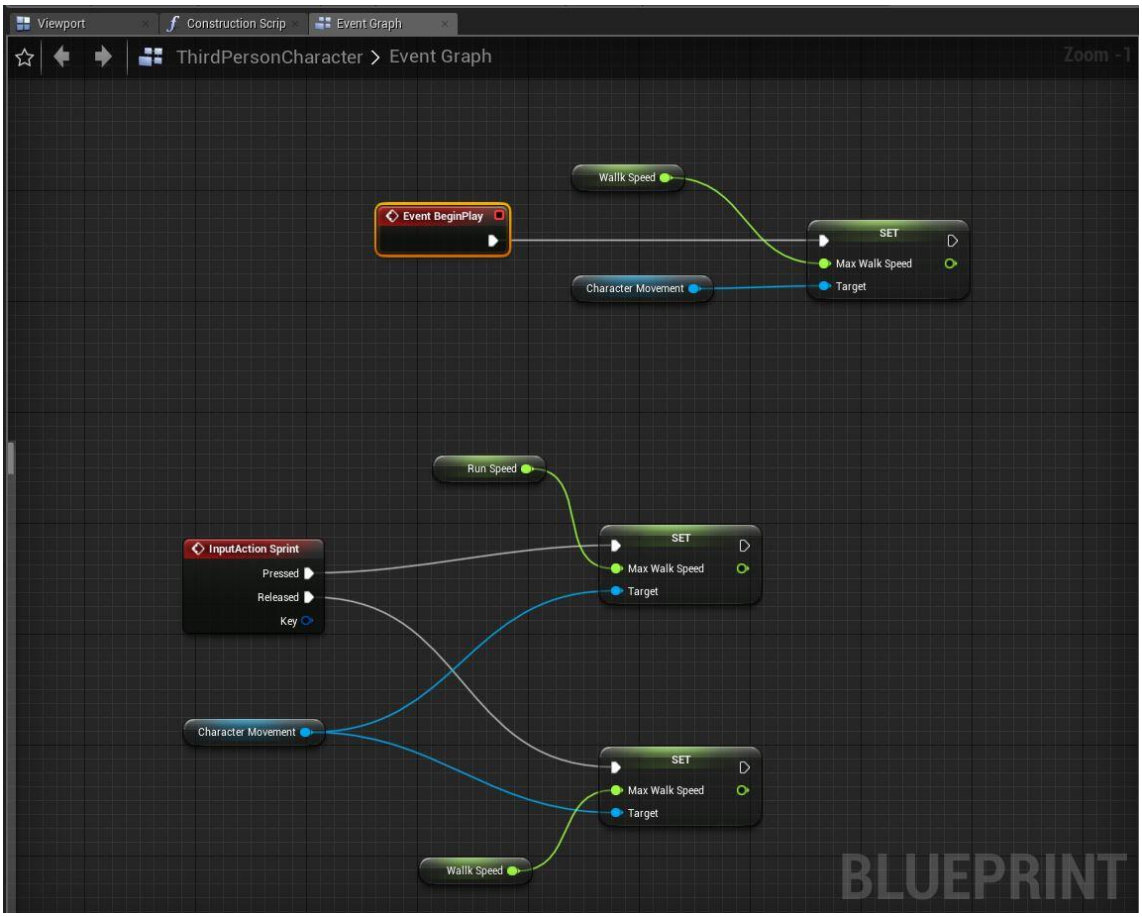


Imagen 3.a.4: Blueprints para la acción de movimiento – correr y andar

b. Grafo de animaciones

Posteriormente, añadí el conjunto de animaciones (que se pueden encontrar en *ThirdPersonBP > MyCharacter > Animations*) principales para caminar, correr y saltar del personaje en el apartado *MyCharacter_Animation > AnimGraph > Locomotion* del personaje.

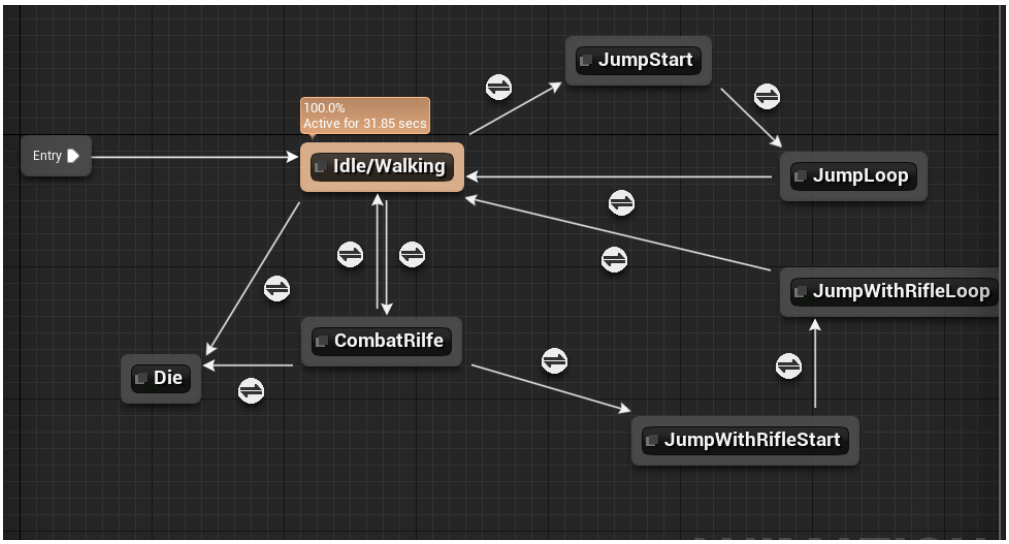


Imagen 3.b.1: AnimGraph del personaje

Principalmente hay que decir que tendremos tres estados principalmente:

- *Idle/Walking:* este estado tendrá las animaciónes de caminar y correr, agrupadas en un Blend Tree, con un solo eje (X) en función de la velocidad. A continuación, explicaré el contenido del Blend Space de movimiento donde se puede ver el único eje X que he mencionado y tres puntos que indican las tres animaciones que se ejecutarán cuando la velocidad del personaje sea 0 (animación *Idle o parado)*, cuando sea 250 (*caminar*) y cuando sea máxima 500 o más (*correr*).

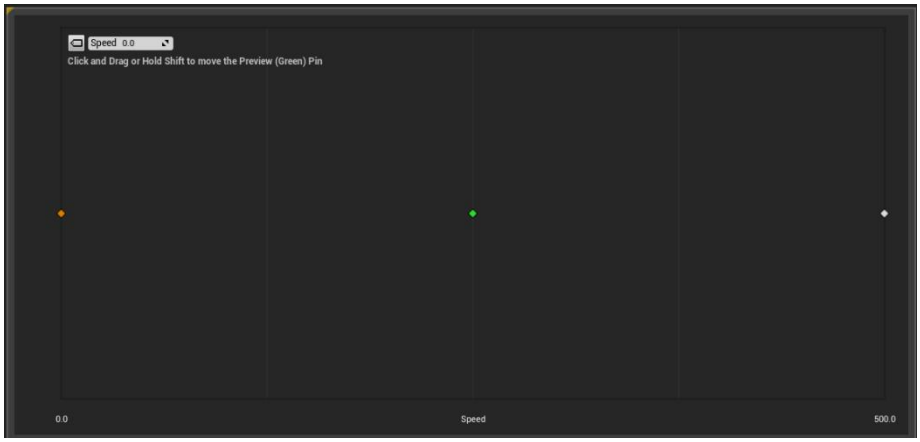


Imagen 3.b.2: AnimGraph del personaje

- *JumpStart y JumpLoop:* la animación de JumpStart y JumpLoop son la misma con la excepción de que se pasa del estado de caminar al primer estado de saltar mencionado, JumpStart, el jugador pulsa la tecla asociada a la acción *Jump.* Y pasa de este estado al segundo estado, JumpLoop, mientras esté en el aire. Esto último se puede detectar a través de una función en el Blueprint del personaje llamado *IsInAir*, intrínseca en el personaje por defecto (como se puede comprobar en la imágen *3.a.3*). Volviendo al estado de caminar cuando el valor de la función anteriormente mencionada es falso.

- *CombatRifle:* para el movimiento con el arma también utilicé un BlendSpace que dependía de la dirección a la que se movía el personaje y su velocidad, al igual que el estado de movimiento normal.

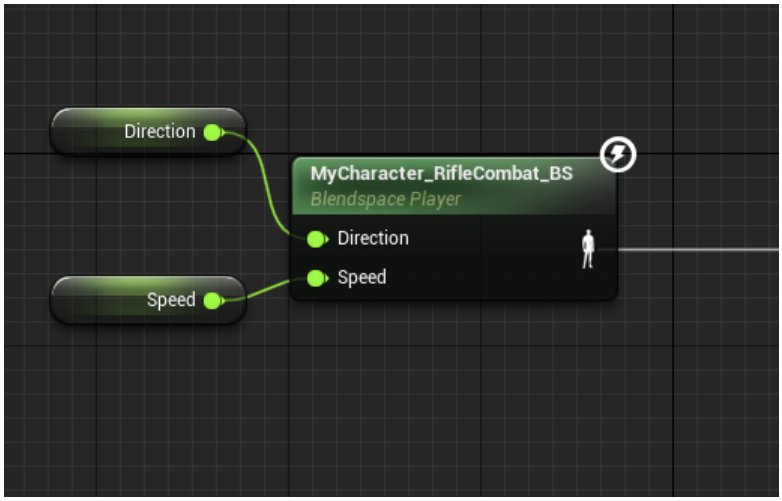


Imagen 3.b.3: BlendSpace dentro del estado CombatRifle en el AnimGraph

Por otro lado, el BlendSpace creado para el movimiento del personaje del mapa dependerá de dos ejes, X e Y, indicando la velocidad de movimiento y la dirección a la que apunta el personaje con el arma, respectivamente. Pudiendo caminar hacia atrás con el arma puesta sin que la cámara gire cuando la dirección del personaje es menor o igual a 180.

Imagen 3.c.3: BlendSpace de movimiento con el arma equipada

El efecto para conseguir que, equipada el arma, la cámara se mantenga en la misma dirección permitiendo que el personaje camine hacia atrás sin rotar, se consigue en el *EventGraph del* ThirdPersonCharacter con un macro llamada *Orient/Yaw* en la que se determina si el personaje girará sobre si mismo sin el movimiento de cámara (*Use Controller Rotation Yaw)* o si , teniendo el arma equipada, no girará al personaje sino que la orientación se mantendrá fija, permitiendo caminar hacia atrás o hacia los lados apuntando siempre hacia el frente con el personaje. Se puede ver en la función *Pitch/Yaw* en el EventGraph del personaje en *MyCharacter_AnimationBluerint*.

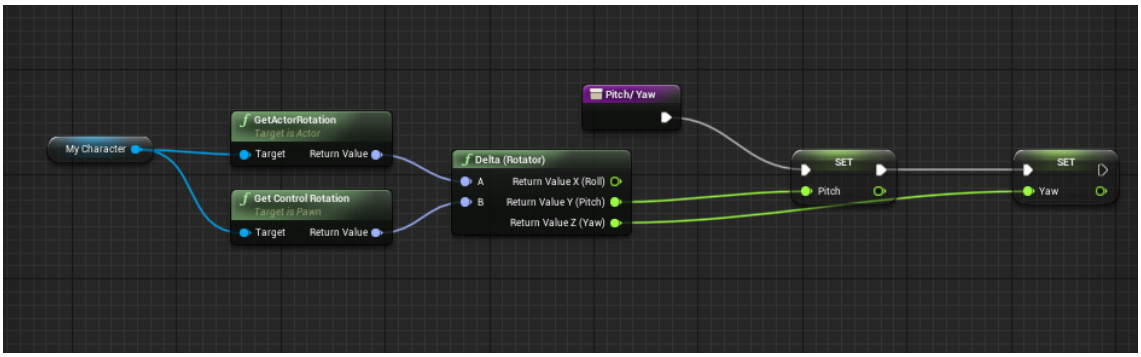


Imagen 3.c.4: Función para asignar la orientación y rotación del personaje en cada momento para las animaciones

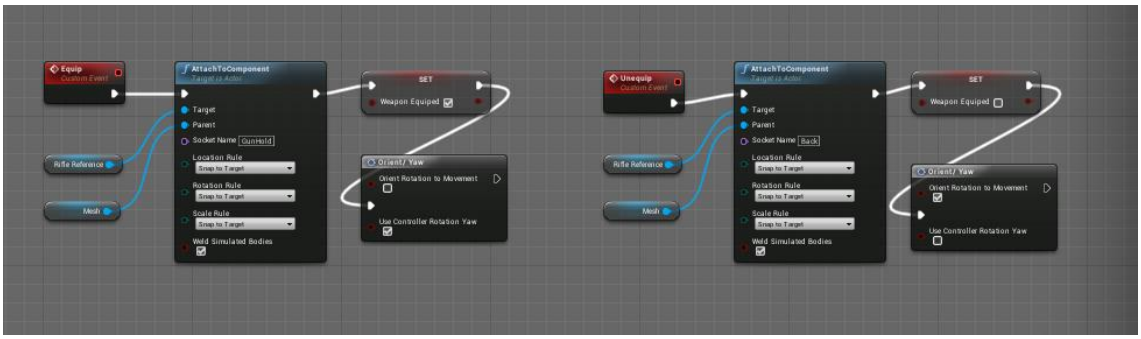


Imagen 3.c.5: Evento de equipar y desequipar el arma en EventGraph del ThirdPersonCharacter (ThirdPersonBP > Blueprints)

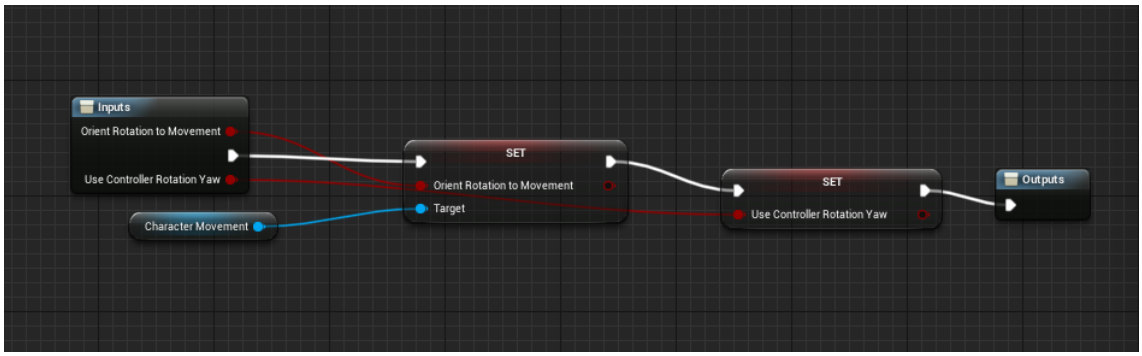


Imagen 3.c.6: Macro para evitar que rote la cámara con el personaje cuando el arma está equipada y siempre mire al frente cambiando la orientación del personaje o bloquearla en movimiento (rectángulo en gris de la imagen 3.c.5)

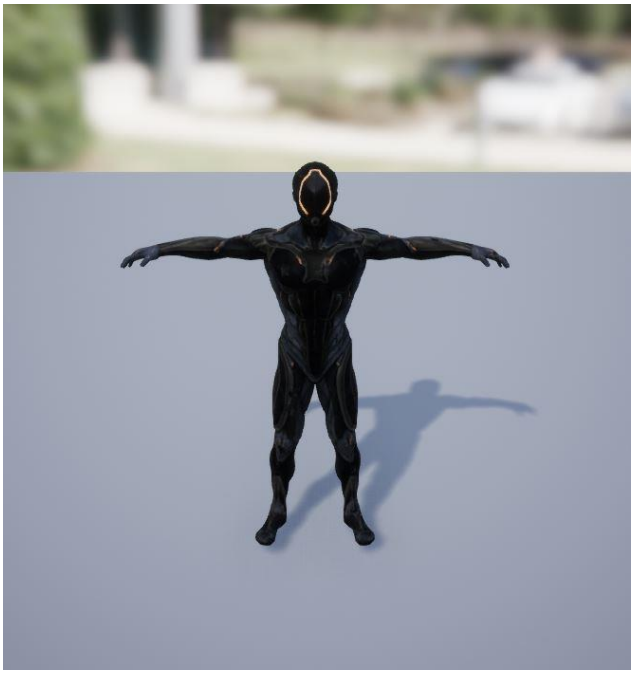
- *JumpWithRifleStart y JumpWithRifleLoop:* de manera análoga a la estructura de las animaciones JumpStart y JumpLoop , con la excepción de que estas animaciones parten del estado CombatRifle (cuando el arma está equipada).
- *Die:* activa la animación de muerte del personaje cuando la variable *Die* del Blueprint del personaje consigue un valor verdadero, que será cuando su vida sea menor o igual a 0.

c. HUD

4. Características

A continuación, se detallarán las características de los objetos utilizados en el juego, ya estén relacionados o no con el jugador, como los enemigos, objetos coleccionables, o los mapas creados.

a. Jugador principal



b. Enemigos

c. Mapa

Para el mapa del juego, he utilizado un mapa predefinido de uno de los paquetes del Bazar de la tienda de *Epic Games: Module Sci Fi Season 1 Starter Bundle* con el que pretendía crear el ambiente y temática de ciencia ficción para el desarrollo del juego.

Cabe destacar que el diseño del mapa es uno que venía por defecto en el paquete al que luego yo, posteriormente, he ido añadiendo y quitando elementos, como habitaciones, iluminación, posiciones de los enemigos, paneles interactuables, … que no venían incluidos como parte del mapa.

Imagen 4.d.1: Vista preliminar del escenario

Imagen 4.d.2: Sala de control para reestablecer la energía

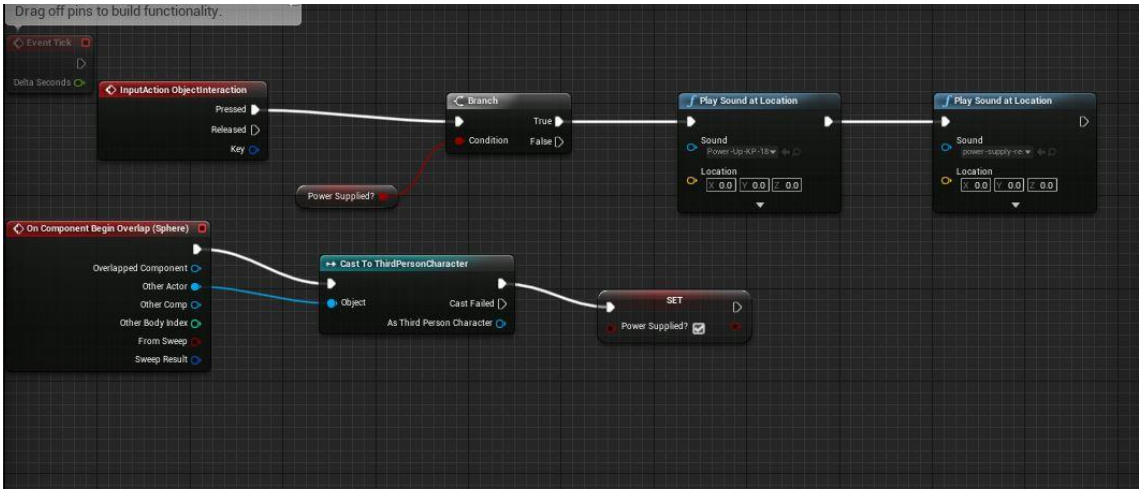


Imagen 4.d.4: Blueprint Actor del objeto de la terminal de la sala de control

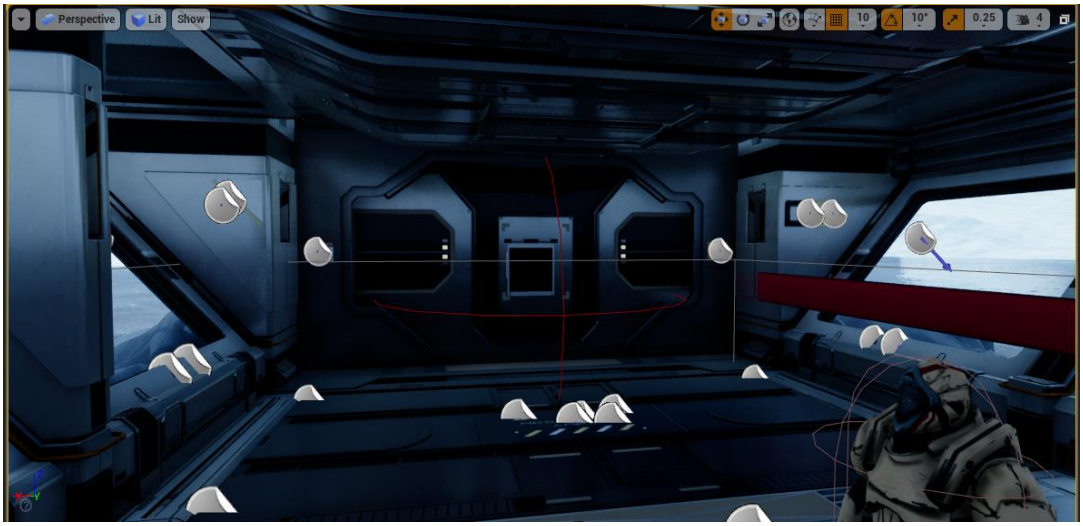


Imagen 4.d.4: Salida del fin de juego

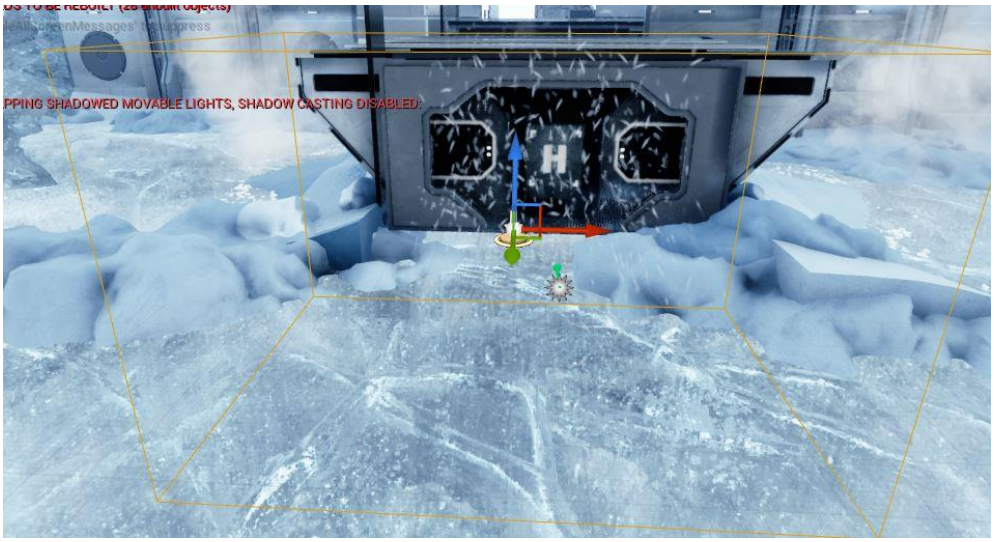


Imagen 4.d.5: Trigger Box (señala el fin del juego)

d. Arma

i. Equipar/Desequipar

ii. AIM Offset

e. Objetos recogibles

i. Vida

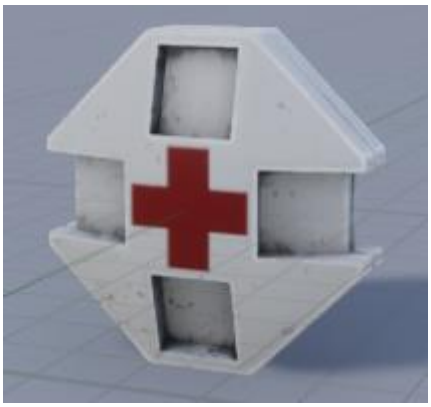


Imagen 4.e.i.1: Static Mesh del objeto de vida

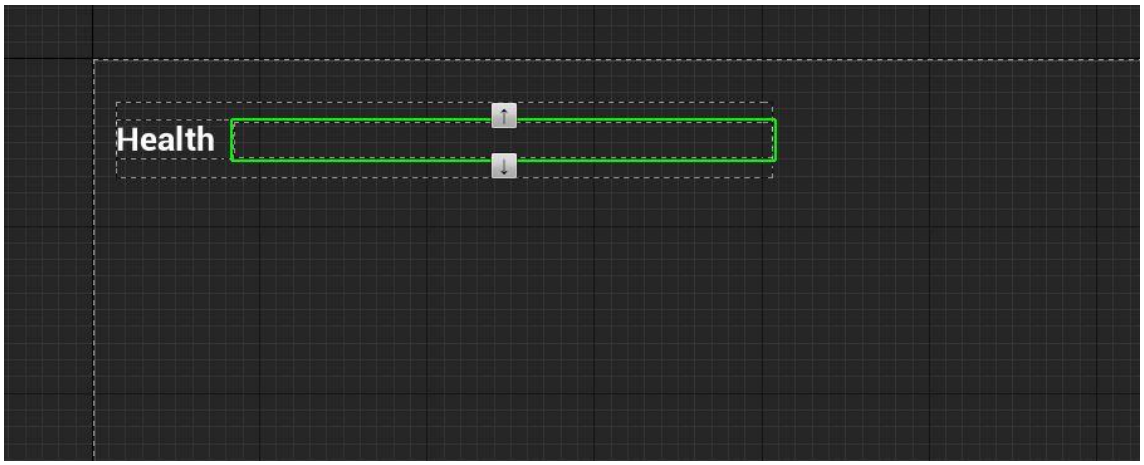


Imagen 4.e.i.2: Barra de vida en el HUD principal (widget)

Imagen 4.e.i.3: Blueprint del binding para controlar la barra de la vida

ii. Munición

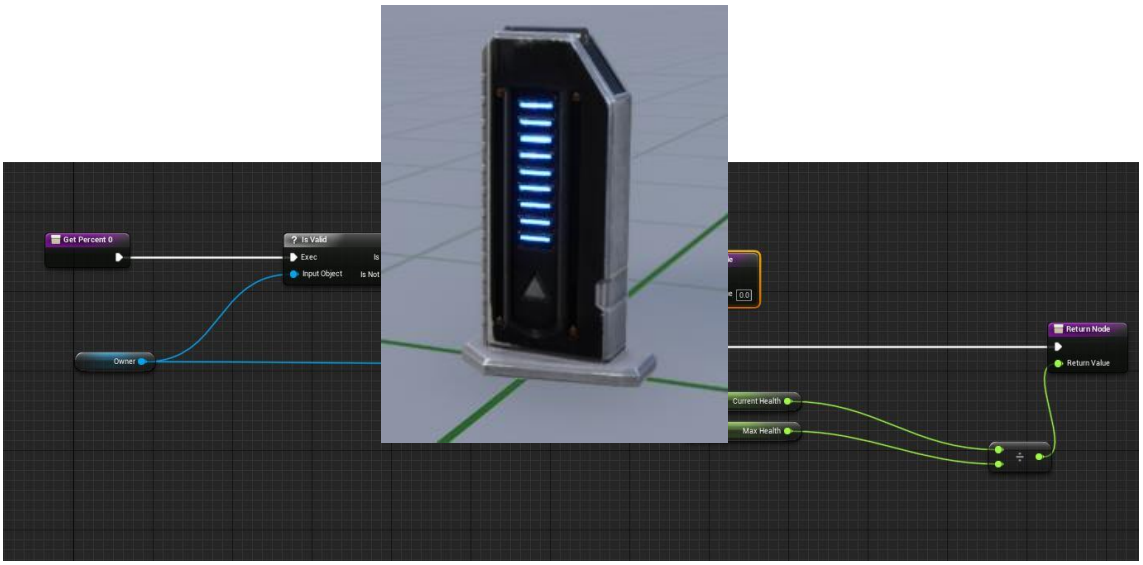


Imagen 4.e.ii.1: Static Mesh del objeto de munición

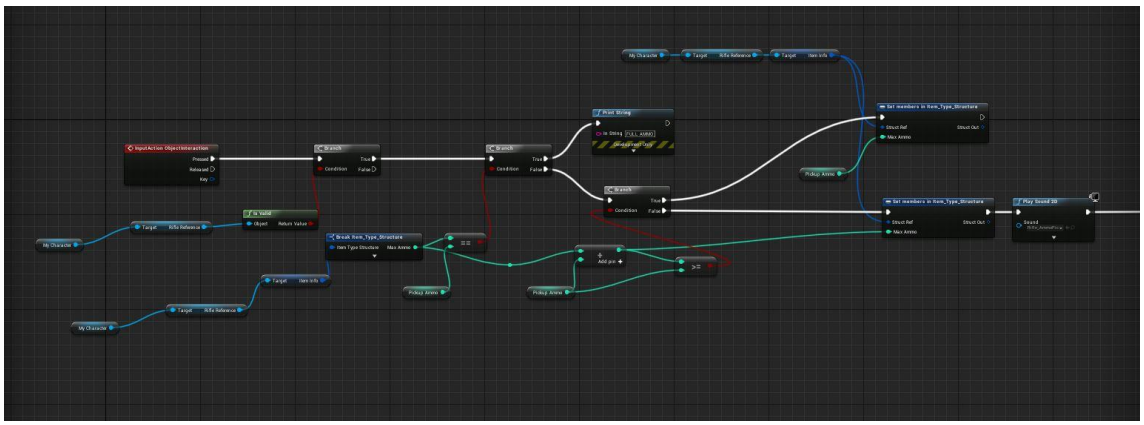


Imagen 4.e.ii.2: Implementación del elemento Overlap con la acción que se realizará cuando el jugador choque con el objeto

5. Bibliografía

- Guia Unreal Third Person Shooter. Diseño Shooter, *Youtube*: https://www.youtube.com/watch?v=QrqKTN-ORzI&list=PLM6ZWbxOgIqsCb3dUJRdYoUbyi_zFCOjs
- Música de Fondo Juego, Tron Legacy BSO: https://www.youtube.com/watch?v=qRUfcp8Ba6g&list=WL&index=335&t=12s
- Animaciones de fondo en Widgets: https://www.youtube.com/watch?v=1ZrQppQGIok&list=WL&index=333&t=0s
- Textos de Inicio: https://www.youtube.com/watch?v=uxbcVucO59Q&list=WL&index=332&t=0s
- Conseguir efecto de fogonazo en el arma: https://www.youtube.com/watch?v=7OeyICXSdVE&list=WL&index=287&t=0s
- Método de recarga de armas automática y avanzada: https://www.youtube.com/watch?v=Pw8FG8-_1gA&list=WL&index=283&t=1260s
- Animaciones con armas: https://www.youtube.com/watch?v=S97c34UkLBI&list=WL&index=282&t=311s
- Como crear enemigos IA inteligentes: https://www.youtube.com/watch?v=fUtqgy511Bw
- Crear Menú principal: https://youtu.be/kM_dmNtXj4o