

FPS multijugador



**UNREAL
ENGINE**

Esperanza Barcia Ortiz

Contenido

Introducción, objetivos y motivación del proyecto	3
Descripción técnica	4
First person character	4
Movimiento	4
Disparo	4
Daños.....	5
Sanación	6
Efectos visuales	7
First Person Gamemode.....	7
Player widget.....	7
Destructible object.....	8
Projectile	9

Introducción, objetivos y motivación del proyecto

En este proyecto se nos pedía crear un juego de disparos utilizando el sistema multijugador de Unreal, el cual debía tener los siguientes requisitos:

- Los jugadores tendrán un sistema de recursos: energía, balas, vida, cargadores.
- Los jugadores harán daño de al menos dos tipos de escuelas diferentes (p. ej. Físico, frío, fuego...)
- Los jugadores van desarrollando inmunidad al daño de una escuela a medida que lo reciben (rendimiento decreciente) La inmunidad es total después de recibir daño tres veces consecutivas del mismo tipo.
- Los jugadores pueden disparar haciendo clic izdo. (un tipo de daño puntual que genera 10 puntos energía) y clic derecho (tipo de en área que consume 70 puntos de energía y destruye las coberturas).
- Los jugadores poseen balas limitadas para el disparo de clic izdo. Si se gastan las balas del cargador activo, el jugador será informado y deberá recargar pulsando una tecla, siempre que tenga suficientes cargadores de repuesto.
- Si el jugador acierta dos disparos consecutivos contra el oponente, gana un cargador. Se pueden tener un máximo de cinco cargadores (1 activo + 4 reserva)
- Los jugadores deben poseer un sistema de salud. Si se pierde toda, el jugador muere. Durante la muerte, el jugador será informado (Se muestra el mensaje Has muerto, y se tinte la imagen de rojo y aparece una cuenta atrás de 10 segundos) durante los cuales, el jugador no podrá moverse durante hasta que no llegue a cero.
- Los jugadores tendrán una interfaz de usuario gráfica donde se muestre su salud, su energía, número de balas, número de cargadores, y tiempo de reutilización.

Opcional

- Los jugadores cuya salud cae por debajo del 30% pueden usar el botón central del ratón para usar una poción que sana 70 puntos de salud, pero posee un tiempo de reutilización de 60 segs).

El objetivo principal de la práctica era tomar contacto con el sistema multijugador de Unreal y hacer frente a las complicaciones que lleva trabajar con un servidor.

Me ha parecido muy interesante para aprender cómo funciona un juego multijugador, tanto lo que se muestra a los jugadores como el funcionamiento a la hora de desarrollarlo.

Descripción técnica

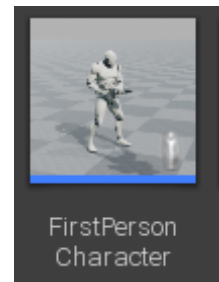
Los elementos principales del proyecto son:

- First Person Character
- First Person GameMode
- Player Widget
- Destructible Object
- Projectile

First person character

Este blueprint contiene todo el funcionamiento del personaje, se puede dividir en:

- Movimiento.
- Disparo.
- Daños.
- Sanación
- Efectos visuales.



Todas las variables tienen activada la función Replicated, ya que todos los jugadores van a tener las suyas propias.



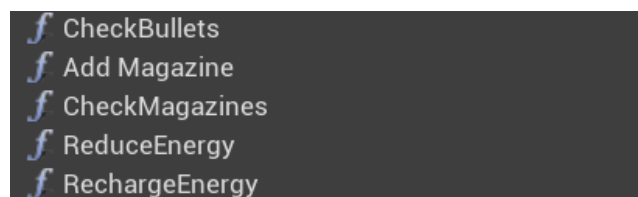
Movimiento

El movimiento utilizado es el que viene por defecto en el proyecto fps de Unreal.

Disparo

El jugador puede disparar dos tipos de proyectil dependiendo de la tecla del ratón pulsada. Es necesario tener un mínimo de energía para cada uno.

Las funciones relacionadas con el disparo son las siguientes:



El jugador debe cumplir una serie de requisitos para poder disparar, los cuales se comprueban en el blueprint:

1. Estar vivo.
2. No estar congelado.
3. Tener balas.
4. Tener energía suficiente dependiendo del disparo.

Tras disparar se consumirá o se añadirá energía.

Como medida preventiva, si el jugador se queda completamente sin energía, si esto pasa se recargará un mínimo automáticamente tras un tiempo ya que ante el usuario puede resultar extraño poder disparar sin disponer de energía.

El jugador podrá recargar siempre que tenga cargadores disponibles. Los cargadores se obtienen de la siguiente forma.

1. El proyectil generado detecta con quien colisiona sabiendo quién le ha disparado.
2. Si el objeto colisionado es el enemigo suma un contador.
3. Al llegar ese contador a 2 se añade un cargador al instigador.

Daños

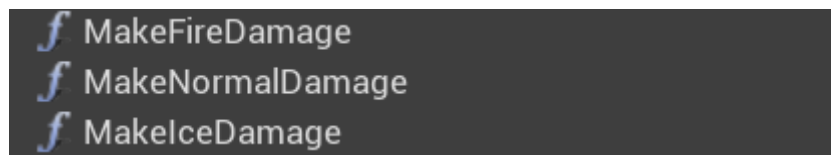
El jugador puede infligir daño de 3 tipos:

1. Daño normal: quita 30 puntos al enemigo.
2. Daño congelador: congela por unos segundos al enemigo
3. Daño por fuego: quita daño aleatorio durante un tiempo.

Por defecto, el jugador tiene seleccionado el normal, puede cambiarlo pulsando los números del teclado.

Cabe recalcar que al cambiar la selección se resetea la inmunidad que el otro personaje pueda tener a cierto tipo de daño.

Las funciones relacionadas son:



```
f MakeFireDamage  
f MakeNormalDamage  
f MakeIceDamage
```

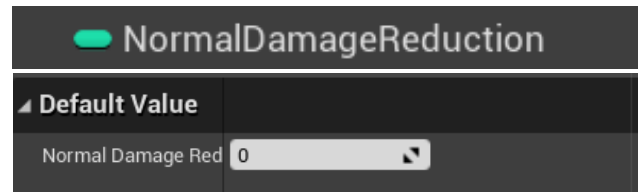
Funcionan de formas diferentes ya que he trabajado experimentando un poco nuevas técnicas y opciones en blueprints.

Estos daños son decrementales, es decir, harán menos efecto según se disparen consecutivamente. Al ser disparado un jugador 3 veces consecutivas será inmune.

Daño normal

Este tipo de daño quita 30 puntos de vida al jugador.

Para disminuir su efecto he creado una variable que comienza en 0 e irá aumentándose con cada disparo recibido. Una vez supere el daño que se realiza al jugador, deja de tener efecto.



Daño congelador

En este caso se impide el movimiento al jugador durante un tiempo, simulando estar congelado.

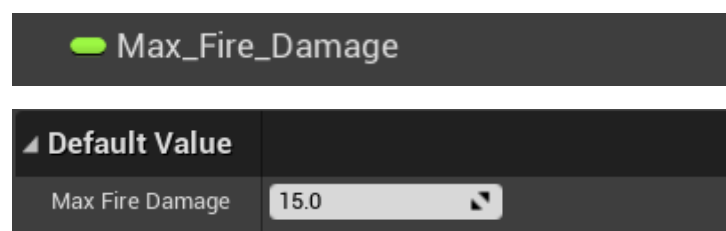


El método utilizado es reducir el tiempo de congelación con cada disparo recibido. Al reducirse hasta 1 deja de hacer efecto.

La cuenta atrás se muestra con un timer.

Daño por fuego

Al dañar por fuego se llama mediante un timer (permite llamar cada cierto tiempo a una función mientras esté activo) a la función que resta vida. Al ser un rango aleatorio, lo que se disminuye es el daño máximo. Al reducirse hasta 1 deja de hacer efecto.



Sanación

En el caso de tener el jugador menos del 30 por ciento de su salud se presenta la opción de sanación. Tras usarla se espera un delay de 60 segundos para poder reutilizarla.

La cuenta atrás se muestra mediante un timer.

Efectos visuales

Al comenzar se guarda en una variable un material instance del material del personaje para poder variar su apariencia.

Los eventos Change color y Reset Color permiten esto.

Al ser dañado con fuego o congelado su apariencia cambiará.



First Person Gamemode

El gamemode de nuestro proyecto.

Es importante cambiar el parámetro para que se spawnee a nuestro personaje.



Player widget

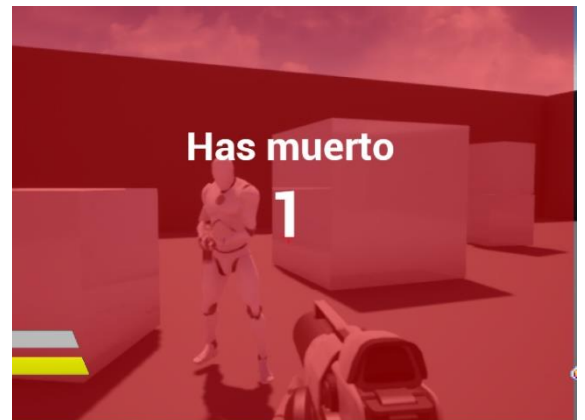
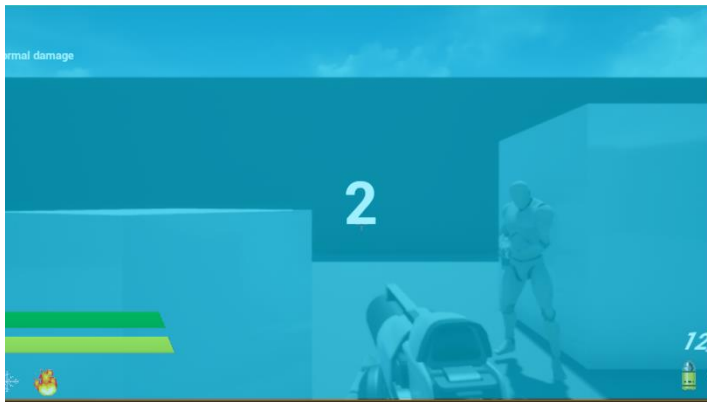
La interfaz que verá el jugador.

Para evitar problemas, al inicio del juego se guarda en una variable que estará replicada de forma que ambos jugadores tengan su interfaz propia.



Los valores de cada elemento de la interfaz están enlazados con las variables del first person character directamente.

Posee distintas pantallas y textos que se muestran solo cuando sucede algo concreto.

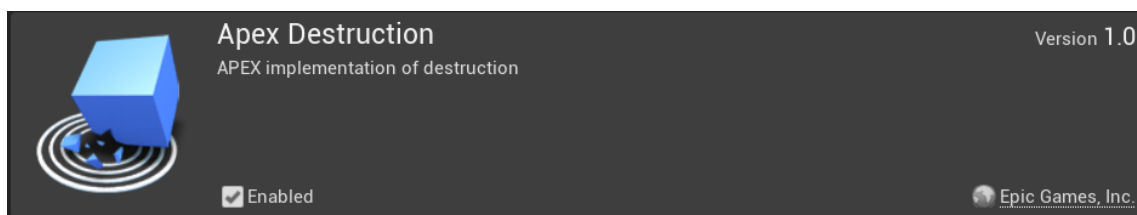


Además muestra imágenes negras o no dependiendo de si el enemigo es inmune a un daño concreto. Esto se ha enlazado con las variables correspondientes a la inmunidad, cuando ciertas cosas son true, se muestran.

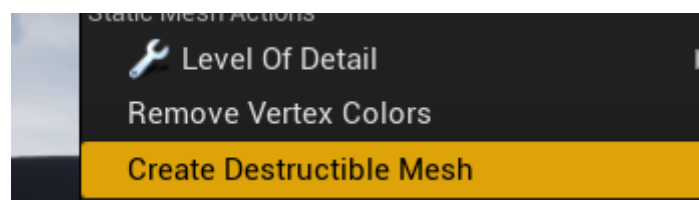


Destructible object

Las coberturas destructibles están realizadas gracias a un plugin



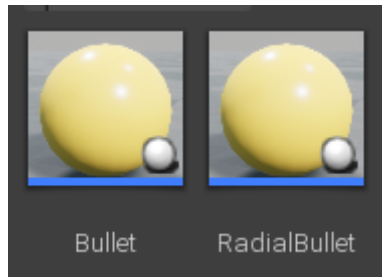
Para crear una cobertura destructible elegimos una malla. Crearemos una malla destructible.



En el blueprint de este objeto crearemos un evento para que al ser golpeado se destruya la malla en trozos.

Projectile

Ya que existen dos tipos de proyectil he creado dos objetos diferentes. Aquel que hace daño puntual y el que hace daño radial además de romper las coberturas. De esta forma se spawneará uno u otro.



Dentro de estos blueprint haremos una serie de comprobaciones para evitar el daño por rebote y haremos uso de los siguientes nodos que resultan muy útiles.

