

Explosión de bala en Multicast

Introducción

En esta actividad expondremos cómo, partiendo del proyecto *MultiplayerTutorial* trabajado en clase, se ha generado un efecto de explosión de bala en cada cliente cada vez que una bala impacta con alguna superficie.

Multicast

Los métodos Multicast son una forma para pedir, desde el servidor a todos los clientes, que estos ejecuten alguna función. Estos métodos solo pueden ser llamados desde el servidor, ya que es el único que conoce a todos los clientes y tiene comunicación directa con ellos. Si son llamados desde un cliente, serán ejecutados tan solo por ese cliente.

En el proyecto trabajado en clase, dentro de la bala tenemos un override del `OnComponentHit` de la esfera de colisión de la bala. Este método es ejecutado por todo el mundo, tanto servidor como clientes. Por lo tanto, lo único que deberíamos hacer para generar una explosión desde todos los clientes es, dentro del `OnComponentHit`, justo antes de destruir la bala, comprobar que estemos en servidor y, en caso afirmativo, llamar al método Multicast que generará un efecto Niagara de explosión. Entonces, el servidor lo que hará será pedir a cada cliente que ejecute ese método. Como podemos ver en la siguiente tabla, si llamásemos ese mismo método desde un cliente, este solo sería ejecutado por ese cliente, ya que el servidor es el único que tiene comunicación directa con todos los clientes.

RPC invoked from the server

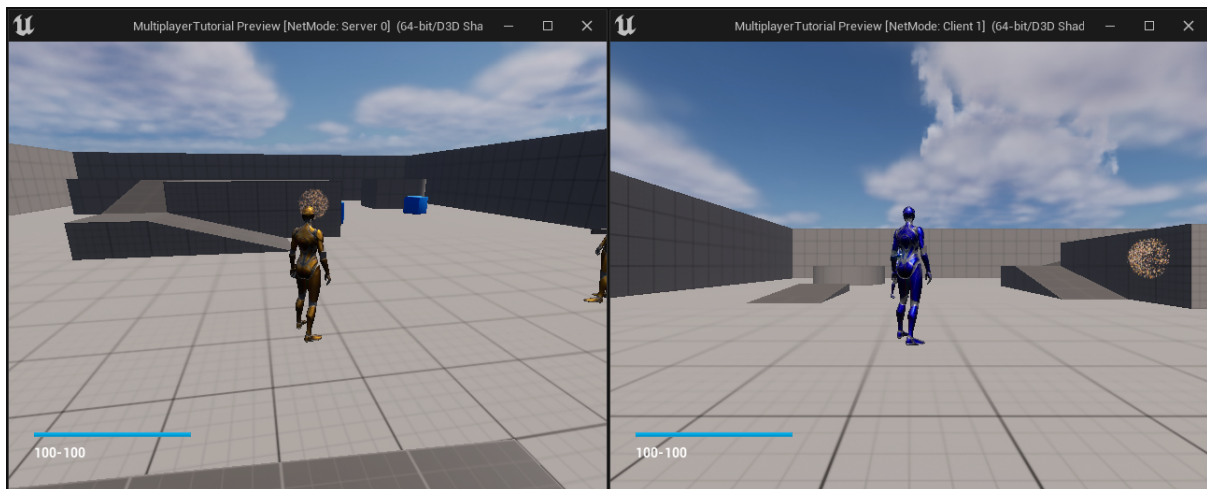
Actor ownership	Not replicated	NetMulticast	Server	Client
Client-owned actor	Runs on server	Runs on server and all clients	Runs on server	Runs on actor's owning client
Server-owned actor	Runs on server	Runs on server and all clients	Runs on server	Runs on server
Unowned actor	Runs on server	Runs on server and all clients	Runs on server	Runs on server

RPC invoked from a client

Actor ownership	Not replicated	NetMulticast	Server	Client
Owned by invoking client	Runs on invoking client	Runs on invoking client	Runs on server	Runs on invoking client
Owned by a different client	Runs on invoking client	Runs on invoking client	Dropped	Runs on invoking client
Server-owned actor	Runs on invoking client	Runs on invoking client	Dropped	Runs on invoking client
Unowned actor	Runs on invoking client	Runs on invoking client	Dropped	Runs on invoking client

Fuente: <https://docs.unrealengine.com/5.1/en-US/rpcs-in-unreal-engine/>

A continuación, podemos ver el resultado de la explosión de bala vista desde dos clientes en *Net Mode = Play as Listen Server*.



Finalmente, a continuación podemos ver el resultado de la explosión de bala vista desde dos clientes en *Net Mode = Play as Client*:

