

## Introduction sur le multijoueur sur UE4

*(NB : J'espère que ce tuto vous sera utile et qu'il sera compréhensible par tous. Si vous avez des questions n'hésitez pas à m'envoyer un message sur Facebook « Antoine Gargasson » ou par mail [antoine.gargasson@gmail.com](mailto:antoine.gargasson@gmail.com))*

*(NB2 : Ce tuto est réalisé sur la version 4.10.4 de UE4)*

Avant de commencer à entrer dans le vif du sujet, il est important de bien comprendre comment une structure serveur/clients fonctionne. Je vous propose dans ce tuto de nous pencher sur les bases du multijoueur sur UE4 en C++.

**Afin d'en savoir plus sur la structure du multijoueur et comment fonctionne la structure multijoueur, je vous demanderai de vous référer à la version Blueprint du tutoriel.**

Ce tuto ne traitera donc **que** de la partie pratique du multijoueur en C++ sur Unreal Engine 4.

## La pratique

Passons à la pratique ! Je vais vous montrer où se situe les changements de type et comment tout est représenté !

Commençons par les variables. Lorsque vous créez votre variable vous écrivez quelque chose comme suit dans le .h:

```
8  visibilité: //public, private, protected
9  type nom;
```

Vous nommez donc votre variable et changez son type à convenance.

```
3  private:
4  float vie;
5
```

Vous vous rendez ensuite dans la partie qui nous intéresse : Replication

```
3  private:
4  UPROPERTY(EditAnywhere, BlueprintReadWrite, Replicated, category = "Categorie")
5  float vie;
6
```

Vous avez le choix entre plusieurs types de réplication : Non répliquée, Répliquée (replicated) ou Répliquée avec notification.

N'oubliez pas de rajouter aussi cette fonction dans le .cpp sinon votre variable ne sera pas actualisée :

```
26  void ATutoPlayer::GetLifetimeReplicatedProps(TArray<FLifetimeProperty> &OutLifetimeProps) const
27  {
28      Super::GetLifetimeReplicatedProps(OutLifetimeProps);
29
30      DOREPLIFETIME(ATutoPlayer, Health);
31      DOREPLIFETIME(ATutoPlayer, Damage);
32  }
```

En ce qui concerne les événements. Ci-dessous vous pouvez voir comment un client envoie des données au serveur.

Il faut utiliser 4 fonctions différentes pour appeler une fonction sur le serveur. C'est un peu plus lourd qu'en Blueprint mais vous aurez plus de contrôle.

Dans le .h :

```
14     private:
15         // Handles the Fire on the client
16         void Fire();
17
18         // Handles the Fire on the server
19         UFUNCTION(Reliable, Server, WithValidation)
20         void ServerFire();
21         virtual void ServerFire_Implementation();
22         virtual bool ServerFire_Validate();
23
```

Dans le .cpp :

```
37     void ATutoPlayer::Fire()
38     {
39         if (Role < ROLE_Authority)
40         {
41             ServerFire();
42         }
43     }
44
45     bool ATutoPlayer::ServerFire_Validate()
46     {
47         return true;
48     }
49
50     void ATutoPlayer::ServerFire_Implementation()
51     {
52         //do stuff here
53     }
54
```

**Remarquez qu'on implémente pas la fonction "ServerFire", elle dépend de ServerFire\_Validate() et ServerFire\_Implementation().**

Vous avez le choix entre plusieurs types de répliquion : Non répliquée, Répliquée sur le serveur (Server), répliquée à tous (Multicast), répliquée sur le client(OwningClient).

Vous avez aussi dû remarquer le "Reliable". Il signifie que votre fonction doit à tout prix arriver au serveur. C'est très coûteux et peut ralentir votre jeu si vous en faites trop. Utilisez des "Unreliable" pour les fonctions moins importantes !

Pour la réplification des actors, une simple variable suffit à répliquer l'actor.

```
56  AReplicatedActor::AReplicatedActor(const class FPostConstructInitializeProperties& PCIP)  
57      : Super(PCIP)  
58      {  
59          bReplicates = true;  
60      }
```

## Pour tester vos jeux une fois packagé :

Créez deux raccourcis de votre .exe .

Nommez le premier "Server" (le nom importe peu). Cliquez droit dessus, propriété et localisez le champ "Cible" (ou target).

Rajoutez à la fin : " leNomDeVotreMap?listen -game"

Emplacement :           Haut\_DemiereBuild  
Cible :                    Build\Haut\_DemiereBuild\ProtoSI2016.exe map?listen -game

De Même pour le Client.

Rajoutez à la fin : " adresseIP -game"

Emplacement :           Haut\_DemiereBuild  
Cible :                    reBuild\Haut\_DemiereBuild\ProtoSI2016.exe 10.0.1.9 -game