

网络

Actor Replicates

启用同步复制，启用后Actor的创建和销毁自动同步，可以执行RPC等。

NetLoadOnClient:客户端加载地图时是否默认加载。如果该选项没有启用，若Replicates启用也可以在网络同步时被加载。

AlwaysRelevant:是否一直与其它客户端关联。如果没有启用，则根据下面选项

NetCullDistanceSquared确认关联距离

NetCullDistanceSquared:关联距离

IsLocallyControlled

用于判断Pawn是否是本地控制。

Actor Role

- | | |
|------------------------------|-------------|
| • None | - 不复制 |
| • SimulatedProxy | - 本地模拟 |
| • AutonomousProxy | - 本地 |
| • controller owner Authority | - 服务器原actor |

常用方法：

Has Authority(Switch Has Authority) - 判断当前Actor所在是客户端还是服务器

GetLocalRole - 判断本地Actor的Role：服务器-Authority，本地Simulate or Autonomous（如果开启Replication）

GetRemoteRole - 判断远端Actor的Role：客户端的远端是服务器Authority，服务器的远端是客户端：Simulate or Autonomous

Actor的两种复制方式

Variable Replication

1. Replicate
2. RepNotify

可以变量复制的应尽量使用变量复制，减少RPC调用，因为变量复制更高效。

RPC

1. None
2. Run On Server(Server)：客户端调用，在服务器执行
3. Multicast(NetMulticast)：服务器调用，广播到所有客户端；客户端调用，仅本地执行。
4. OwningClient(Client)：服务器调用，在所在客户端执行，服务器不执行

RPC调用时，应注意几个问题：1. 是否开启ActorReplicated；2. 本地是否OwningActor；3. 客户端不能Multicast.

Reliable：同步频率高的函数不建议使用Reliable，例如Tick等，因为同步频率高的函数实时性要求更高且高频率的Reliable容易使消息堆栈溢出

组件复制

Owner

Owner对象同步给获取到Owner的客户端连接，不是Owner的客户端连接因为客户端本地不存在其它客户端的Controller，因此其ActorOwner为null。

Relevancy按优先级由低到高

- 都为false，一定同步给Owner。如果A在范围，即时同步；后进入范围，则延迟同步。
- AlwaysRelevant（同步给所有人，无论Owner和范围）
- NetUseOwnerRelevancy（如果有Owner，只同步给Owner，即使A在范围内，也不同步；如果没有Owner，同步给A等位置相关等连接）
- OnlyRelevantToOwner（如果有Owner，只同步给Owner，即使A在范围内，也不同步；如果没有Owner，则不同步）

Gameplay的同步

GameInstance：引擎启动时创建，关闭时销毁。存在于各游戏实例，且不同步。

GameMode：仅存在于服务器

GameState：存在于服务器和客户端，服务器同步到客户端

PlayerController：服务器存有所有连接客户端的Controller，客户端本地仅有自己的Controller

PlayerState：存在于服务器和客户端，客户端上也拥有所有连接到服务器的PlayerState，生命周期和PlayerController类似，不会因Pawn而销毁

Pawn：存在于服务器和客户端，客户端上有所有连接到服务器的Pawn
