Tips: Open Level 这个参数，要填写完整路径，要不然会进行遍历

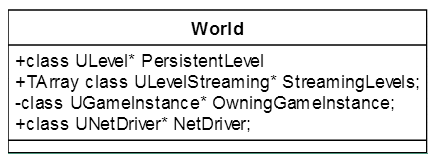
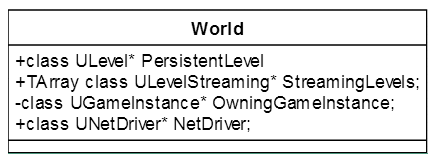
1，关卡加载

2，UE4如何实现关卡加载

3，什么是UE4的流关卡

即关卡数据可以以数据流的形式加载到游戏中，比较平稳，对当前的关卡没有影响

在UE4中，每一个World至少有一个PersistentLevel以及0-N个StreamingLevels



控制流关卡的加载总体上有两种方式

1，通过通过关卡流体积控制（Level Streaming Volume），即定制的触发器，当玩家摄像机进入LevelStreamingVolume体积内的时候，刘关卡就会加载

2，第二种通过代码，想怎么玩写就怎么写，具体教程参考官方文档！（需要通过UE4文档进行补全）

世界构成器 World Composition（暂时没啥用）

可以帮助把N个关卡用拼图的形式拼接成一个大世界地图。

使用这个功能需要在当前的WorldSetting里面勾选Enable World Composition（如果当前你的world里已经添加了子关卡，是无法开启该功能的）

对于地图切换，分类为下面的情况（这儿应该为第一部分）

1，客户端断开链接自行切换地图，服务器地图不变

2，客户端断开链接加入新的服务器地图，原服务器地图不变

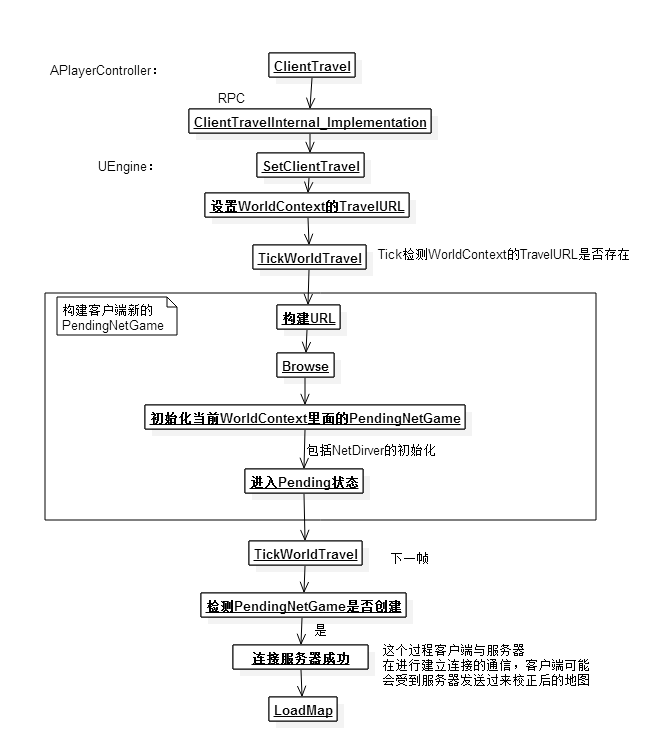
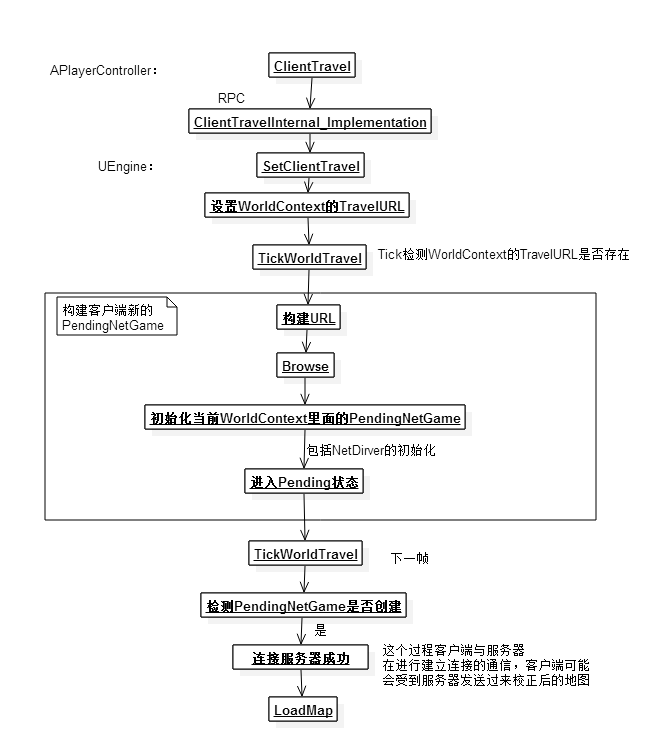
3，服务器切换地图，客户端跟随服务器切换地图

4，客户端，服务器都断开链接，各自切换到自己的新地图

ClientTravel

即UEngine::SetClientTravel(UWorld\* InWorld,const FString& URL,enum ETravelType TravelType);

这个地方的Traveltype如果是Relative,URL就会设置Protocol，Host为原来的URL里面的对应信息。 如果TravelType是Absolute，Host IP端口等信息就完全按照传入的URL设置，可能就是空的（因为我们只传入了一个地图名称）。如果当前Travel的URL没有任何IP信息，引擎就会把这个URL当成本地全局的URL（也就是不受服务器控制），因此客户端就可以自行打开一个地图，不过没有意义，相当于是客户端自己当自己的服务器，这就会造成与上面ClientTravel执行结果完全不同。



ClientTravel的主要目的是将客户端从一个服务器迁移到另一个服务器（也可以重新加入当前的服务器），这个过程一定会断开连接

ServerTravel

这个主要功能就是让服务器去加载新的地图并且通知他所有连接下的客户端都跟着他进入到新的地图去（只能在客户端运行），因为相当于是服务器自己加载地图，所以IP地址没有什么关系了。

Tips:这个在编辑器下表现会比较奇怪，客户端无法与服务器建立连接而一直处于Pending状态。表现上就是客户端角色卡住的效果。

Browse

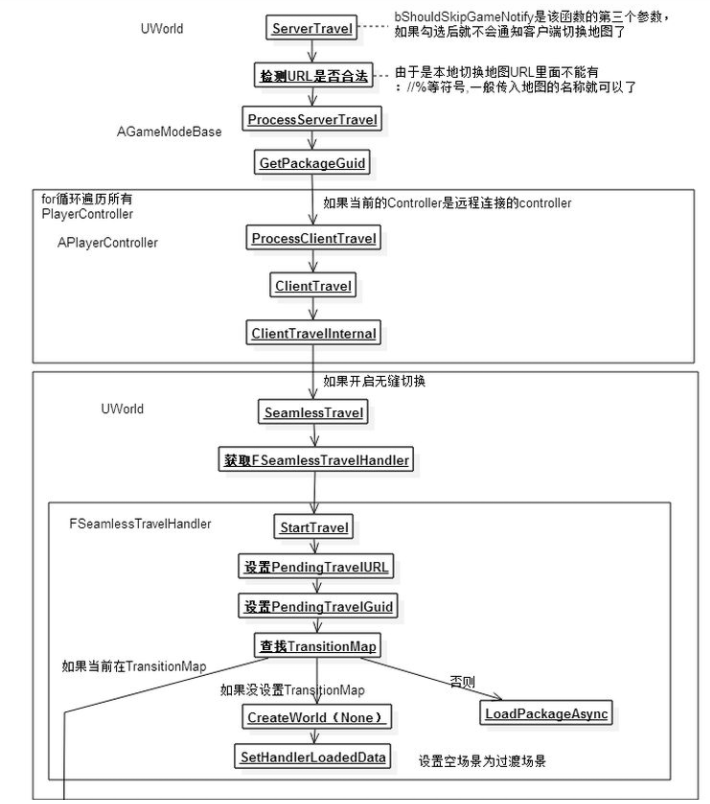
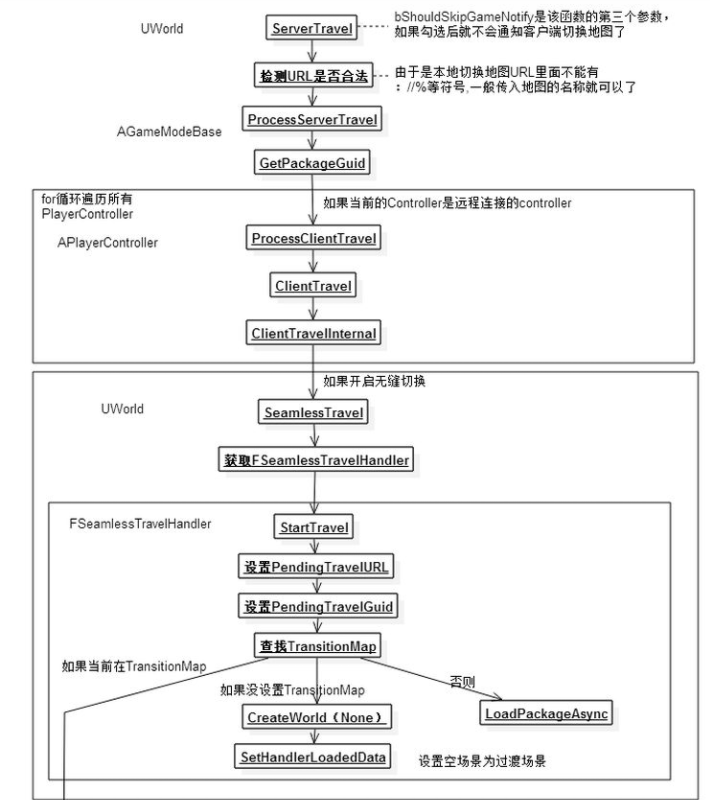
每次调用ClientTravel以及ServerTravel的时候都一定会用到，即加载地图的硬重置。

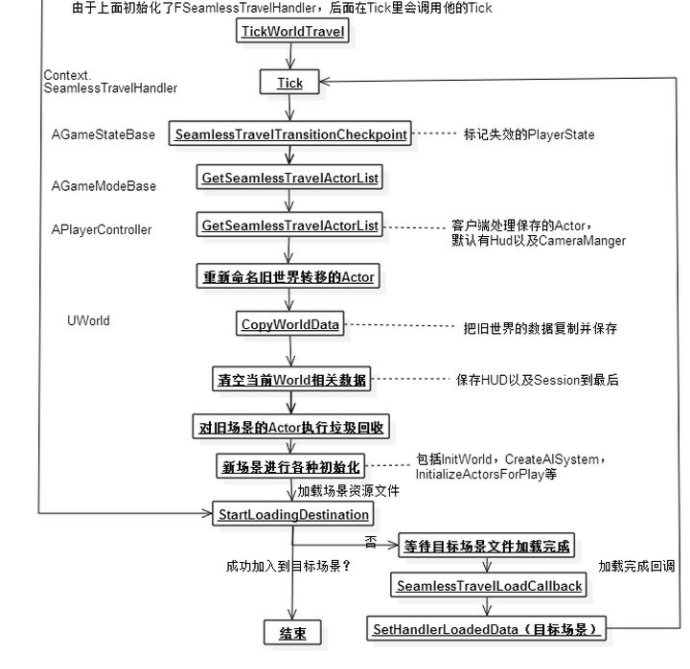
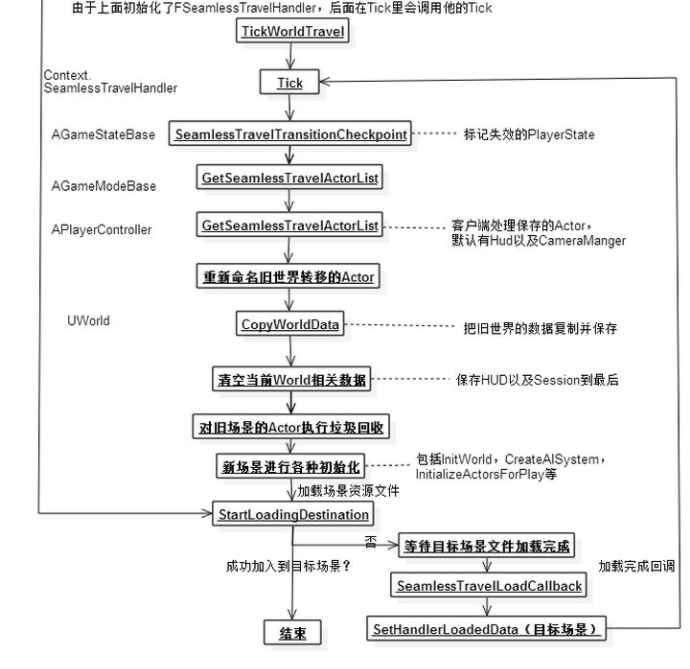
无缝地图切换

UE4的无缝切换即在不断开连接的情况下切换地图（相当于切换PersistentLevel）

一般来说，无缝切换的正常切换只有一种：服务器切换地图，客户端与服务器在保持连接的情况下也跟着切换地图。使用情景类似于一个房间服务器，玩家们从A场景结束任务后开始新的任务。

无缝切换流程





无缝切换时如何保持连接

无缝切换通过一个FSeamlessTravelHandler类Tick操作覆盖了原来的Browse操作，这个过程中不会直接释放地图资源，而是通过一定机制将Map数据通过拷贝进行转移，可以看到在函数FSeamlessTravelHandler::CopyWorldData里面会将当前的World的NetDriver赋值给要加载的World