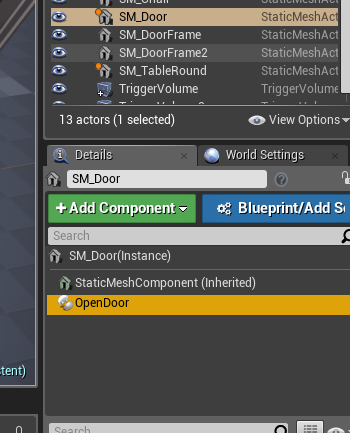
先从最近的来，

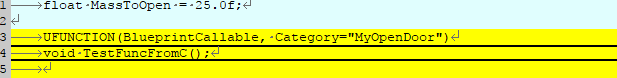
Blue print 调用 C++中的方法：

将 原来 actor 转成 blueprint

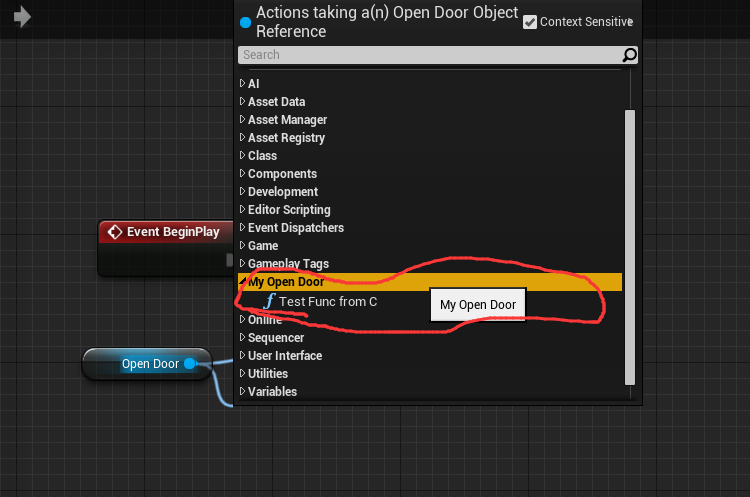


这个 openDoor 是 C++的 component, 自然能够调用原来C++中的方法， 但是要对方法进行修饰

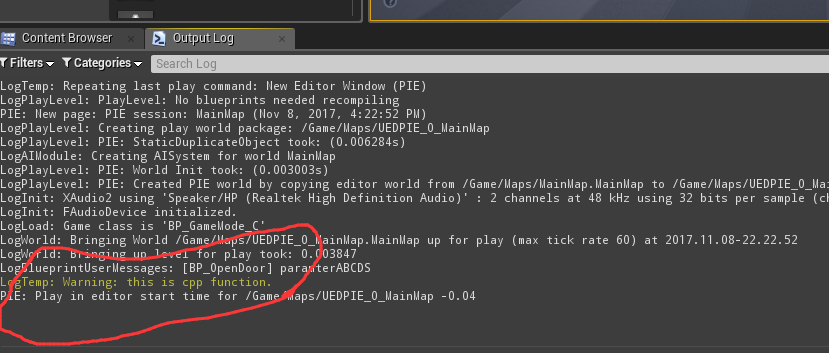
加：再在声明该函数的时候， 使用如下方式：



其中的 MyOpenDoor， 是 后面 blueprint 中 的方法的种类：

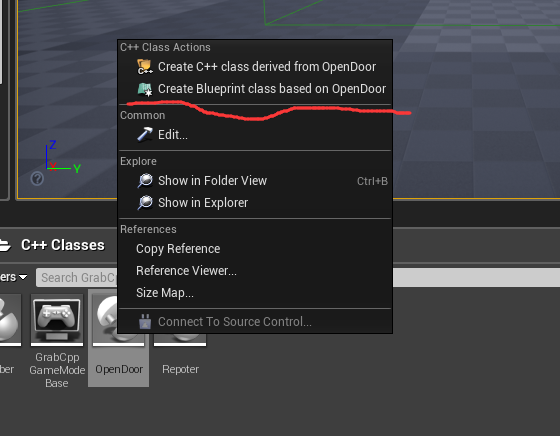


This function called successfully:

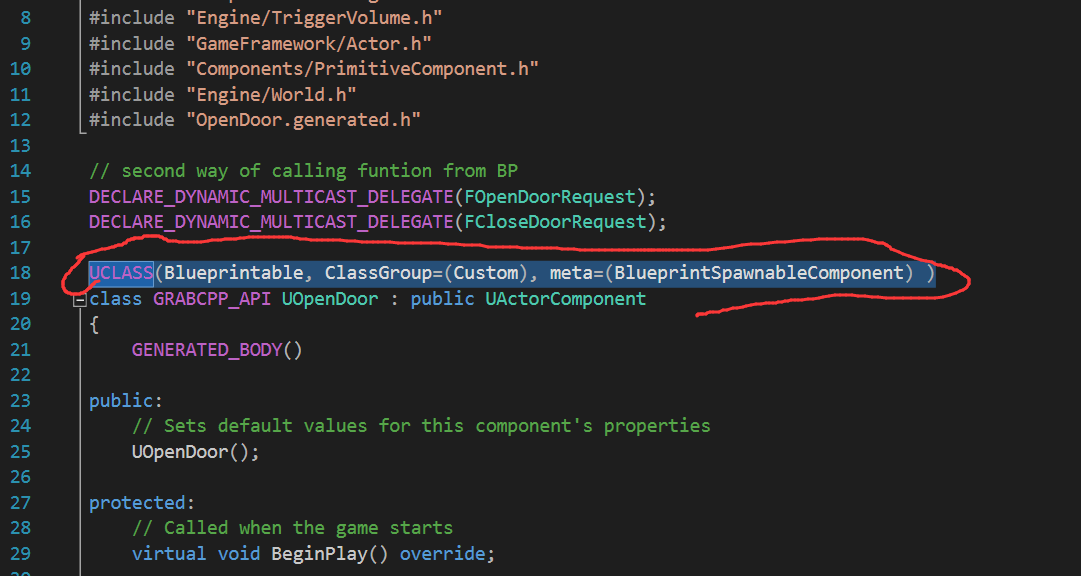


C++ call function from blueprint:

这种blueprint 是比较特殊的， 它是由 C++的 CLASS生成的， 类似于一个C++类的子类

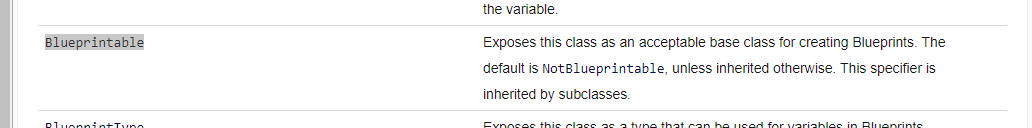


但是，默认生成的 C++ 类，是无法直接生成 Blueprint的，需要在 C++ 类声明处添

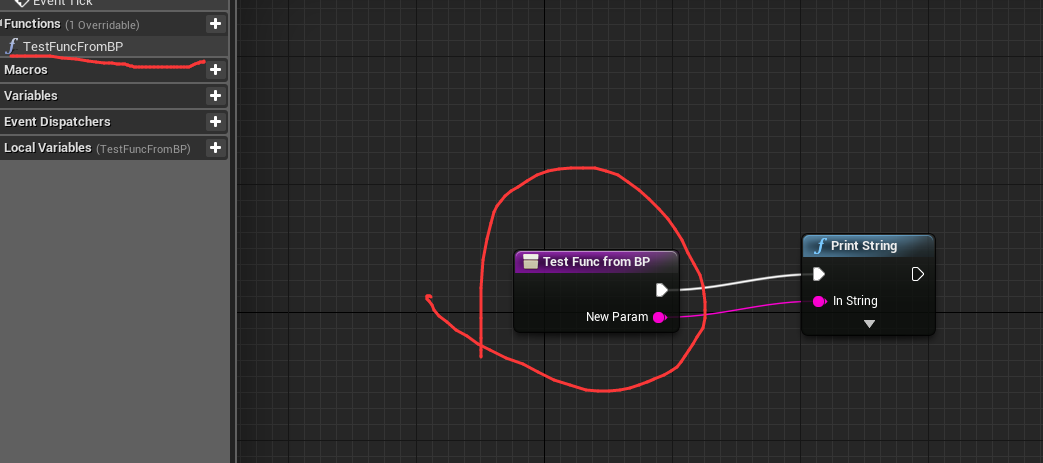


UCLASS的其他用法和参数可以查看 官方文档：

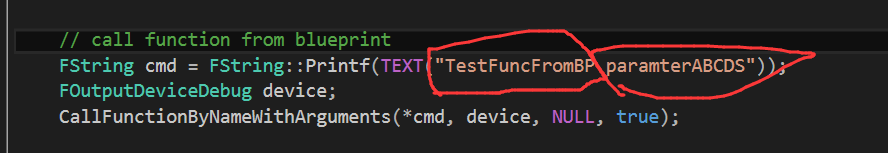
<https://docs.unrealengine.com/latest/INT/Programming/UnrealArchitecture/Reference/Classes/>



在生成的blueprint 中声明方法， 再在 其原来的 C++类中调用， 这个就可以理解了：

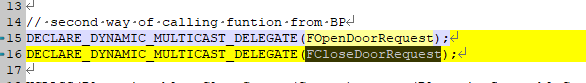


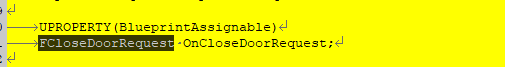
C++中调用：



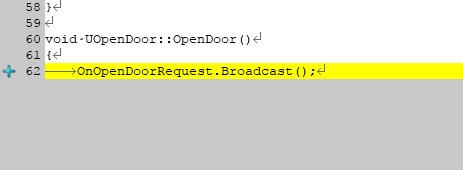
分别是 函数名 和参数

然后C++调用blueprint中的方法，还有另外一种方式， 通过 在 C++ 中广播，而在 blueprint中响应该event, 可以完成此操作：

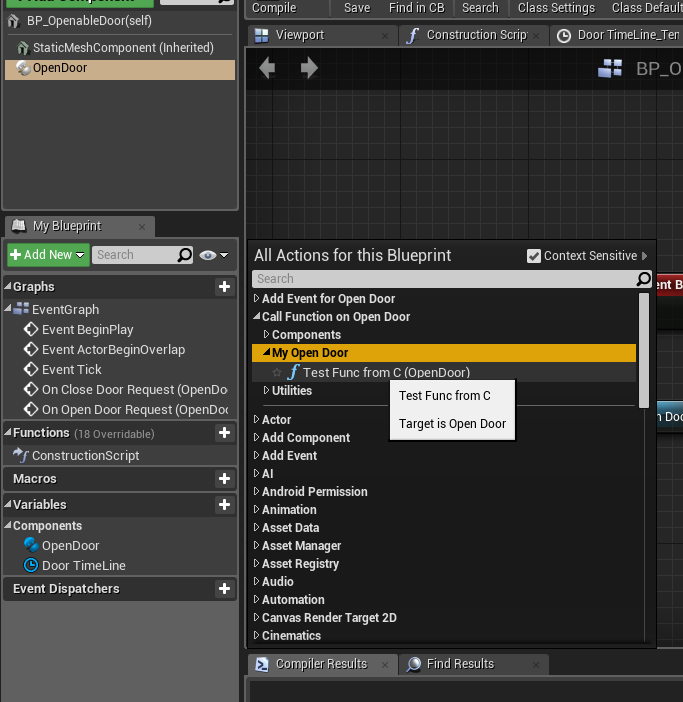




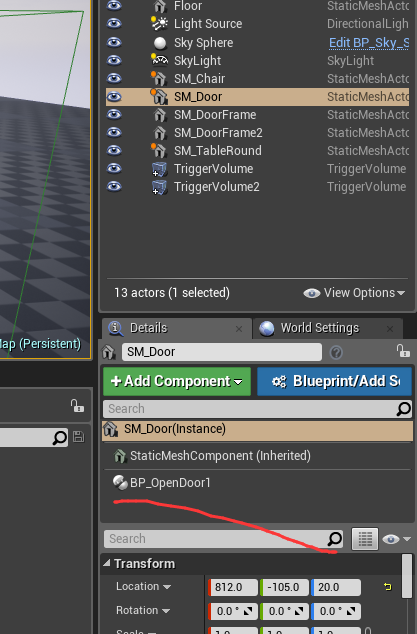
在需要调用 blueprint的地方 broadcast 该事件就好了：



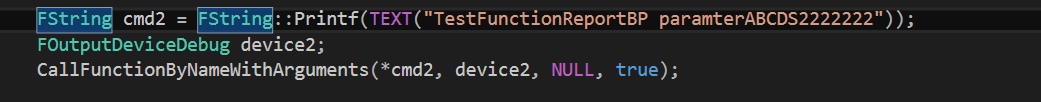
选中 openDoor组件， 同样这个buleprint是原来 的 ACTOR component 转换而成的，这个原本的 actor component 本来就有 opendoor component ，这个是 C++的class，



问题，将blueprint 放置到 某一个Actor 下面， 到底什么意思？



到底 C++调用 blueprint 是如何操作的



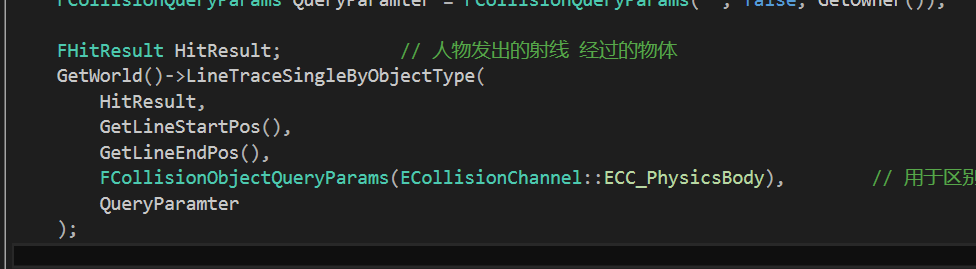
这段代码的 到底如何运行的？

这个上面已经解释了

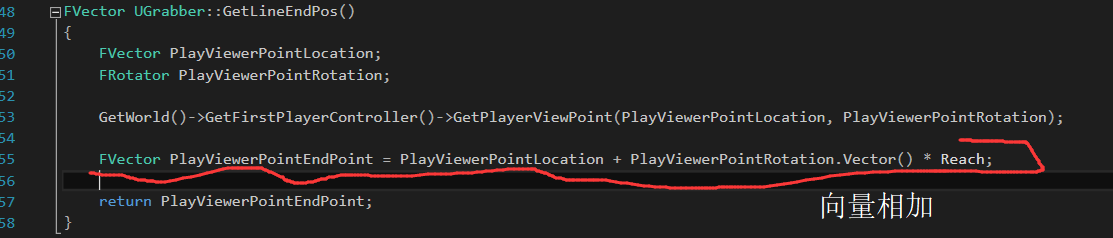
明天再继续，先睡觉了

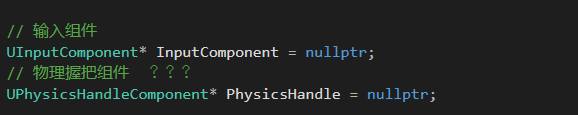
关于 grab 的操作：

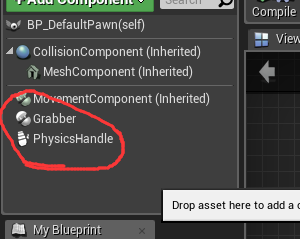
先说射线，核心的 函数就是：



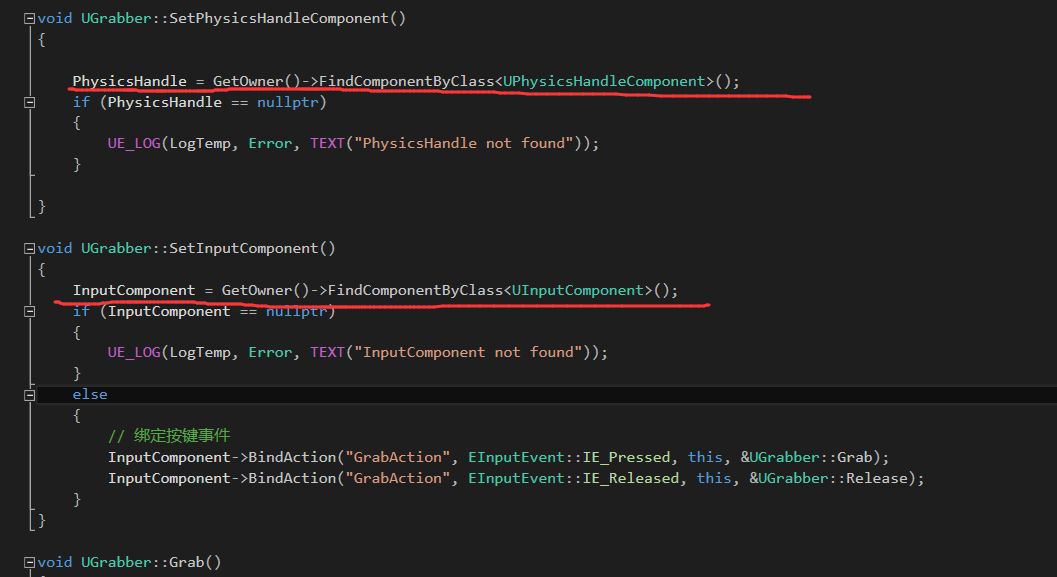
其中的 HitResult 用来存储 被射线阻挡的 物体的信息， 其中的 GetLineEndPos 用到了向量相加的含义，这个我应该知道，不必细究：

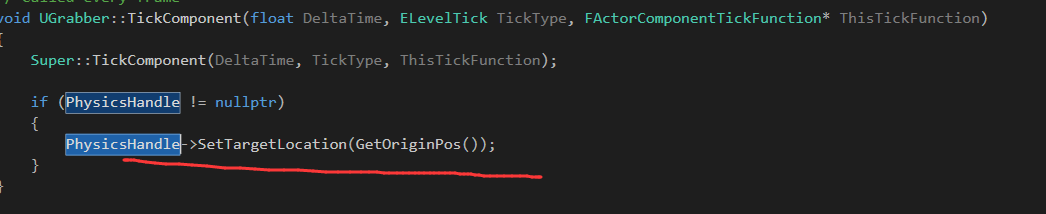


还有两个比较重要的地方， 在当前 pawn 上， 需要有响应键盘事件，和 抓取的操作， 这些都需要先添加一个 component 到 该 pawn 上面， 然后， 再添加

这两个组件， 注意在获取之前， 需要现在 pawn 的 blueprint中添加 

这样的话， 就可以在CPP 代码中获取了，



获取的方式， 是通过 FindComponentByClass 而最终 看到 被抓的物体，可以随便改变位置， 是通过实时刷新通过PhysicsHandle被grab 的物体 ，

设置其位置。

OK, 就这么多， 后面每次学习一个， 都整理一次， 加强记忆和理解