问题：

⚪AI move to一直往天上飘？

已解决：UNavMovementComponent的GetActorFeetLocation()导致，默认返回updatecomponent的location-actor bound的Z，如果Actor的mesh component的原点不在中心，就会产生偏差导致初始位置偏低，移动时往天上飞。解决方法，override GetActorFeetLocation函数

⚪Move to到了位置一直不结束。

已解决：没有触发OnActorBump，改Actor的collision；改完后没移动到就停止：取消勾选moveto节点的重叠半径选项

⚪C++类细节面板是空的

已解决：重新reparent一下，或者重新编译

⚪Enhanced input 输入值为0

已解决：不要加其他触发器，默认的就是1

⚪函数参数有原生C++型的时候函数用UFUNCTION标记，编译报错Unable to find 'class', 'delegate', 'enum', or 'struct' with name …

已解决：UFUNCTION去掉就能过了，反射不支持原生C++类，以下是GPT的回答：

在 Unreal Engine C++ 中，你可以使用 UFUNCTION 标记一个函数，但它的参数和返回类型需要遵循一定的规则。特别是，Unreal Engine 反射系统（Reflection System）对函数参数类型有严格的限制。基本原则是，函数参数应该是引擎能够识别和处理的类型。支持的参数类型：

Unreal Engine 反射系统支持以下类型的参数：

基本类型：如 int32, float, bool 等。

FString：字符串类型。

引擎的结构体类型：如 FVector, FRotator 等。

引擎的对象类型：如 AActor\*, UObject\* 及其子类。

TArray：引擎的数组类型。

不支持的原生C++类型

原生C++类型（如 std::string, std::vector 等）是不能直接用在 UFUNCTION 参数中的，因为这些类型不在引擎的反射系统支持范围内。

⚪打包没问题，运行报错LowLevelFatalError。。。LowLevelFatalError

已解决：打包前是livecoding编译的，需要退了用vs编译再打包

⚪打包后连不上steam

已解决：改配置后没有重新generate

⚪XXX已经在obj中定义

已解决：RPC函数定义没加Implementation！！！

⚪为什么C++调用destroy杀不掉？？一开始是没写super begin，加上了还是不行

⚪RPC问题：

被调用RPC函数的actor，必须要拥有链接，简单来说它需要是playercontroller，pawn之类的类型

⚪Subsystem找不到定义

已解决：调用GetSubsystem的地方没有包含对应头文件

⚪运行时开启自定义深度无效：

已解决：需要MarkRenderStateDirty ()

⚪SpawnActor的旋转有问题：

已解决：FRotator需要初始化：FRotator::ZeroRotator，而不能直接FRotator()

⚪PawnMovement , AI move临近终点时速度方向会突然抽搐？

问题原因：因为之前移动组件改了GetActorFeetLocation()函数，现在碰撞出了问题，导致移动时脚位置又上天了，速度就不水平，tick里面setrotation就错了；

解决方案：unit改为自定义box碰撞，不使用mesh的碰撞，然后取消之前移动组件的修改，OK了

⚪声明一个变量时报错error C4458: “XXX”的声明隐藏了类成员

已解决：不是因为别的，因为变量名已经在父类定义过了

⚪如何获取蓝图派生类的属性值？

已解决：可以通过UClass->GetDefaultObject()获取CDO，然后获取变量值

⚪碰撞问题各种BUG

⚪导航，动态修改，避让问题

CharacterMovmentComponent可以开RVO，Pawn怎么避障？

笔记：

⚪meta = (WorldContext = "WorldContextObject")

将调用此函数的object赋值给WorldContextObject参数

⚪PRAGMA\_DISABLE\_OPTIMIZATION

PRAGMA\_ENABLE\_OPTIMIZATION 可以关闭优化！！避免调试时变量被优化！！

⚪编译报错无法识别的外部链接，可能是build.cs里没有添加相应的模块

⚪非dynamic delegate声明不需要加参数名称

⚪Subsystem:

可以蓝图继承，加上UCLASS(Abstract, Blueprintable, BlueprintType)

C++中获取，用World->GetSubsystemBase(Class)

⚪UE C++接口声明，定义和使用：

<https://blog.csdn.net/ZFSR05255134/article/details/119459043>

⚪C++重载、重写（覆盖）、重定义（隐藏）

⚪C++ 接口定义的函数，如果不是BlueprintCallable或者BlueprintNativeEvent，接口cpp中必须有实现，否则编译不过，如果是则不用

⚪meshcomponnet的setmaterial和meshcomponent->getmesh->setmaterial是不一样的