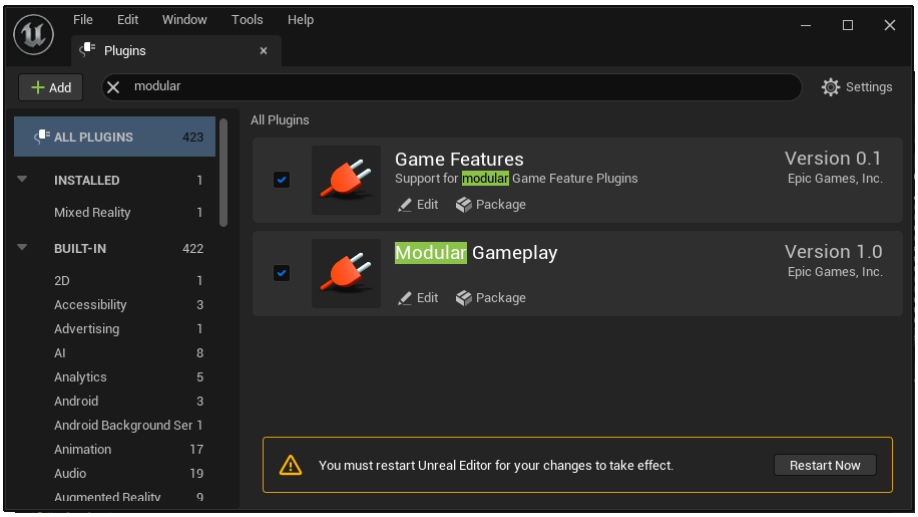
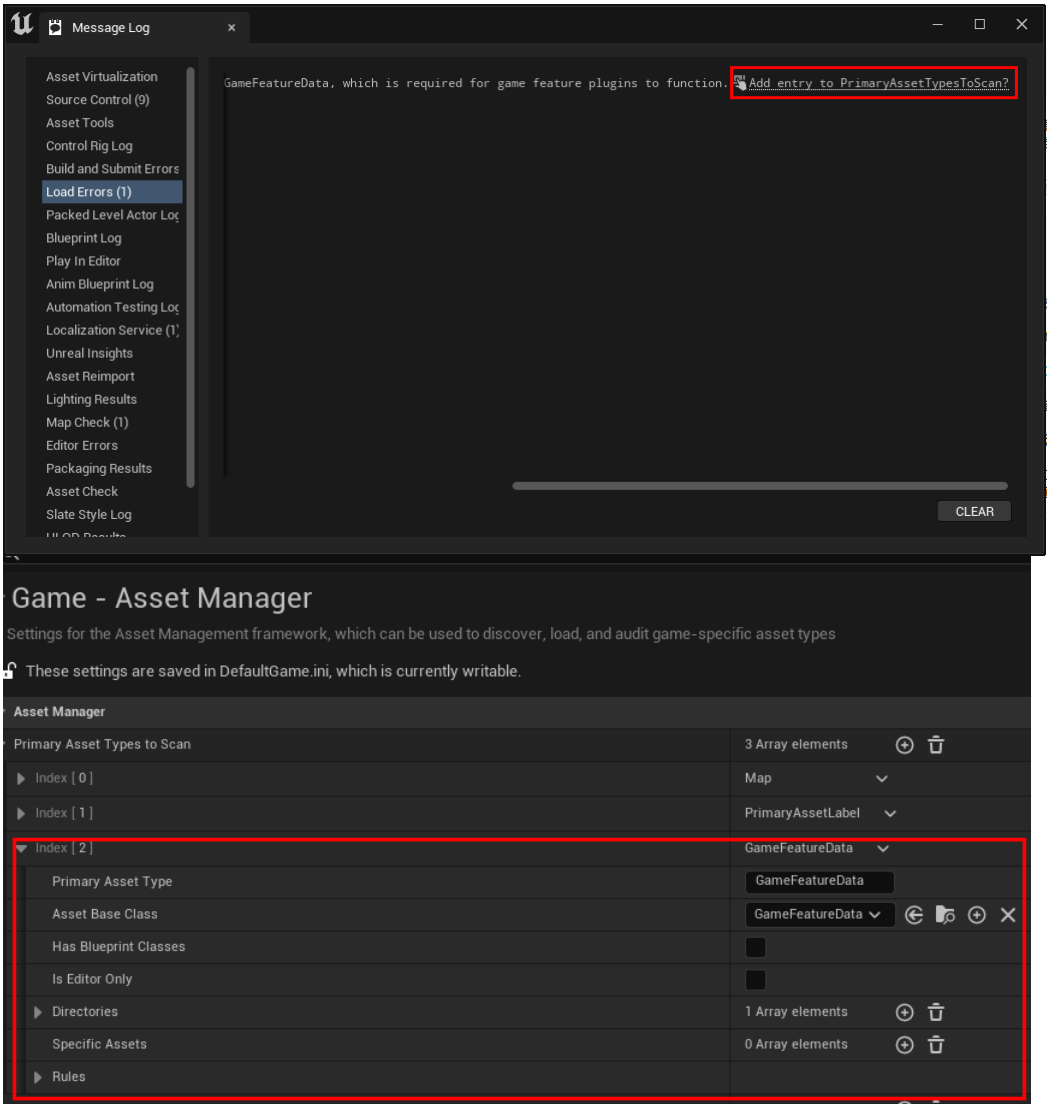
https://zhuanlan.zhihu.com/p/470184973

启用插件

首先启用两个插件：Game Features和Modular Gameplay

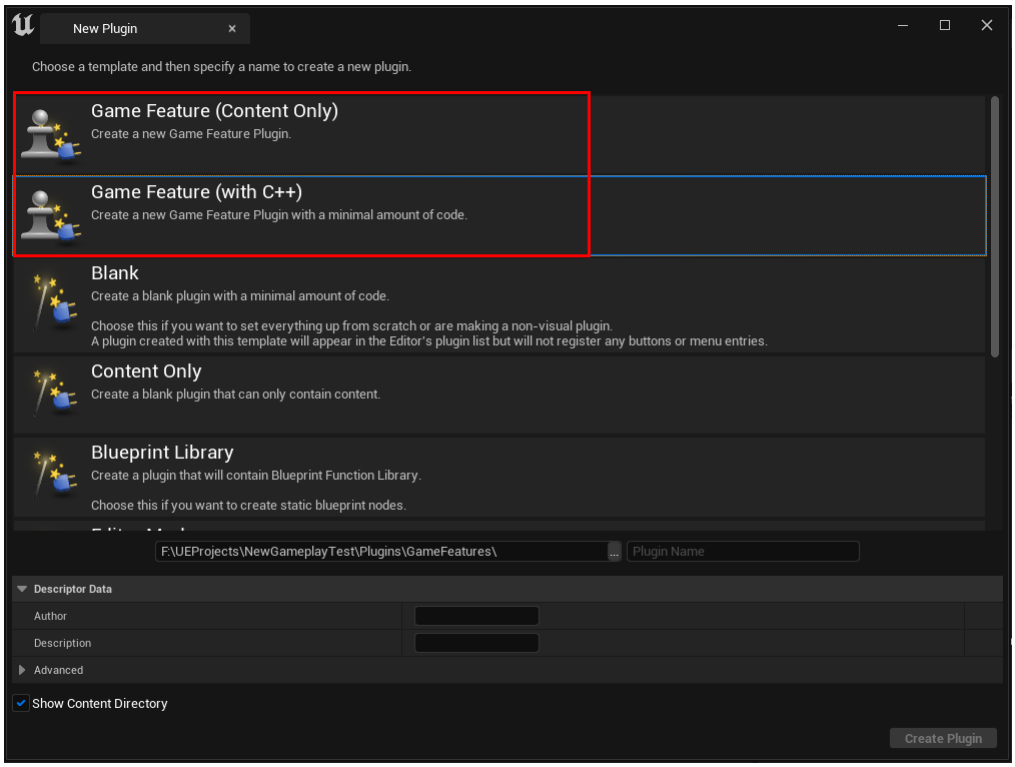


重启编辑器后，会出现弹窗，提示需要将GameFeatureData添加为PrimaryAssetDataScan。直接点击弹窗消息最后的链接，直接添加。



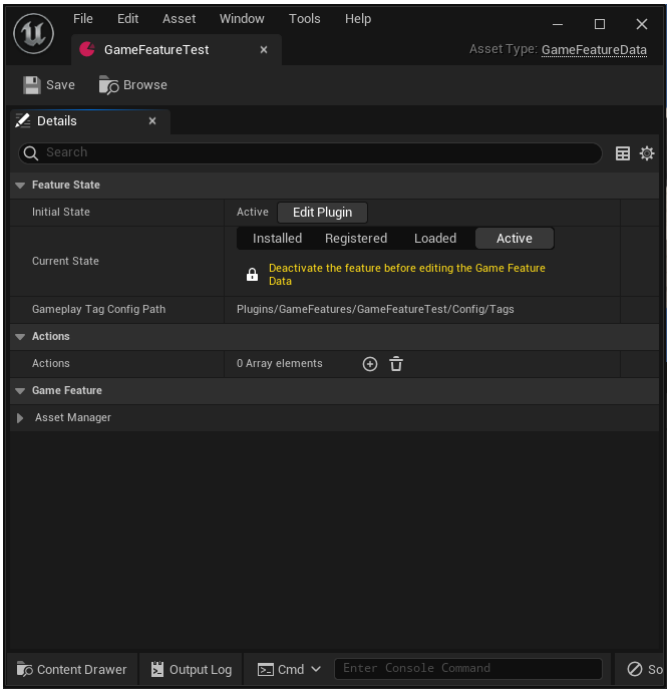
创建GameFeature插件和模块

可以通过创建插件的方式创建一个GameFeature，有两个可选插件模板，一个是仅包含资产的插件，另一个则是也包含C++的GameFeature插件。



GameFeature注册表

插件创建完成后，在Content目录中会有一个DataAsset资产文件用于注册和管理该GameFeature的功能。



Initial State

可以编辑插件信息，也可以设置该GameFeature在编辑器和游戏启动时的初始化状态。

Current State

当前的Game Feature状态，包含四个状态：

Installed: 当前模块仅存在于存储硬件，不加载

Registered: 插件中的资产可以被发现，但不加载

Loaded: 加载插件中的资产到内存，但是不激活

Active: 激活插件功能

Actions

功能使用

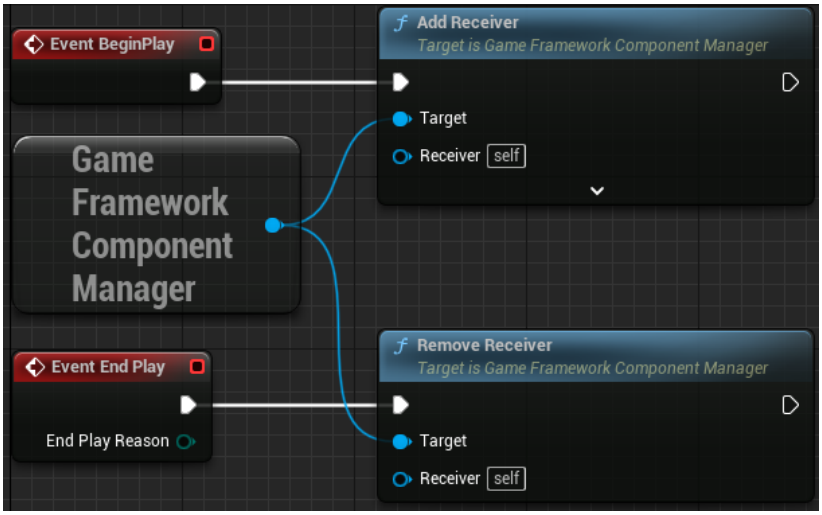
仅在注册表中为指定类型添加组件或功能并不能直接起作用，还需要对被处理的类注册Receiver，下例为给ACustPlayerController添加一个组件的设置。

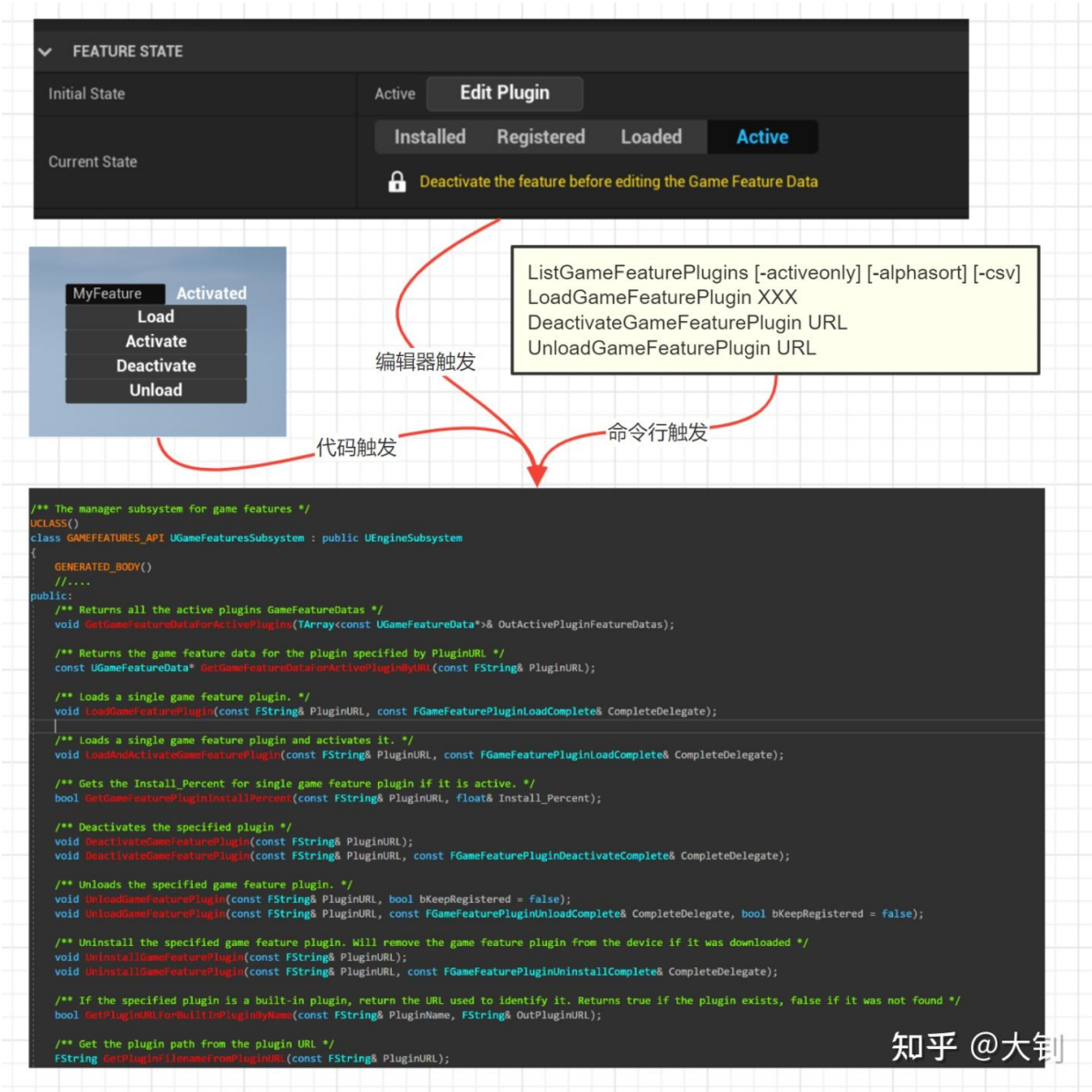
在ACustPlayerController中注册Receiver：

```
void ACustPlayerController::PreInitializeComponents()
{
    Super::PreInitializeComponents();
    UGameFrameworkComponentManager::AddGameFrameworkComponentsReceiver(this);
}

void ACustPlayerController::EndPlay(const EEndPlayReason::Type EndPlayReason)
{
    UGameFrameworkComponentManager::RemoveGameFrameworkComponentReceiver(this);
    Super::EndPlay(EndPlayReason);
}
```

蓝图：





Modular Gameplay

UGameFrameworkComponentManager(GameInstanceSubsystem)

AGameFrameworkComponent(ActorComponent)

Game Features

UGameFeaturesSubsystem(EngineSubsystem)

UGameFeatureData(PrimaryDataAsset)

UGameFeatureAction(Object)

UGameFeaturePluginStateMachine(Object)