## 角色类

创建BaseCharacter类

创建Character类，public，命名为AuraCharacterBase

创建AuraCharacter类和AuraEnemy类继承自上面

给AuraCharacterBase添加一个骨骼网格体组件，并将骨骼网格体组件作为根节点

还需要有一个武器的骨骼网格体，将其附着到角色骨骼，这需要保证身体骨骼网格体有用于这个功能的socket

在protected中

UPROPERTY(EditAnywhere, Category = "Combat")

TObjectPtr<USkeletalMeshComponent> Weapon;

然后在构造函数中，初始化该武器

Weapon = CreateDefaultSubobject<USkeletalMeshComponent>("Weapon");

然后将武器连接到socket上

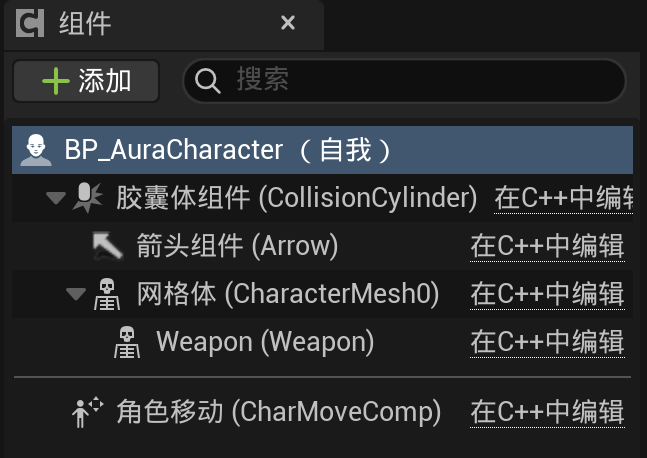
Weapon->SetupAttachment(GetMesh(), FName("WeaponHandSocket"));

武器不应该有任何碰撞，

Weapon->SetCollisionEnabled(ECollisionEnabled::NoCollision);

然后创建AuraCharacter的蓝图类

打开之后如下



我们先给网格体提供骨骼网格体资产

在提供武器的socket时，在C++中定义的socket名称是WeaponHandSocket，所以需要进入Aura的角色骨骼网格体，找到一个合适的骨骼（如hand\_l），右键选择添加插槽，并给插槽命名为WeaponHandSocket

然后右键插槽，选择“添加预览资产”，选择Aura的武器，预览一下。预览的时候可以设置一下插槽的相对位置。

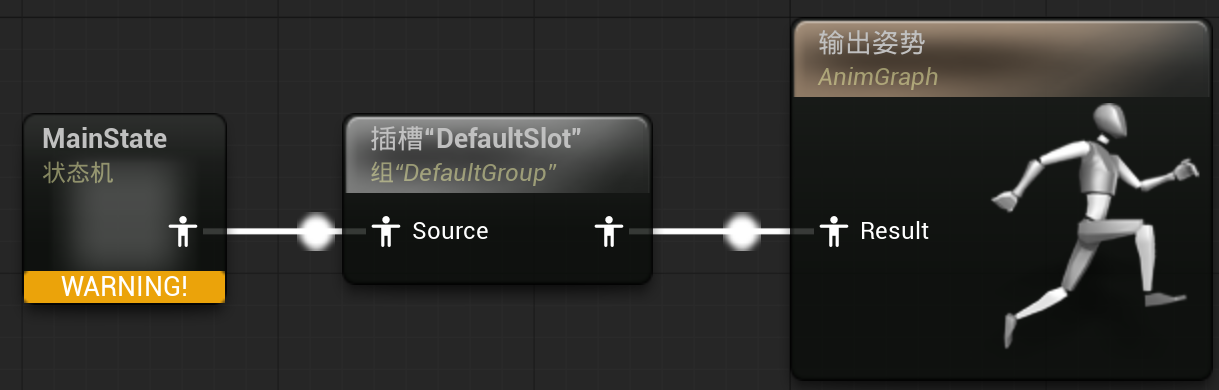
然后再在BP\_Aura中添加武器的骨骼网格体

复习

然后创建BP\_Goblin

## 动画蓝图类

动画蓝图中的slot插槽用于播放动画蒙太奇





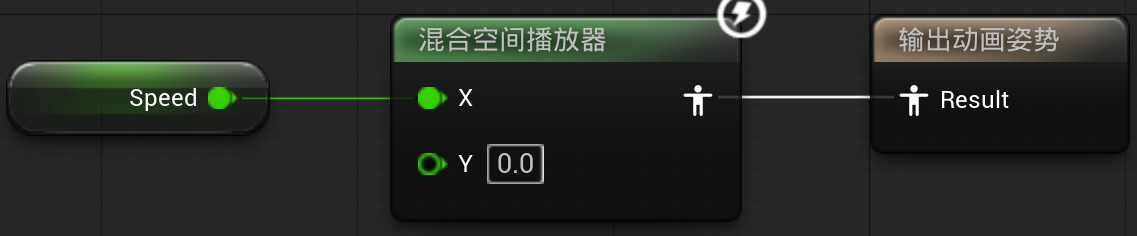
状态机中

当创建敌人类的动画蓝图时，因为敌人的骨骼不一定一样，所以我们创建的是模板动画蓝图。

动画蓝图的设计和Player的一样，都是状态机后链接Slot后输出，状态机都是直接连接状态并命名为IdleWalkRun

差别体现在状态内部了

在模板的状态中，使用Blendspace Player节点，通过变量提供参数。这个变量会在子类中赋值。



接下来创建模板的子类动画蓝图。这时就可以选择特定骨架了

然后在子类的“资产覆盖编辑器AssetOverrideEditor”中，设置父类中的混合空间播放器



复习

实现BP\_Goblin\_Slingshot

也可以看一下，也有别的骨骼网格体，试一试