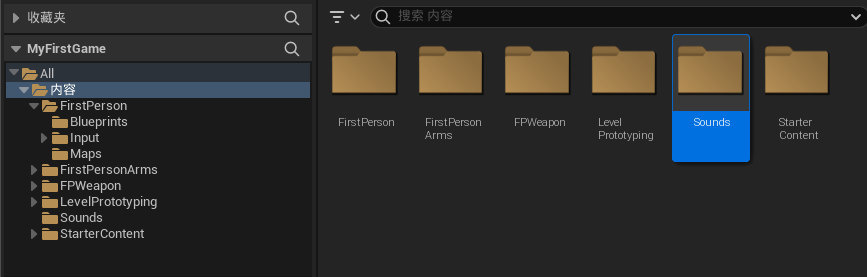
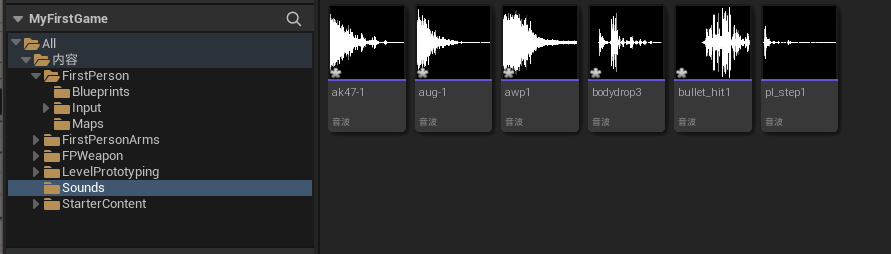
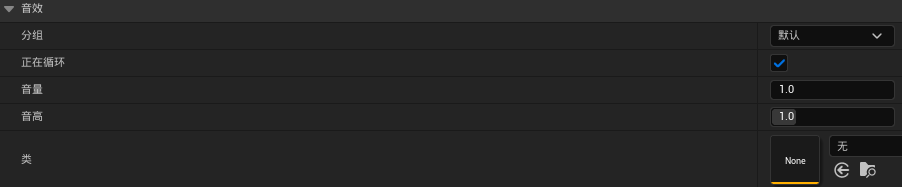
需要用到前置blog：blog-虚幻5-添加头部摆动效果

**安置音效**  
首先在内容文件夹下创建Sounds文件夹用于保存音频资产  
  


值得注意的是，step音频因为是玩家步行过程中发出的音频，所以要对step音频进行循环处理

双击打开step音频，在“音效--正在循环”（Sound--Looping）处勾选

这样我们在角色蓝图中播放声音时，它就会不断循环



之后进入第一人称角色蓝图  
**落地音效**

节点的使用

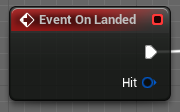
**EventOnLanded**节点

//在坠落落地时调用，基于命中结果执行操作

//注意：在此事件中运动模式仍在“坠落”，当前速度值为落地的速度

//还需考虑OnMovementModeChange（），因为运动模式变为新模式（很可能步行）可能会使用一次

//目标是角色



**Switch on Int**节点

//选择一个与输入值匹配的输出



此处需要添加两个引脚，链接落地的两个音效

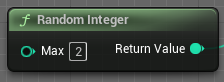


**Random Integer**节点

//返回同一分布的数字，其位于0和“最大值-1”之间

//目标是Kismet数学库

自行设置最大值



**Play Sound 2D**节点

//播放不含衰减的音效，最适用于UI音效

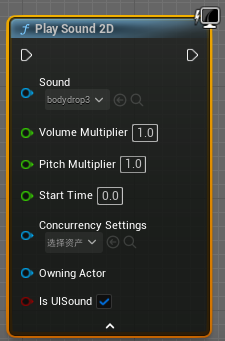
//发射之后即遗忘

//并非复制

//目标是Gameplay静态



这里要展开节点，并且给Sound形式参数一个“落地声音”的WAV格式的声音文件实际参数

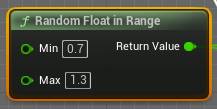


这里我们给Pitch Multiplier（音高乘数）形参，用于乘以音高的线性标量。通过给Pitch Multiplier一个范围内的随机参数，使得Play Sound 2D节点绑定的音频文件在播放的时候有随机的不同表现效果

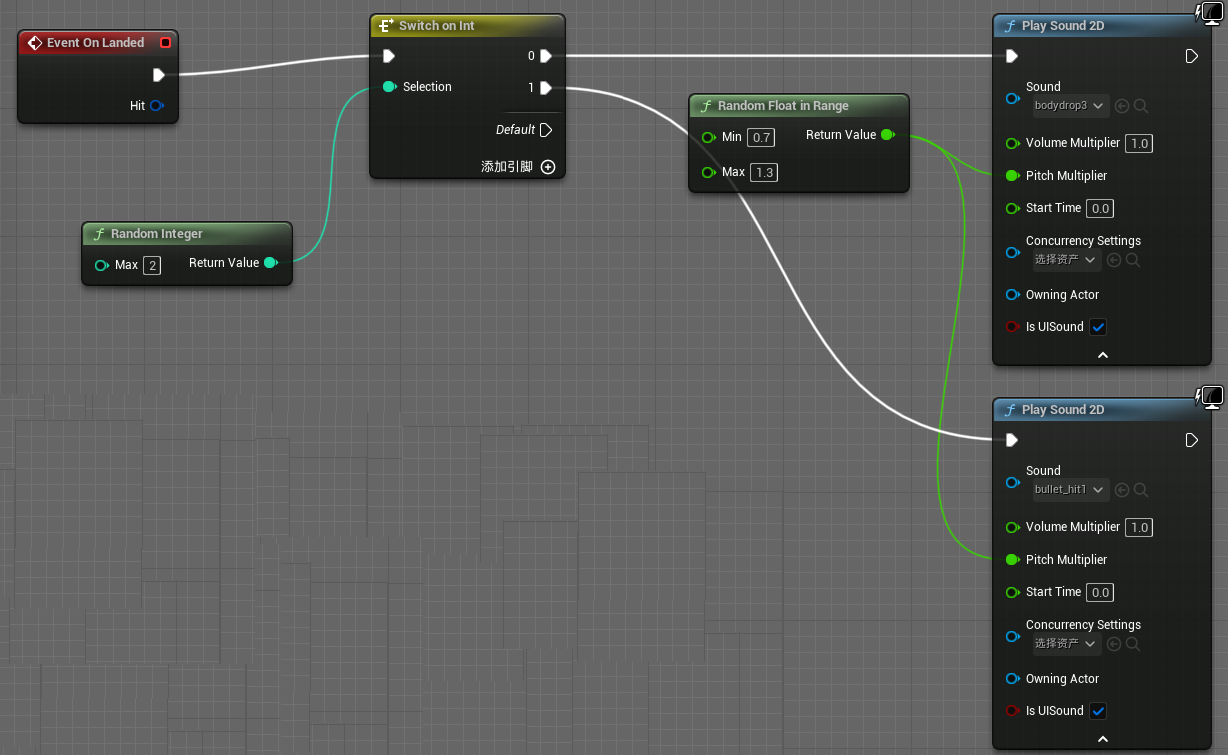
**Random Float in Range**节点

//生成最小值和最大值之间的一个随机数字

//目标是Kismet数学库

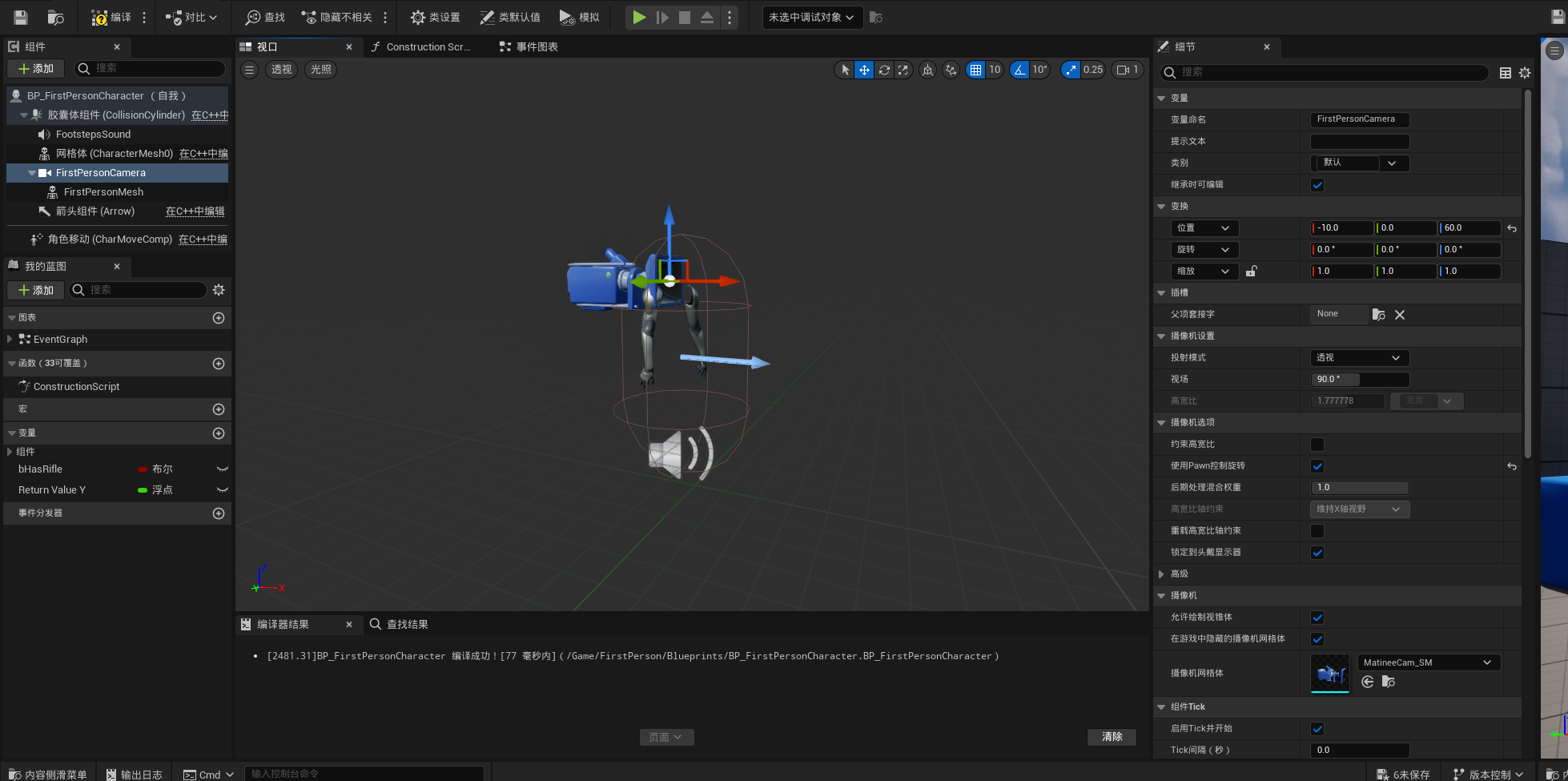


节点结果

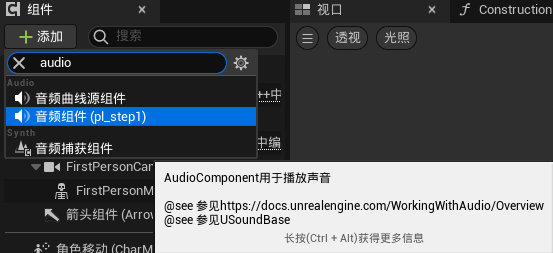


**移动音效**

首先进入FirstPersonCharactor蓝图中，选择“视口”界面（注意，按理说此时不应该有人物胶囊下面的声音表示和左侧的FootstepsSound（实际上这两件时一个东西），但是这是blog不是教程）



在左上角的“组件--添加”中，添加“音频组件”（搜索关键字：audio），选择和图片中注释一样的那个选项，并起个名字，此处叫FootstepsSound



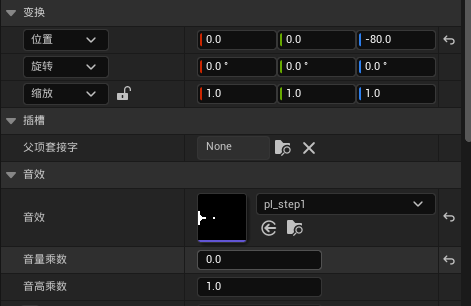
点击创建好的该组件，在右侧的“细节”中修改其部分数值

这里修改了：

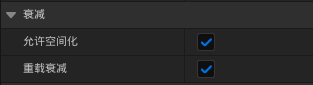
位置：用于保证声音发出是从脚下发出的

音效：选择刚刚进行循环设置的step音效进行绑定

音量乘数：把音量乘数调整至0，使得在未移动时发出的声音为0.调整声音大小在蓝图中调整



在衰减中给“重载衰减”打勾



衰减（音量）和衰减（空间化）中，给我修改的（右侧有重置选项）参数进行修改。其中：内部半径和衰减距离表示的是音量在向外传播过程中发生衰减的情况。自行更改。



结束后进入蓝图的事件图表界面，安置节点

节点的使用

**FootstepsSound**节点

直接在组件中把这一选项拉进事件图表中，即可创建FootstepsSound节点

//音频组件，对象引用



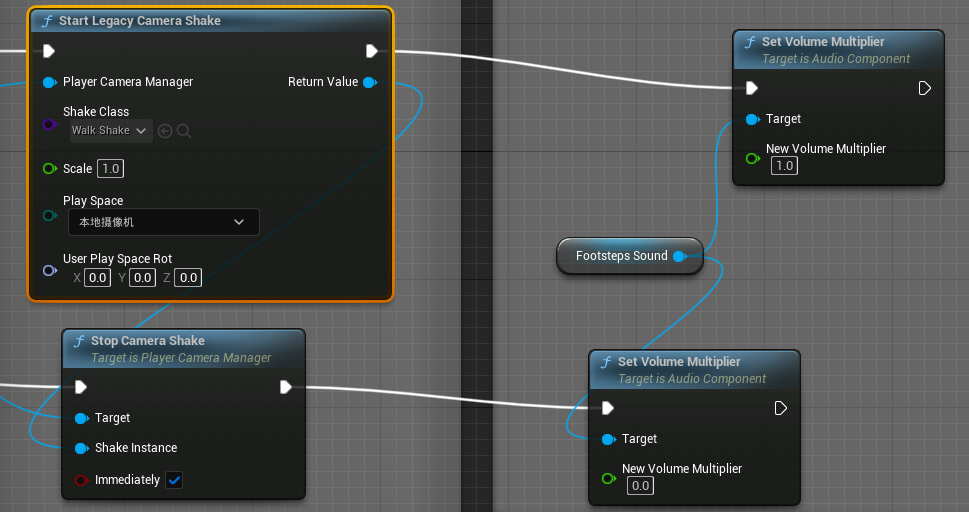
**Set Volume Multiplier**节点

//设置一个新的音量乘数

此处要注意，应当将步行期间的音量设置为非0，静止期间步行音量设置为0



与“头部摄像机晃动”的一系列节点进行链接



（该图片左侧的两个节点是属于头部摄像机晃动的，右侧的两个节点是步行声音的）

注意

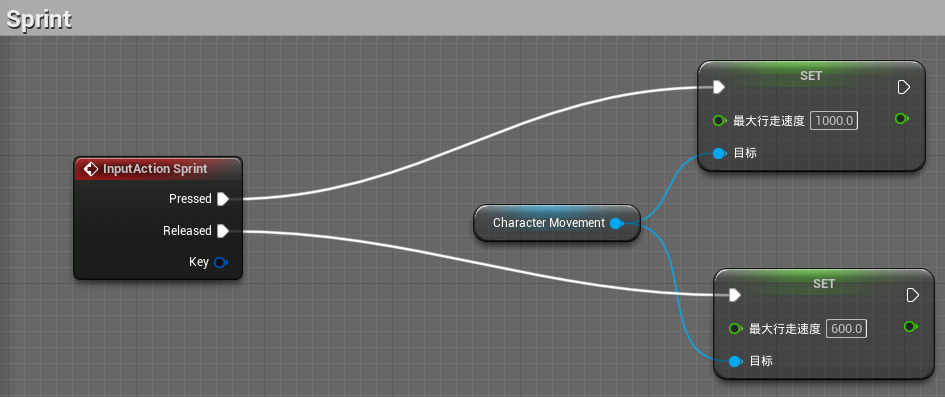
步行期间的声音，我们让”步行期间”事件与“步行导致摄像机发生运动”事件进行绑定，这样可以简化流程

如果不这样绑定，依旧可以用类似于“头部摄像机晃动”的蓝图框架进行编程，只是会增大工作量

**冲刺音效**

当我们在跑步时，声音会加速，在Sprint框架中添加声音节点

Sprint初始框架

新增节点

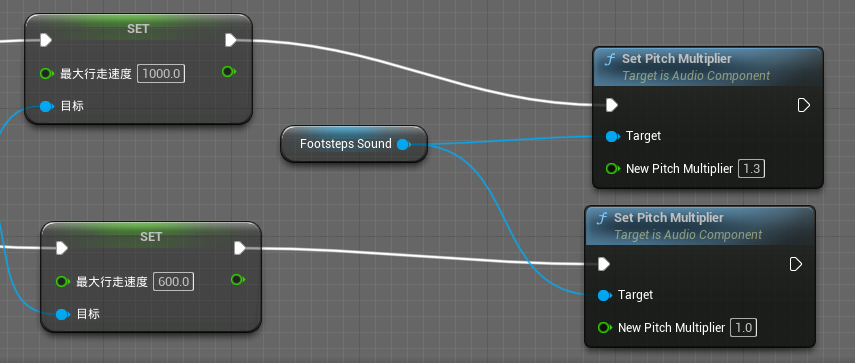
**Set Pitch Multiplier**节点

//设置一个新的音高乘数

//目标是音频组件



节点结果



此处要注意，应当将冲刺期间的乘数设置为大于1的数，步行期间乘数设置为1