自定义重力的方式是通过角色移动组件的SetGravityDirection函数实现的，是在5.4版本提供的全新节点。

创建第三人称项目，创建Actor，命名为BP\_GravityField。这是重力场，指的是当角色在某种条件下，会受到该重力场的影响。

现在我们创建这个BP，来模拟星球。我们将前面提到的条件，设置为当角色与重力场相碰的时候，会受到重力场的影响。

BP中加一个SphereCollision和一个球体。碰撞体积应当略大于球体。



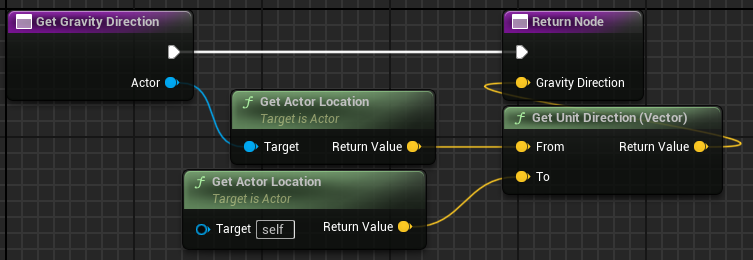
在BP中实现函数GetGravityDirection，形参为Actor类对象。



后面我们可以基于BP\_GravityField创建子类，也可以在这个BP里实现。

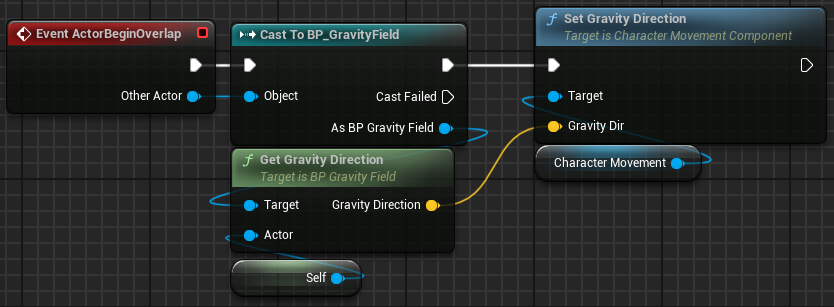
当然，如果是子类来实现的话，SphereCollision和球体Mesh也可以在子类中添加。

下面给出就在本类中实现的方法。



接下来实现碰撞改重力。

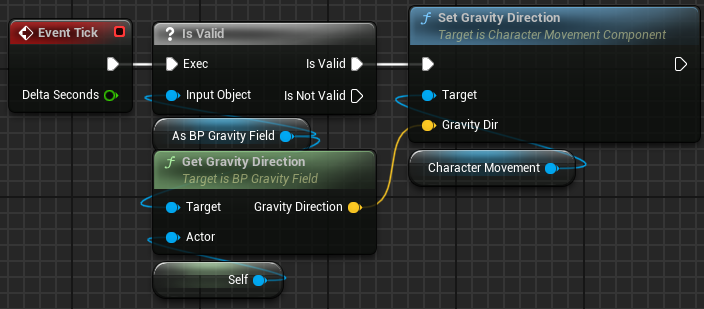
在BP\_ThirdPerson中，当EventActorBeginOverlap时，如果OtherActor是GravityField，则调用Get函数并将返回值提供给Set函数



这是一次性设置。如果是球形星球，需要每帧设置GrivatyDir。

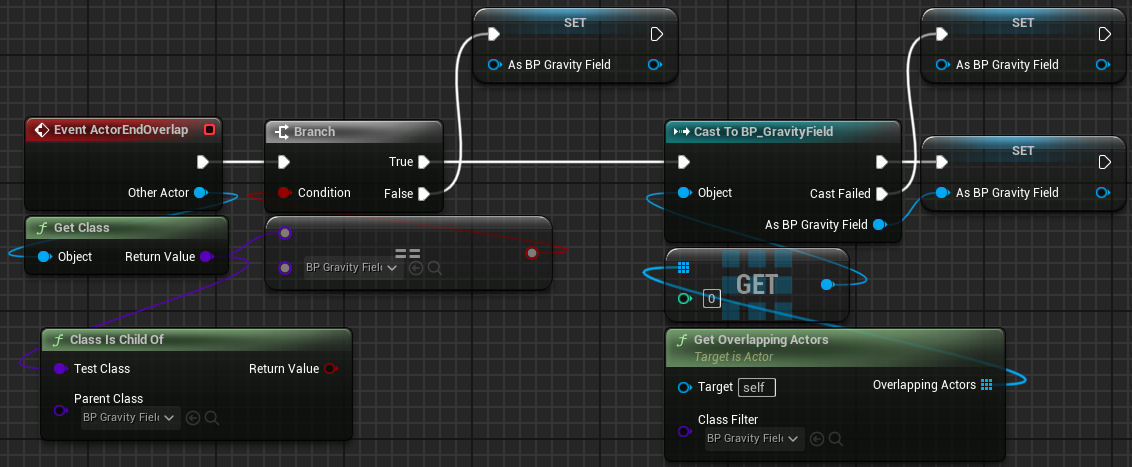
我们将碰撞到的GravityField提升为变量。

在EventTick中



顺便弄一下EndOverlap。

在EndOverlap时，如果有很多重力场，我们需要检查我们是否在任何其他重力场内



如果创建子类，则使用ClassIsChildOf函数

现在的情况是：

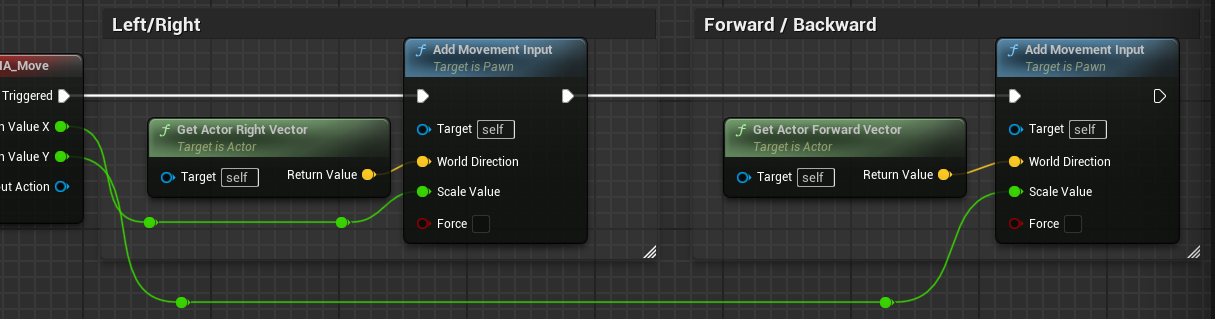
摄像机旋转有点奇怪、角色不能移动到比赤道还高的位置。

在BP\_Player中创建自定义事件，命名为OnGravityChanged。我们在BeginOverlap时调用这个事件。

现在我们需要修改一下角色移动功能

第三人称自带的移动，是目前正在使用GetControlRotation，ControlRotation是世界旋转。当摁住W时是告诉你在世界上前进。

所以现在我们不能再依靠ControlRotation来进行移动，而是通过ActorRotation

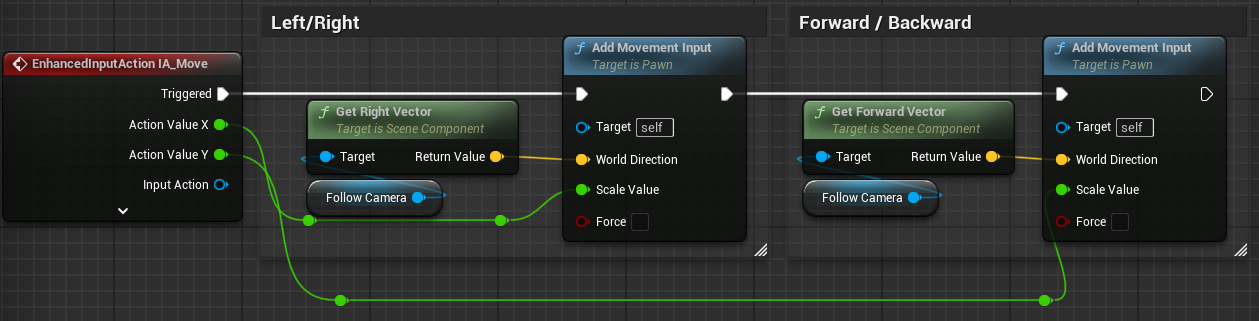


现在，A和D键不再是平移而是旋转。但是摁住W可以在球上任意位置移动了

下面解决摄像机和角色控制的问题。

我希望角色可以向着摄像机朝向的方向进行前向移动，我需要得到摄像机相对于重力的方向。

也很简单，直接用摄像机组件的对应Vector即可。



当然，这之后还是有问题的。因为角色的实际移动是受到摄像机影响，所以当角色站立地面而摄像机垂直地面时，角色的理论移动应该是朝着地面的方向走，但是地面方向并不能移动，所以实际移动就是朝着地面方向极其缓慢的移动。解决方案自己实现吧。