#### 子弹的发射、Cascade特效与Niagara入门

通过点击一个按键，实现一个子弹的发射。同样的可以发射光球。这里我们实现：发射出一个能够发光的球体

在内容浏览器中创建一个Actor类蓝图，作为子弹的obj；创建一个材质，作为子弹的表面



打开材质，对材质进行编辑

在材质图表中，摁住键盘3的同时点击鼠标左键，创建**3 vector constant**节点



可以点击左下角的黑色进行颜色选择，也可以通过编辑常量值XYZ对颜色进行编辑。其中，X、Y、Z分别代表R、G、B，当数值处于[ 0 , 1 ]时，XYZ表示表现颜色的RGB，当数值大于1时，表现为发光

对常量编辑好后，将该节点连接至自发光颜色

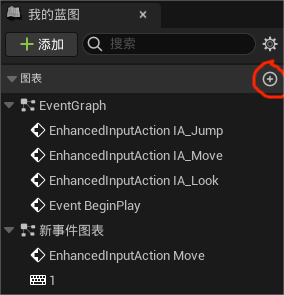


打开BrightBall蓝图，创建一个球体，将球体的材质设置为刚刚编辑的材质。作为子弹，将球的碰撞预设调整为NoCollision

##### 在场景中创建球体

首先，我们希望角色可以通过摁下1键来创建一个球体，所以我们要进入到角色蓝图中，实现蓝图

为了减少一页图表的蓝图内容，可以在“我的蓝图--图表”中创建新事件图表



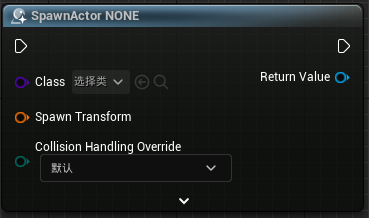
创建**1**节点



当摁下1键时，我们希望能够从类生成Actor，即

**Spawn Actor from Class**节点

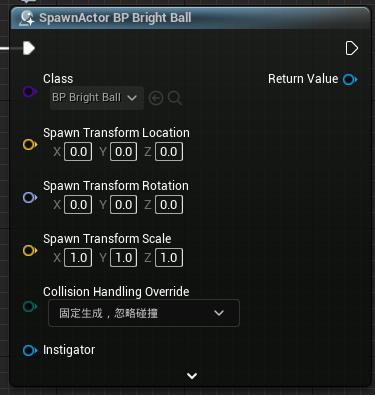
//尝试使用指定的变换来生成新Actor



将类选择为BrightBall

此时点击编译，我们会发现有报错，显示Spawn Transform必须链接一个输入

我们先不给它输入，我们将SpawnTransform右键，分割结构体引脚，此时再点击编译就没有报错了。因为分割结构体引脚后自己就有了默认的值。

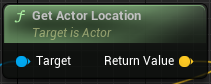


此时进入游戏，摁1键就能在世界的Location=（0，0，0）的位置生成一个光球。但是现在我们希望球生成的位置不是指定地图位置，而是相对角色位置。

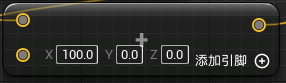
我们先得到一个角色自身的参考，即**Get a reference to self**节点



我们需要得到自身的位置，即：**Get Actor Location**节点



此时我们可以将返回值传递给Spawn Transform Location，进入游戏，即可在自身位置创建一个球。但是我们希望球能够在自身某个方向（XYZ方向）创建球体，可以从GetActorLocation的返回值处拉出引线，创建**+**节点



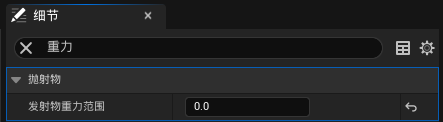
此时可以通过给XYZ进行相加赋值来改变球体创建位置

##### 二、使球体移动

进入球的蓝图，创建“发射物移动组件”，作为根组件的子组件。

此时进入游戏，摁1键就能生成一个光球，但是生成的光球会掉落，这是因为球有重力。我们进入球的蓝图中，将球组件的“物理--启用重力”关闭，此时球就不会往下走。

同样的，我们可以在发射物组件中可以修改“抛射物--发射物重力范围”。将其修改为0表示不受重力，将其修改为负数表示收到的是升力。



接下来设置移动速度。我们可以在发射物组件的细节中设置速度。但我们希望球能够跟着主角朝向去飞

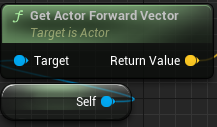
我们在BrightBall蓝图的事件图表中，拖入发射物组件，创建节点，并设置该发射物组件的速度，即**Set Velocity**节点



将Set与Event Begin Play进行连接

然后我们回到角色蓝图，获取到角色的Rotation，即**Get Actor Rotation**节点，并将返回值给SpawnActor的Rotation参数赋值

回到BrightBall蓝图，我们需要获得一个他自己的Reference，获取它的向前向量，即**Get Actor Forward Vecto**r节点



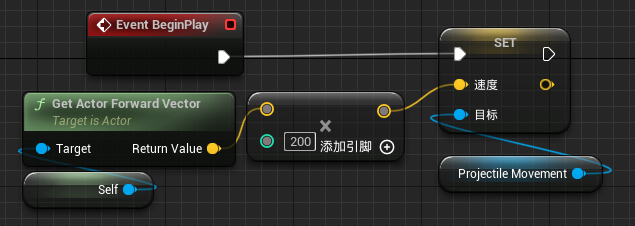
这里的Self是球，因为角色调用了球，并把自己的朝向作为一个向量传递给了球

我们给Forward Vector的返回值进行一个乘积



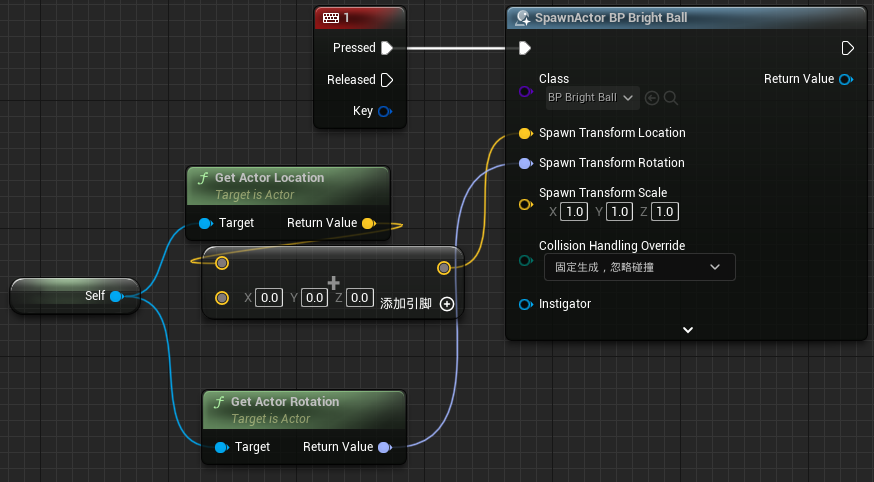
将左下角的参数右键，改为整数形式，选择一个合适的速度值，即可。当然这里也可以将整数形式的参数右键选择“提升为变量”，进行变量操作

BrightBall蓝图最终结果



此时我们进入游戏，创建的球体的速度方向即和角色方向一致

回到角色蓝图，当前角色蓝图如图所示



此时调整+节点的参数，进入游戏，会发现球的创建位置并不是角色的朝向方向

我们需要设计一下球的诞生位置的算法

观察Get Actor Forward Vector节点

**Get Actor Forward Vector**节点

//从场景空间中的此Actor获取向前（X）向量，长度1.0

//目标是Actor



所以诞生位置的算法是：

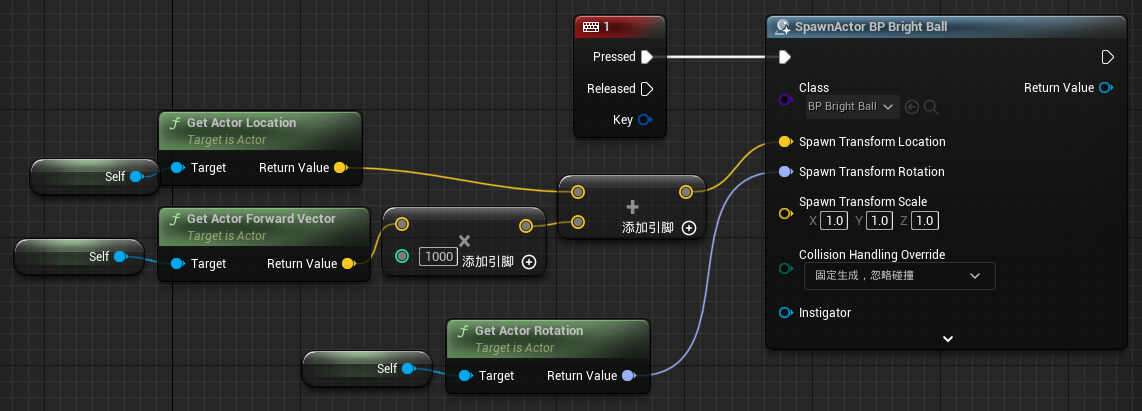
球的诞生位置=角色自身位置+角色朝向\*诞生距离

球的诞生位置是SpawnActorfromClass（BrightBall）的SpawnTransformLocation参数

角色自身位置为GetActorLocation

角色朝向为GetActorForwardVector

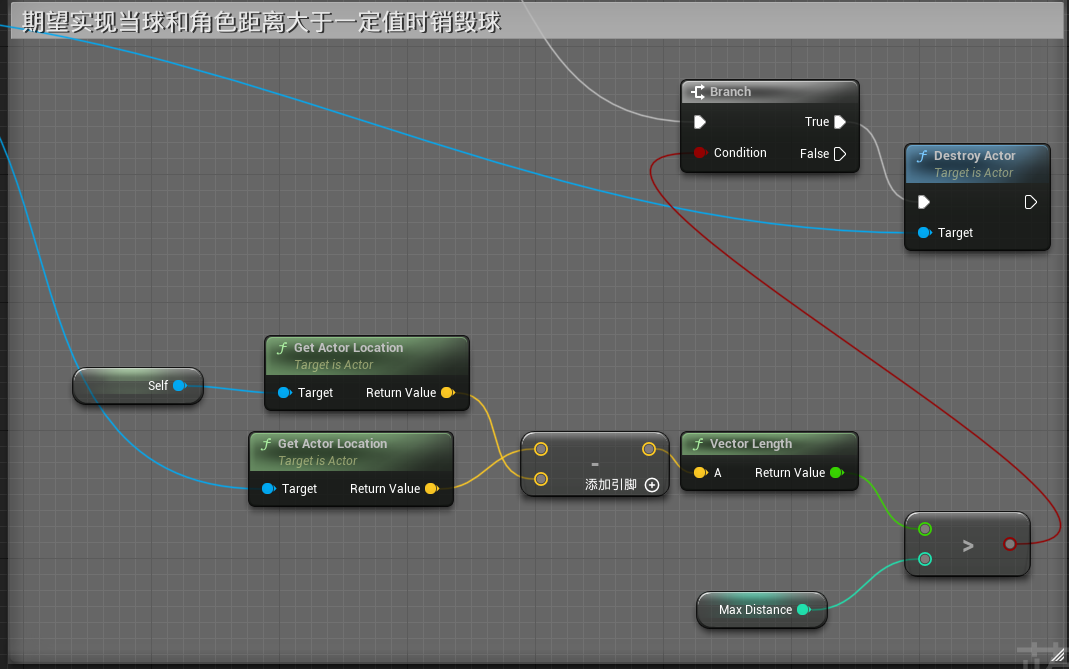
即



此时便完成了球体的创建。

###### 三、销毁球体

这里我们选择让球在运动一段距离后被销毁。这里并没有直接实现球在运动一段距离后销毁，因为未知原因



但是我们可以换个思路，让球在运动一段时间后销毁，时间=距离/速度。这样就间接实现了球在运动一段距离后销毁的需求。但性能肯定不足。

如果要销毁某个东西，建议不要直接销毁，因为有些Actor有触发事件，销毁可能导致某些触发事件失效。建议先隐藏，过段时间再销毁。

###### 四、粒子特效

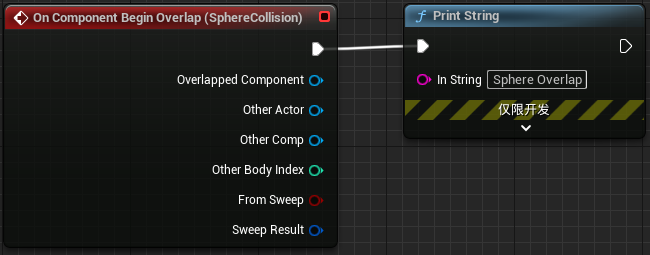
首先在虚幻商城中下载并导入“FX Variety Pack”到项目中，然后在在内容浏览器中创建Special Effects文件夹，将导入的特效文件移动至Special Effects文件夹内

接下来我们看一下导入的特效文件，在SpecialEffects/FXVarietyPack/Map文件夹中有一个地图，可以进入该地图查看特效

在SpecialEffects/FXVarietyPack/Particles文件夹中由粒子特效。播放特效最简单的方式就是将特效文件直接拉入世界中。现在我们希望发出的子弹能够附带P\_ky\_waterBall粒子特效，并在命中时触发P\_ky\_waterBallHit特效。

首先在内容浏览器中选中P\_ky\_waterBall特效，然后进入BrightBall蓝图，选择添加组件。此时会自行优先显示出“Niagara Particle System Component”选项。但是我们现在不用它，我们要用Cascade Particle System Component，创建好组件。此时在视口界面中直接就会显示该粒子特效。（当然可以选择创建的组件，在组建的“细节--粒子--模板”中选择粒子特效）这时当角色创建球体时会附带特效。

接下来实现命中的P\_ky\_waterBallHit特效。首先给球设置一个Sphere Collision组件。然后创建Sphere Collision的OnComponentBeginOverlap节点，使用Print String节点判断碰撞是否发生



（若想知道球碰到了什么东西，可以将Overlap节点的Other Actor返回值赋值给Print String节点的In String参数。这时我们会发现：当发射球体过快的时候，下一个球会和上一个球发生碰撞。这里我们将Sphere Collision的大小调小即可）

此时运行游戏，当角色打出球体的时候就会触发Print String，而与其他大部分物体发生碰撞的时候并不会Print，鉴定为在创造球的时候，Sphere会和Sphere Collision发生碰撞。解决方法是在发射球体的蓝图中找到“碰撞--碰撞预设”，将物体响应中的WorldDynamic值设置为忽略。



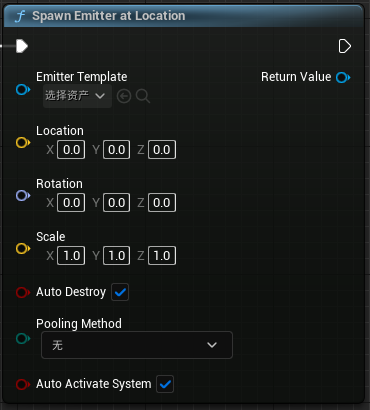
而与其他大部分物体发生碰撞的时候并不会Print，是因为球碰撞到的物体并没有**生成重叠事件**。要想让球碰到某物体时触发碰撞事件，应当在世界视口中选择该物体，在细节中找到“碰撞--生成重叠事件”，打勾，即可在球OnComponentBeginOverlap的时候触发事件。

接下来实现BallHit特效。当Overlap事件触发时，要“在位置处生成发射器”，即

**Spawn Emitter at Location**节点

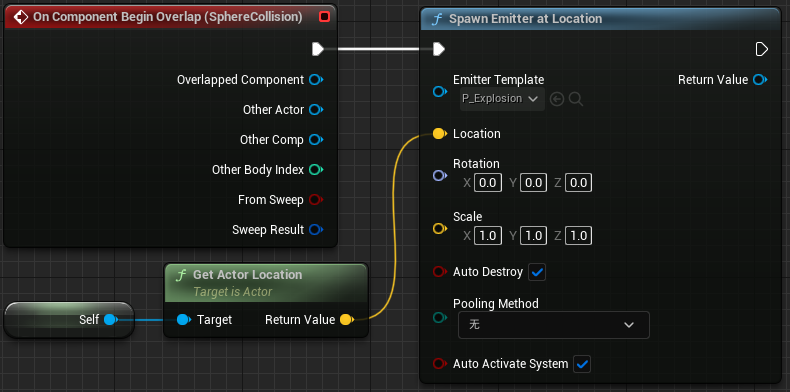
//在给定位置和旋转处播放指定效果，发射后即遗忘。效果完成后系统将关闭，不进行复制。

//目标时Gameplay静态



EmitterTemplate为命中的特效（注意特效的播放类型。爆炸类型的特效会在播放完后消失，但持续类型的特效会持续播放）

Location为自己所在位置，即Get a reference to self和Get Actor Location

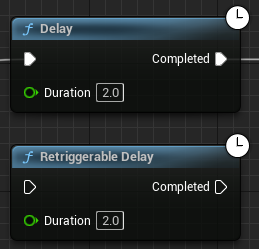


###### 五、Niagara入门

在内容浏览器中右键可以直接创建Niagara系统，选择“来自所选发射器的新系统”。之后会有许多模板，双击合适的模板后选择完成。（这里可以选择Hanging Particulates。这是一个场景效果，可以长时间存在的。如果是Omnidirectional Burst，这是一个爆炸效果，一段时间后就结束了）这里我们选择Hanging Particulates。这是一个很常见的效果，可以作为比较好的场景氛围特效。双击打开该特效，进入特效系统总览。我们可以在该总览中对特效进行设置。如在“粒子生成--Initialize Particle”中可以设置例子的颜色，在“发射器更新--Spawn Rate”中可以设置粒子生成数量

###### 六、新节点介绍

延迟与可再触发延迟



在Delay期间，再触发不会被识别，也就是说在Delay期间的再触发不会刷新Duration。但是Retiggerable Delay可以在delay期间再触发时刷新Duration