#### 摄像机与小地图

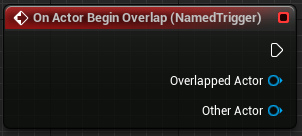
###### 切换摄像机

在一个位置创建**摄像机Actor**（或**电影摄像机Actor**），命名为NamedCameraActor，用于实现切换的目标摄像机。

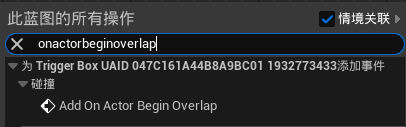
在一个位置创建**触发框**，命名为NamedTigger，用于实现事件触发，将触发框放到地面（End键）用于让角色Actor与其发生碰撞；在触发框的位置创建一个平面，用于让触发框能够被定位到。

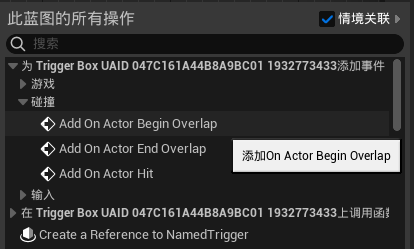
上述创建都是在关卡中进行的创建，不是创建的蓝图

接下来点击触发框Actor后进入关卡蓝图，右键创建与其相关的节点OnActorBeginOverlap节点



找到该节点的方式为如下两图



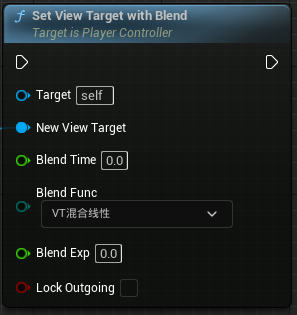


然后回到关卡创建界面，点击创建的摄像机后回到关卡蓝图，右键以创建对摄像机的引用。然后我们需要通过“**使用混合设置视图目标**”来启用上述创建的摄像机，即

**Set View Target with Blend**节点

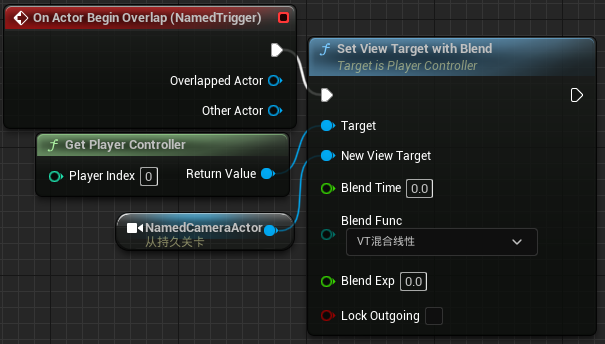
//设置与变量控制混合的查看目标

//目标是玩家控制器



既然目标是玩家控制器，Target自然是Get Player Controller的返回值

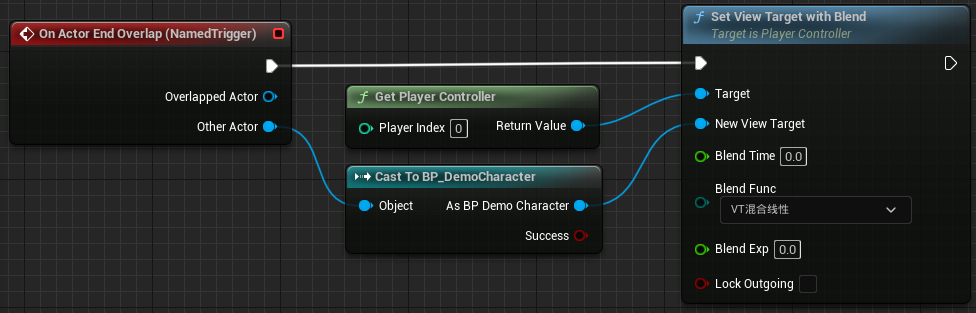
可以通过设置BlendTime让摄像机的切换过程变得丝滑



这样即可实现切换一个摄像机。但是如果我想切换回去，也就是当Overlap事件结束后切换回原摄像机，即OnActorEndOverlap节点。

要回到原摄像机，同样的是“**使用混合设置视图目标**”，将原摄像机的reference赋值给NewViewTarget参数即可

要获得角色Actor（蓝图为BP\_DemoCharactor）的原摄像机组件（在角色蓝图的组件中，名称为BackCamera），不用Get BackCamera，直接将角色赋值给NewViewTarget即可。因为Get到的BackCamera的数据类型是对象引用，而NewViewTarget参数的数据类型是Actor对象引用



###### 二、摄像机画面放大（拉伸镜头）：奔跑时适当拉伸镜头营造速度感

在角色Actor蓝图中，创建事件以使得摄像机显示的画面能够放大

创建键盘事件或鼠标事件的快捷方法：首先创建键盘1事件，然后在1事件节点的细节中找到“输入--输入键”，可以快速创建新的键鼠事件（经实验，手柄的摁键基本上都能得到相应）

找到摄像机组件，在细节中找到“摄像机设置--视场”。视场即可设置摄像机的视野大小。将摄像机组件拉入事件图表创建节点，从摄像机节点引出连接线进行“设置视野”，即

**Set Field Of View**节点

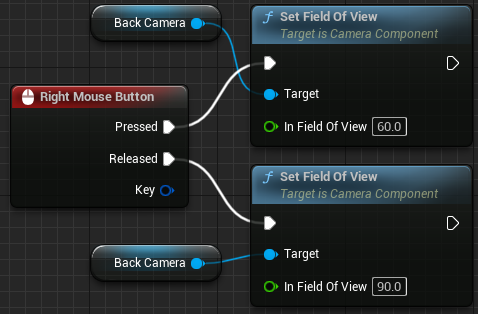
//目标是摄像机组件



视场和视野是一个东西，应该是翻译的问题

这里要区分Set Visibility的翻译。Set Visibility是设置可视性。

要想在时间结束时恢复视野，再设置一个Set Field Of View节点即可



###### 三、小地图

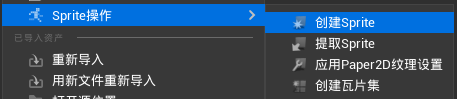
随意创建一个第三人称项目

用PhotoShop编辑一个箭头图像，PNG格式，此处命名为Arrow.png

在内容浏览器中内容文件夹下创建文件夹，命名为MiniMap

将该图片加入到内容浏览器中，放在MiniMap文件夹下，格式为纹理格式

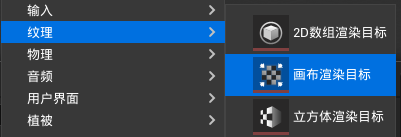
右键该纹理，选择



共产生两个资产



创建“纹理--画布渲染目标”



右键该资产，选择“创建材质”，共产生两个资产



进入材质进行编辑，对NewCanvasRenderTarget2D\_Mat节点进行编辑

在节点的细节中找到“材质--材质域”，将表面修改为用户界面，然后将两节点重新进行连接

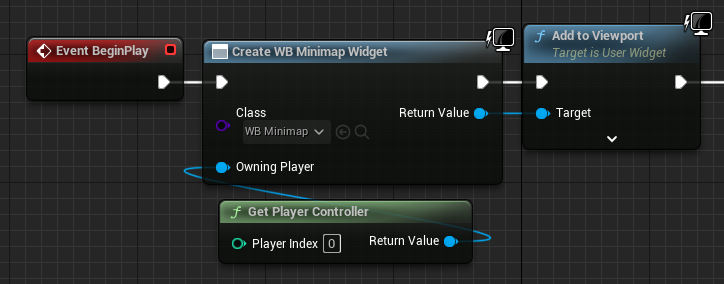
回到内容浏览器，在Minimap文件夹下创建控件蓝图，命名为WB\_Minimap，打开进行编辑，编辑成如下层级形式即可



在层级中有一个图像组件，编辑该图像组件的细节，找到“外观--笔刷--图像”，选择MiniMap\_Target创建出的材质

之后去编辑玩家角色的蓝图

首先一点是要在事件图表中添加WB Minimap控件并添加至viewport，事件绑定在EventBeginPlay下



进入视口界面，在胶囊体组件下创建**弹簧臂（Spring Arm）**组件

在弹簧臂组件下创建**场景捕获组件2D（Scene Capture Component 2D）**组件

之后通过旋转弹簧臂进而将场景捕获2D围绕角色进行旋转，使其摄像头朝向下方

在弹簧臂组件的细节中，通过“摄像机设置”，将所有继承全部关闭，可以使得小地图的摄像机方向不会随着角色进行转动

在场景捕获组件2D的细节中，找到“抛射--投射类型”，将透视换为正交，即保证了小地图摄像机的2D位置，消除了z轴自由度（可以试验不同的投射类型带来的效果）。自行调整正交宽度来修改小地图摄像机能够看到的宽度

在“场景捕获--纹理目标”中，选择Minimap\_Target，资产类型为画布渲染目标

此时小地图就能正常工作了，但我们希望小地图上显示的是Arrow，而不是角色的头顶。

此时我们在角色蓝图中创建Paper Sprite组件，作为网格体的子组件（实测作为其他的子组件也可以），在该组件的细节中进行编辑，找到“精灵--源Sprite（Sprite--Source Sprite）”选择我们的Arrow的Sprite资产。之后在视口中调整箭头的大小、位置，放在角色头顶

但此时进入游戏我们能看到头顶的箭头。我们需要移除它。在箭头组件的细节中找到“渲染--高级--拥有者不可见”，打勾

完成实现