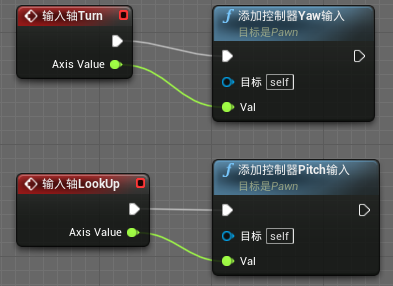
# 一、实现初始角色的选择

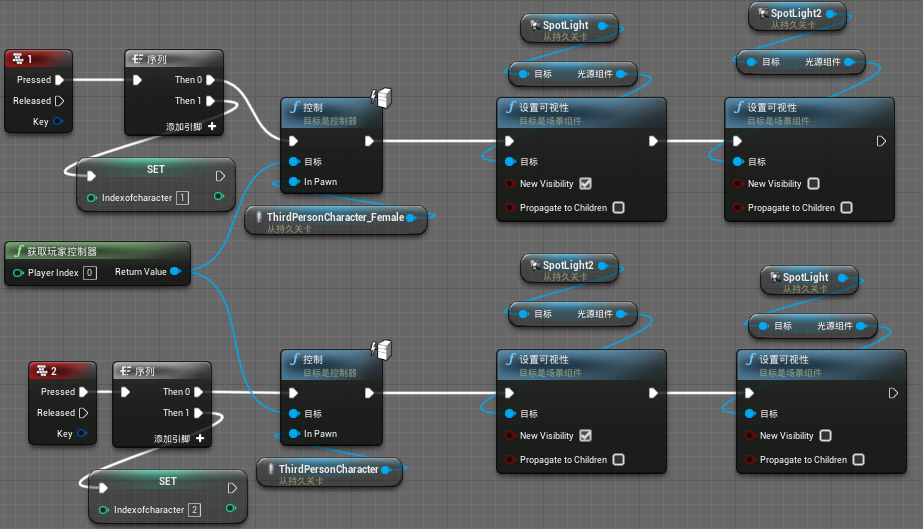
## 1.设计 choose\_characters蓝图类

作为初始的玩家控制对象，带有“摄像机组件”，其其主要功能为控制视角旋转，观察提供的角色模型。

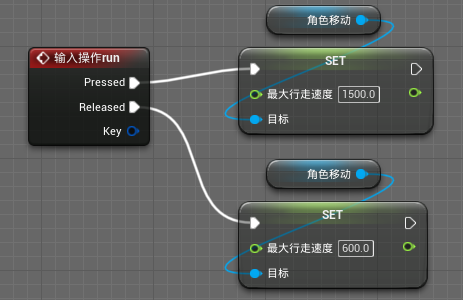


## 2.关卡蓝图设计角色切换

通过数字键“1”、“2”实现男女角色的选取，在获取对应角色模型后，点亮模型上方的光源。



此外，为角色设计加速跑功能，通过“set max walk speed”节点实现按左shift加速跑。先通过“编辑”—“项目设置”—“输入”创建加速跑输入操作。



# 二、电梯设计

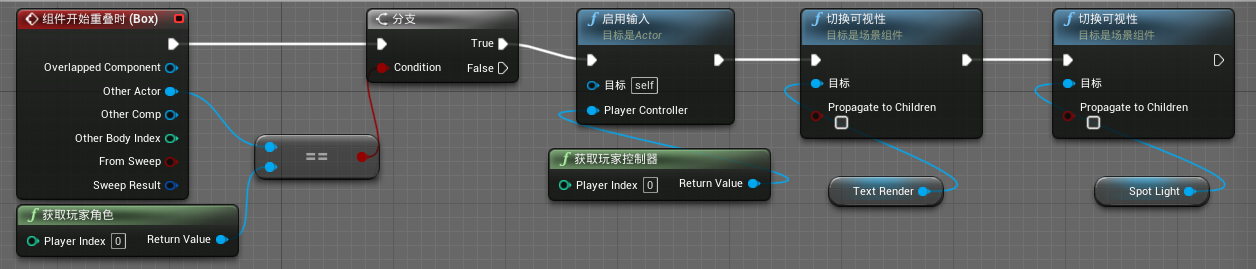
电梯的作用：选角色是在单独的空间内，选完角色后，通过电梯上升到上层空间，进行炮台操作。

（1）电梯包括静态网格体组件、碰撞盒、聚光源组件以及文本渲染组件。

（2）电梯功能为：

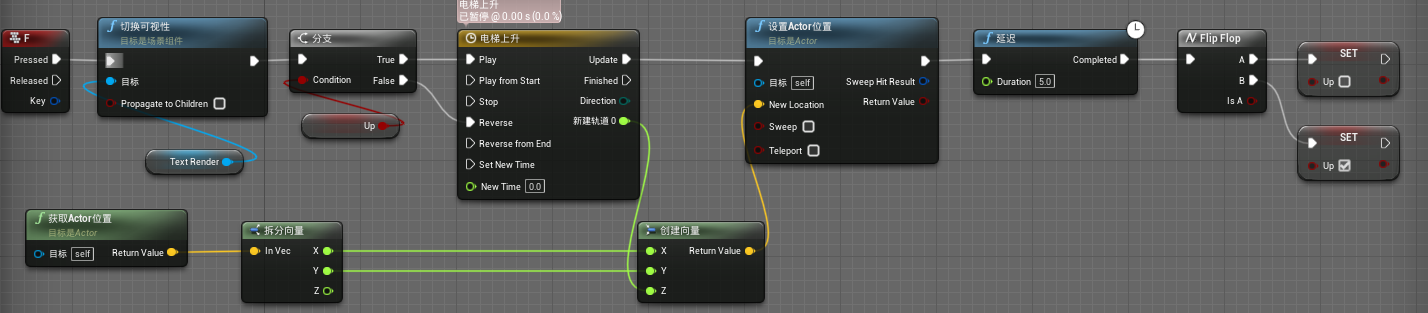
进入电梯位置后，电梯亮灯，同时显示提示文本“press F”。按“F”键后，电梯上升，提示文本消失。玩家离开电梯后，灯灭。

（a）碰撞盒检测是否与当前玩家角色产生碰撞，若是，则启用输入，显示提示文本，同时亮灯；



（b）按“F”键后，电梯上升，其通过Timeline辅助实现；

引入了布尔变量“up”，用于表征电梯该上升还是下降。在电梯到达指定位置后，通过“Flip Floop”节点切换“up”的值，实现上、下标定。



（c）玩家离开电梯后，关灯、关提示文本，并禁用输入。



# 三、门的设计

## 1. 总体思路

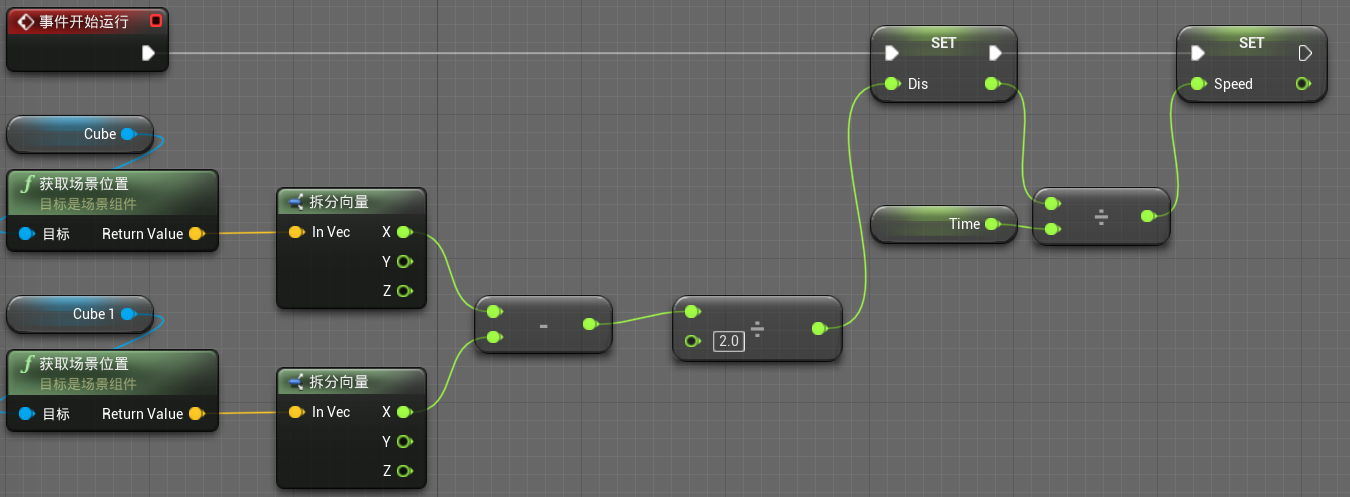
自动开、闭门由两个“cube”及一个碰撞盒组成。通过输入的开门时间与两个“cube”的初始化位置，计算得到门开的速度。玩家与碰撞盒产生碰撞后，两个“cube”分别向开门的方向移动；碰撞结束后，门开的速度取反，实现门的关闭。

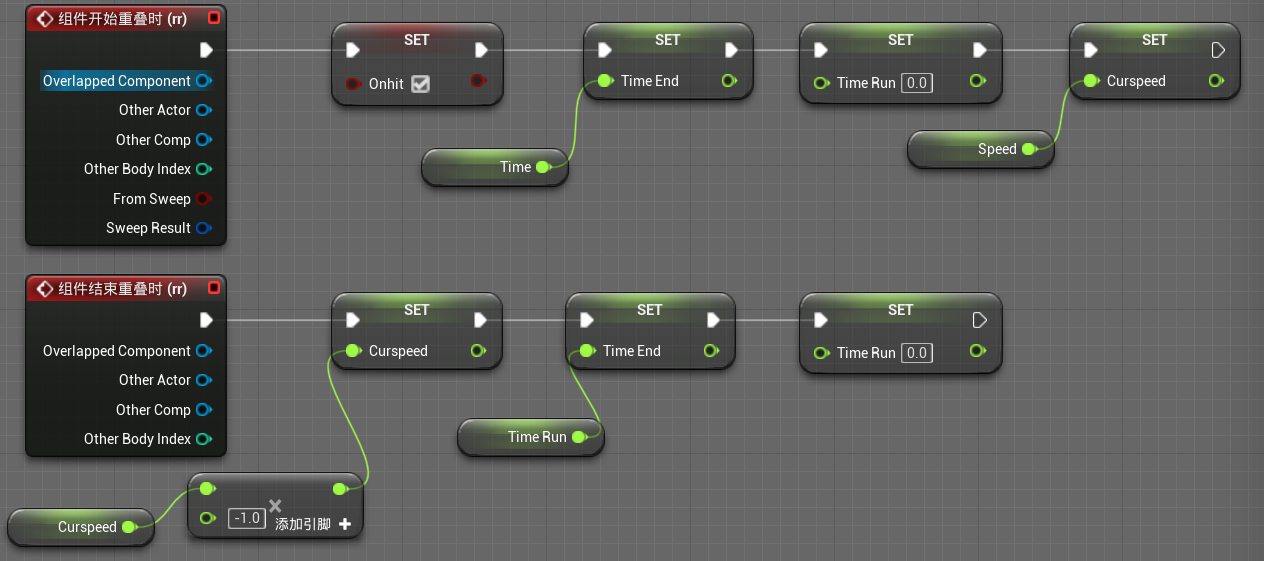
## 2. 开关门不用“Timeline”的原因：

Timeline需预设变化曲线，但门这个类可能有多个实体，每个实体对应不同的位置，需分别设计Timeline。因此，用一定的速度移动门实现开关门的功能。

## 3. 数据初始化

门类中的数据成员有：time、dis、speed、time\_run、time\_end、curspeed、onhit。各变量初始化方法如下图所示。





各成员含义为：

（1）time：用户输入，指定用多长的时间将门开到最大；

（2）dis：两个cube，即两门中心点间的距离，将其作为门开的最大尺寸；

（3）speed：门移动的速度，通过dis/time求得；

以上三个变量游戏一开始便完成初始化。

（4）time\_run：记录门当前运动了的时间，不能大于time，初始值为0；

（5）time\_end：记录门共运行了多长时间，其初始值为time；

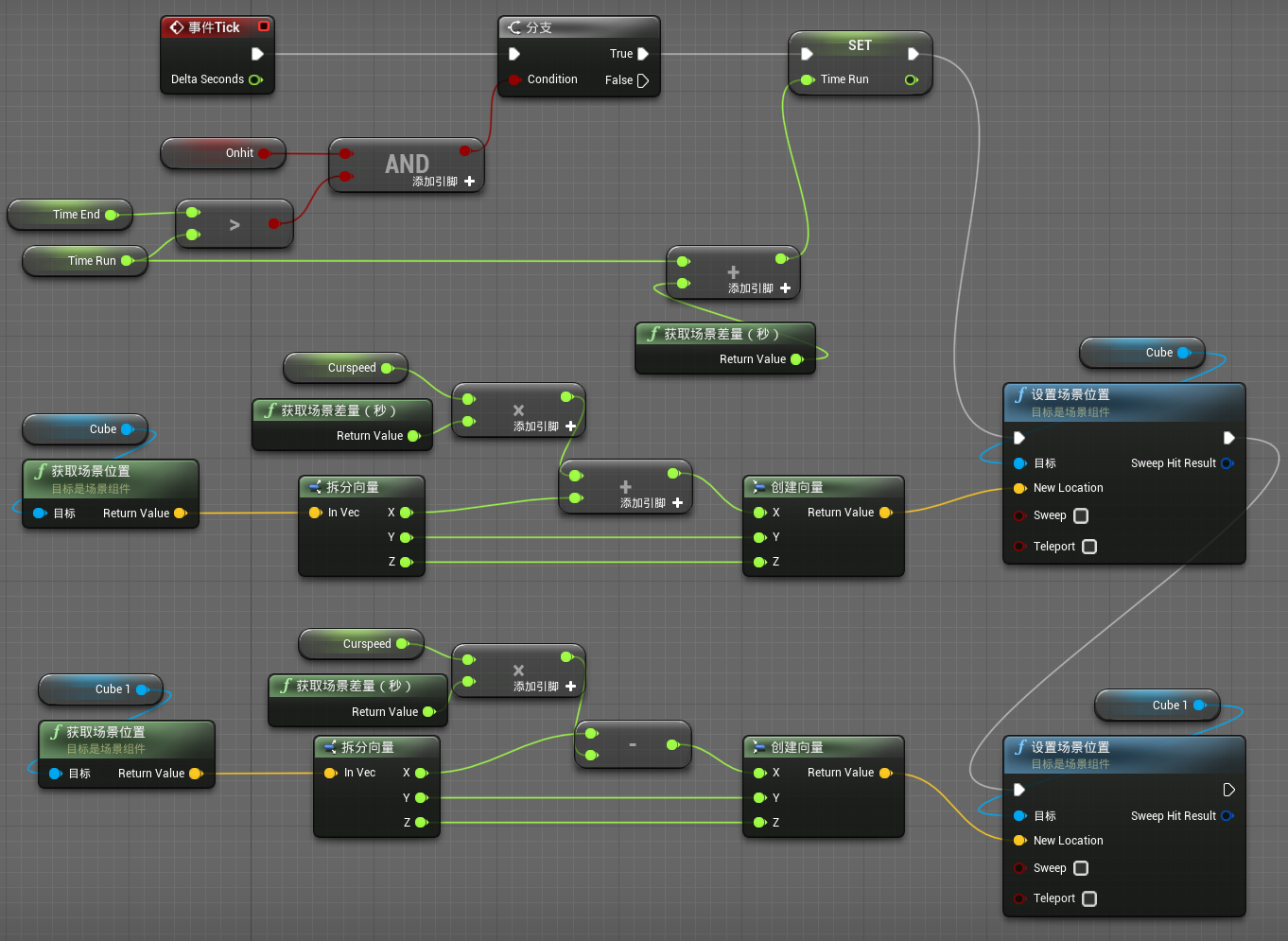
（6）curspeed：初始值为speed，通过curspeed的正负实现门向不同方向的运动；为避免修改speed导致后面再次开门时，门运动的方向发生错误，用curspeed代替speed；

（7）onhit：记录角色是否靠近了门，靠近了再执行开门操作。

通过后四个变量实现开、关门操作的切换，因此在发生重叠与结束重叠时对其进行修改。

## 4. 位置更新

每帧进行一次位置更新：获取各组件位置，通过curspeed对其x坐标进行修改（因为不同计算机每帧时间间隔不同，因此通过“获取场景差量（秒）”节点获取每帧的时间间隔。

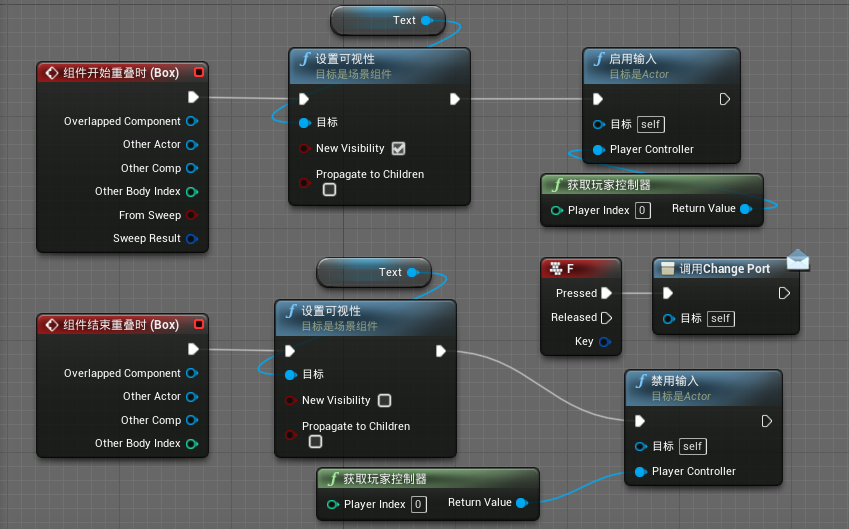


# 四、炮台相关的设计

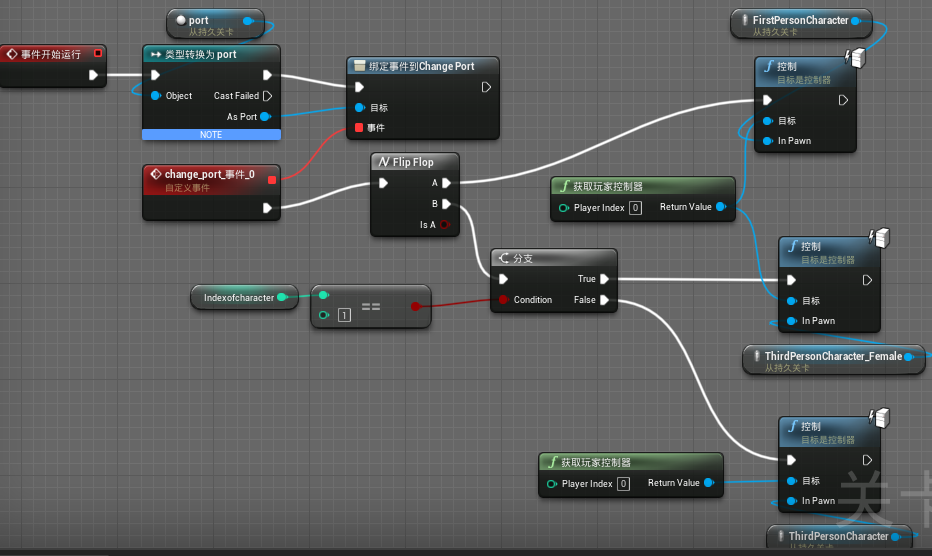
炮台直接使用“第一人称射击游戏”中的角色蓝图。在炮台附近放置了“port”蓝图类。该类由静态网格体、碰撞盒以及文本渲染器组成。其中，碰撞盒用于检测角色是否位于炮台附近，文本渲染器提示用户点击F键控制炮台。

## 1.角色切换

通过事件分发器完成“port”蓝图类与关卡蓝图之间的信息交互，实现：用户在炮台附近点击F键，玩家控制的角色切换为炮台；再次点击F后，控制角色切为原先角色。“port”蓝图的设计为：



关卡蓝图中，对应角色切换部分如下图所示。通过一个整型变量记录控制炮台前，玩家所控制的角色是男性还是女性。



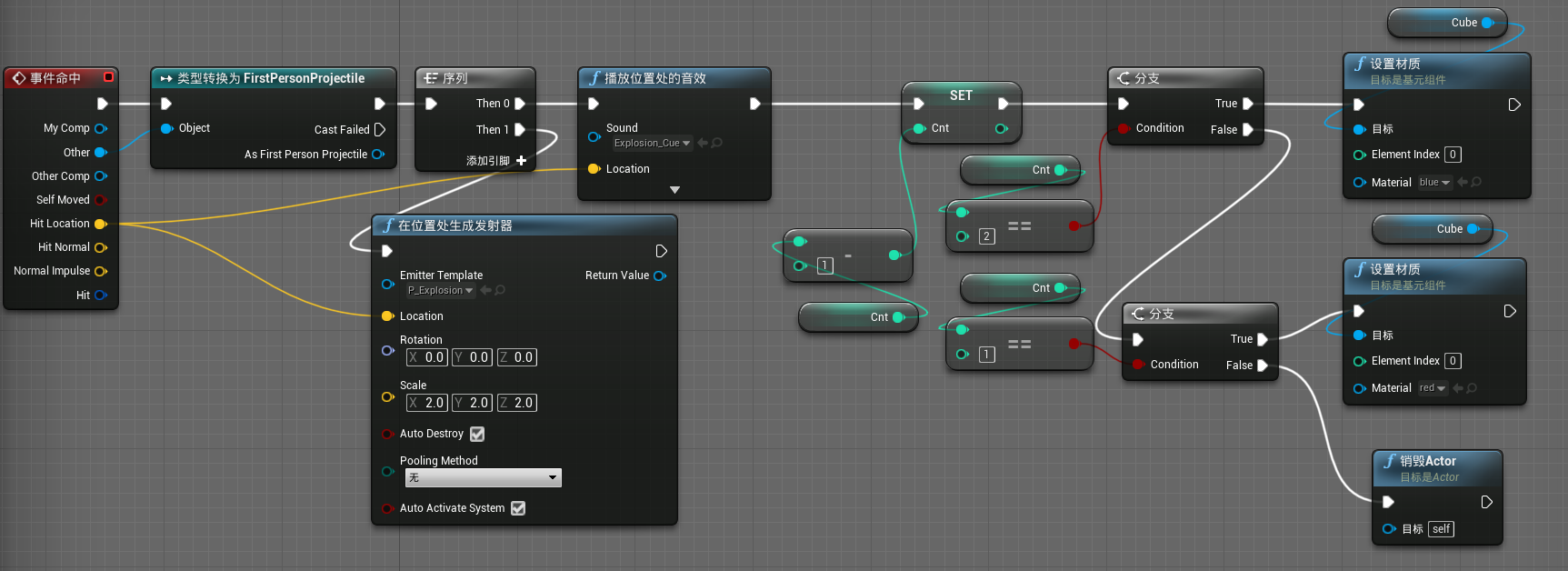
## 2. 瞄准

为炮台设计开镜操作，放大视野，其实现流程如下图所示。关键节点为“set filed of view”，即“设置视野”，更改摄像机组件的视野范围，实现开镜操作。其中“Timeline”用于设置视野变化的曲线。

# 五、NPC设计

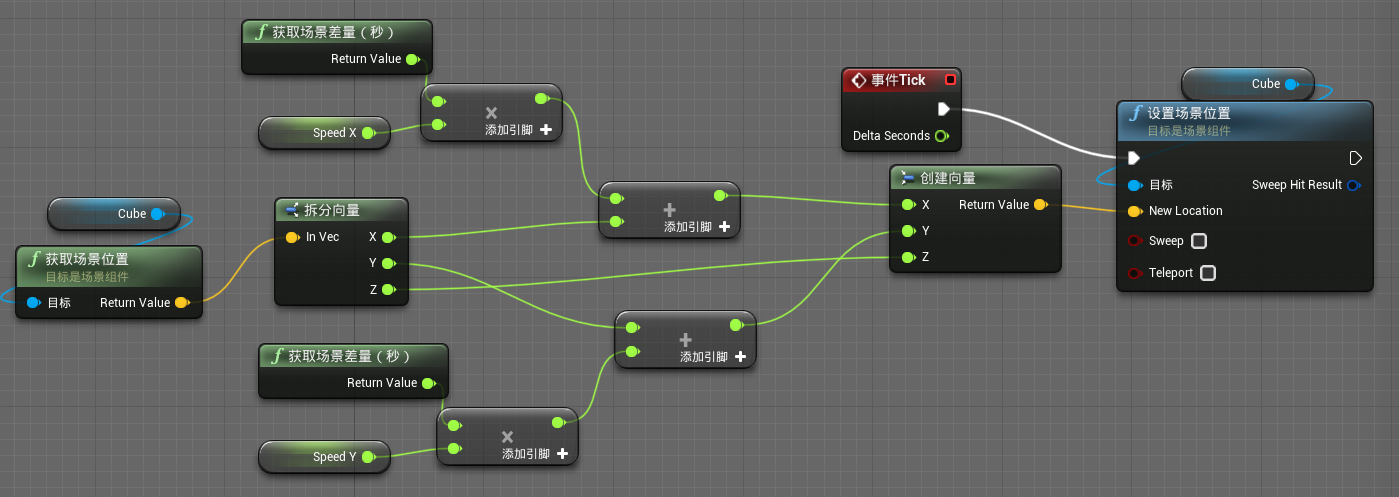
## 1. NPC变色与销毁

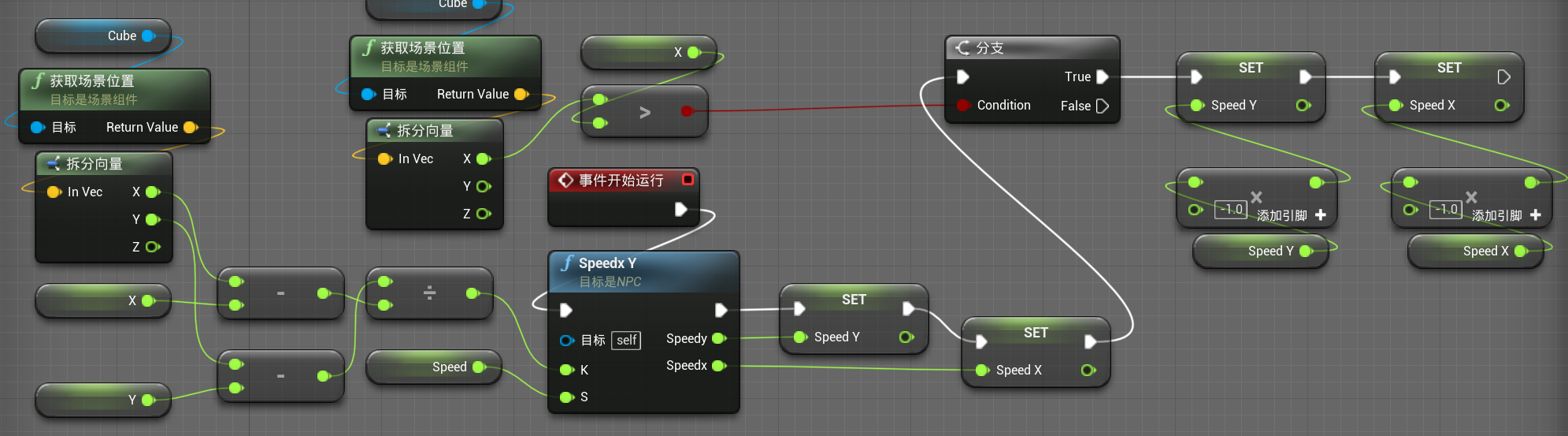
NPC用静态网格体代替，设计了两种颜色的材料，子弹击中NPC后会变色，同时在被击中出设置爆炸声音，产生粒子效果。击中三次后，NPC销毁。



## 2. NPC移动

NPC朝着炮台方向移动，提供speed变量用于设置NPC移动速度。此外，根据NPC生成位置与炮台位置，计算y、x向坐标的比值，以此为依据计算NPC在x、y方向的速度分量。并以生成位置是在炮台的左侧还是右侧为依据，对各方向的速度分量进行正负调整，实现NPC始终向靠近炮台的方向移动。



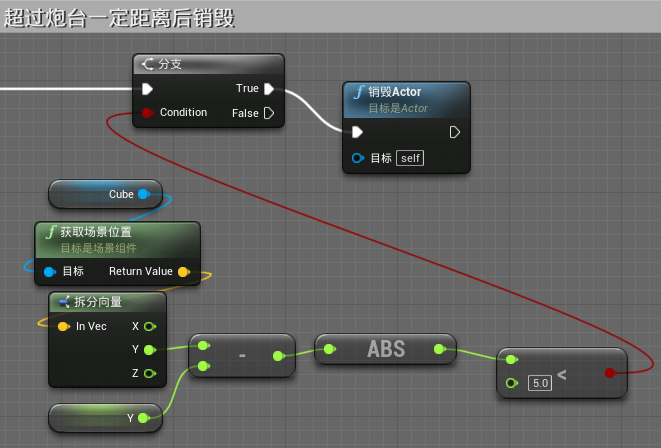


## 3. NPC销毁

NPC在以下两种情况下会进行销毁：

（1）子弹击中NPC三次后，NPC销毁；

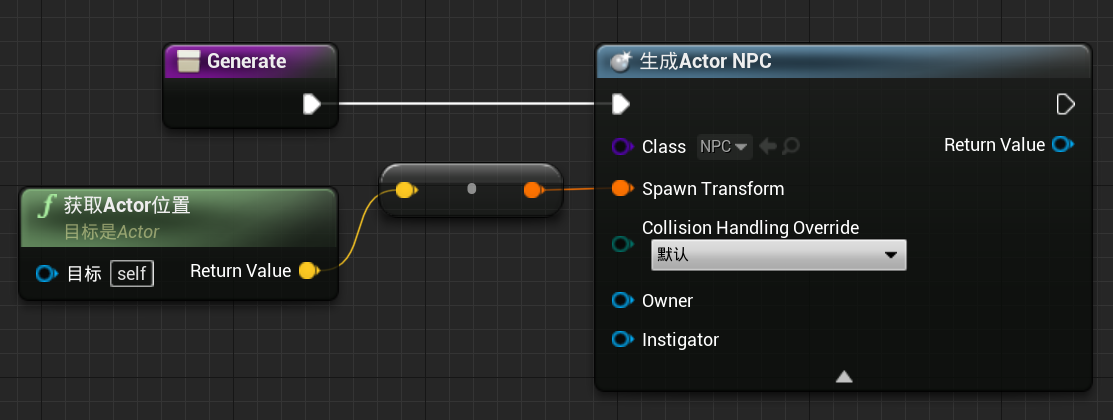
（2）NPC靠近炮台一定距离后会销毁。（触发事件为Event tick）



# 六、NPC生成

## 1. 生成器（generate\_NPC）

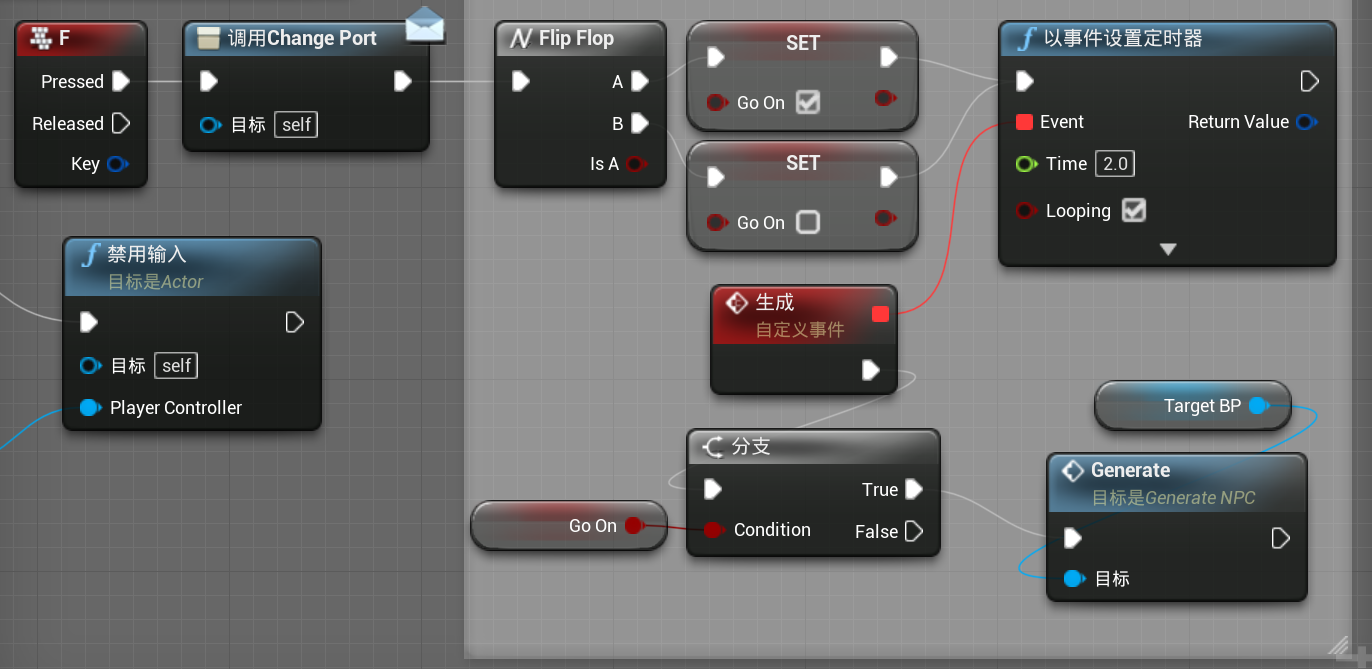
设计NPC生成器，主要调用“生成Actor NPC”节点，完成NPC的生成。为生成器类（generate\_NPC）设计函数“generate”，用于调用“生成Actor NPC”节点。



## 2. NPC间隔两秒生成实现

设计目的为：玩家切换到炮台后开始生成NPC，每隔两秒生成一个NPC。因此，在“”蓝图类中进行控制。通过直接通信方式实现两个蓝图类之间的信息交互。

“Port”蓝图类中NPC生成相关内容如下图所示。在玩家通过“F”按键切换到炮台的同时，通过定时器访问“generate\_NPC”蓝图类中的NPC生成函数，离开炮台后，停止生成。



## 3. NPC随机生成

利用随机流（RandomStream）生成一定范围内的随机数，在NPC生成器位置的x、y方向加上该随机数，实现NPC的随机生成。

将1中的生成函数蓝图修改为：

