

## 第 6 课 蓝图节点(5)

### 5、变量

#### (5) 数组 (Array) 变量

打开 VariableWorkShop 蓝图, 可以先删除事件图表页面下的所有蓝图节点, 并确定该蓝图类对象已经放置在场景中。添加一个“字符串 (string)”类型的变量, 命名为 MyArray, 把“变量类型”右侧更改为九宫格图标, 表示该变量是“字符串数组”类型。编译后“变量类型”和“默认值”都会发生变化。



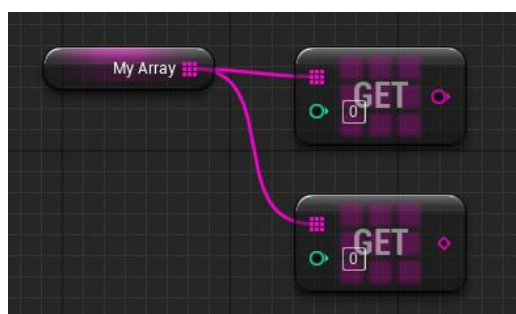
在默认值中添加 3 个字符串类型元素“Alice”、“Bob”和“Tom”。



按住 Ctrl 键把 MyArray 变量拖动到事件图表页面（相当于拖动后选择“获取 MyArray”），可以在“工具>>数组”和“工具集>>数组”里看到与该数组相关的节点。

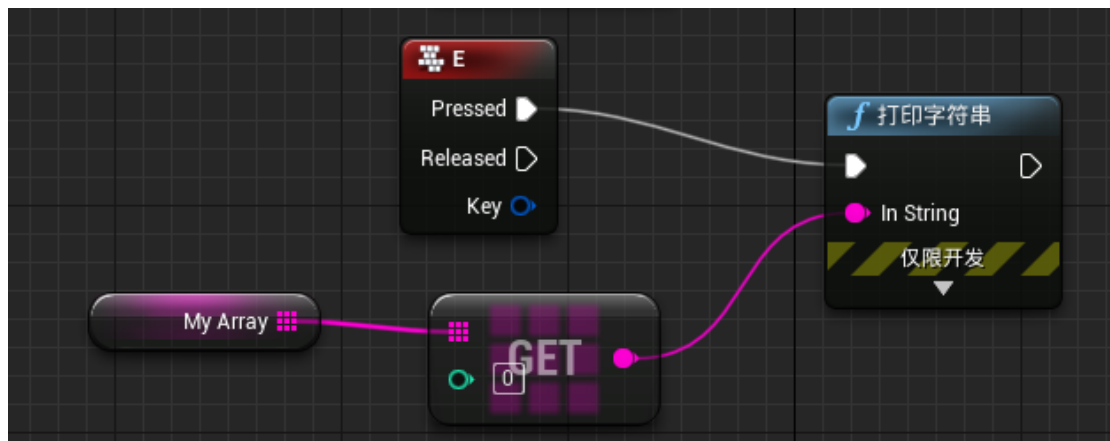


选择其中的“Get（复制）”节点和“Get（引用）”节点，可以看出两者的共同点和区别。



相同点都是根据数组和下标（浅绿色引脚，默认为 0）访问数组中的某个元素。区别在于靠上侧的“Get（复制）”节点返回数组元素的副本，靠下侧的“Get（引用）”节点返回数组元素本身，任何后续的修改都会改变数组元素的值。

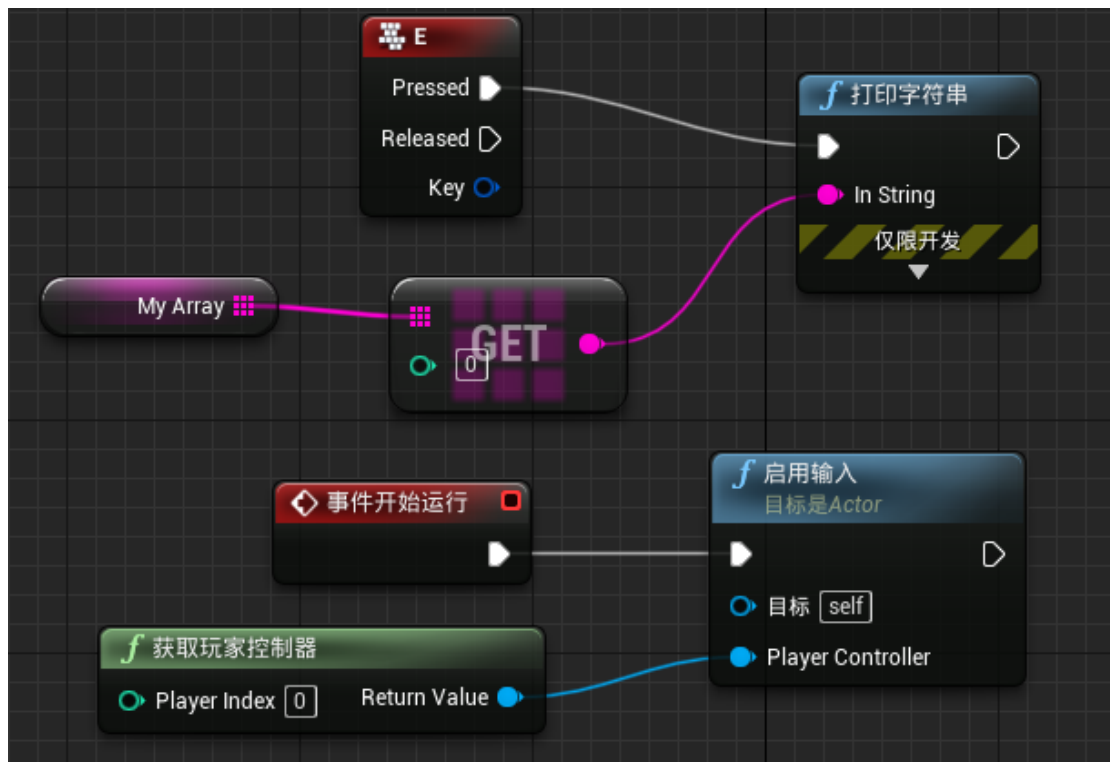
保留其中的“Get（复制）”节点，添加功能：按下 E 键后，屏幕上显示 Alice。



测试游戏，发现按 E 键后屏幕上并没有显示文字，原因在于关卡蓝图的细节面板中，“自动接收输入”一栏默认禁用所有玩家输入，可以设置为当前玩家（玩家 0），就可以在游戏开始时自动注册该玩家的输入。



也可以在 VariableWorkShop 蓝图中添加一段注册玩家输入的节点。



游戏运行时，当按下 E 键后，屏幕上会显示 Alice。（作业第 1、2 题）



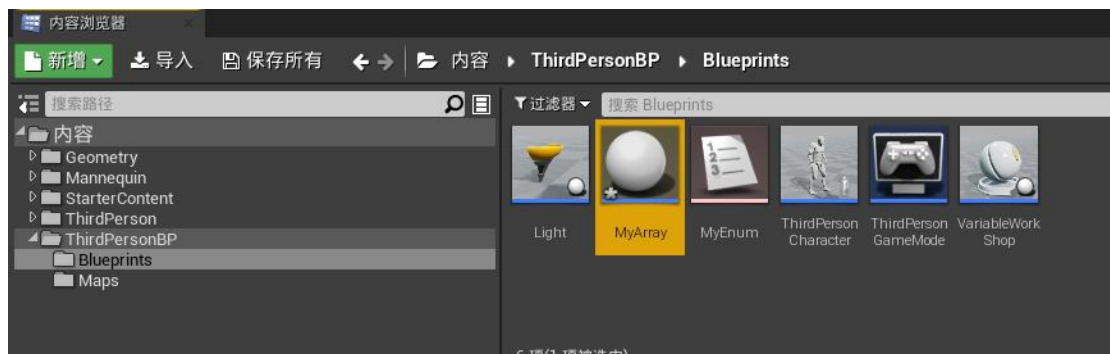
接下来要用数组实现一种让某一类对象随机变换摆放形式的机制。在游戏场景中创建多个 StarterContent>>Props>>SM\_MatPreviewMesh\_02 静态模型实例对象。



把所有 SM\_MatPreviewMesh\_02 静态模型实例对象的细节面板中的移动性修改为“可移动”，默认值“静态”将导致对象在游戏中不可修改不可移动，“固定”指对象可修改但不可移动。对于光照渲染和烘焙而言，“静态”所耗费的计算代价最小，“可移动”的代价最大。



创建一个以 Actor 为父类的蓝图类，命名为 MyArray（注意：不是 VariableWorkShop 蓝图中的字符串数组 MyArray，只是重名而已）。



打开 MyArray 蓝图，在事件图表页面搜索“获取类的所有 Actor（Get all actor of class）”节点。

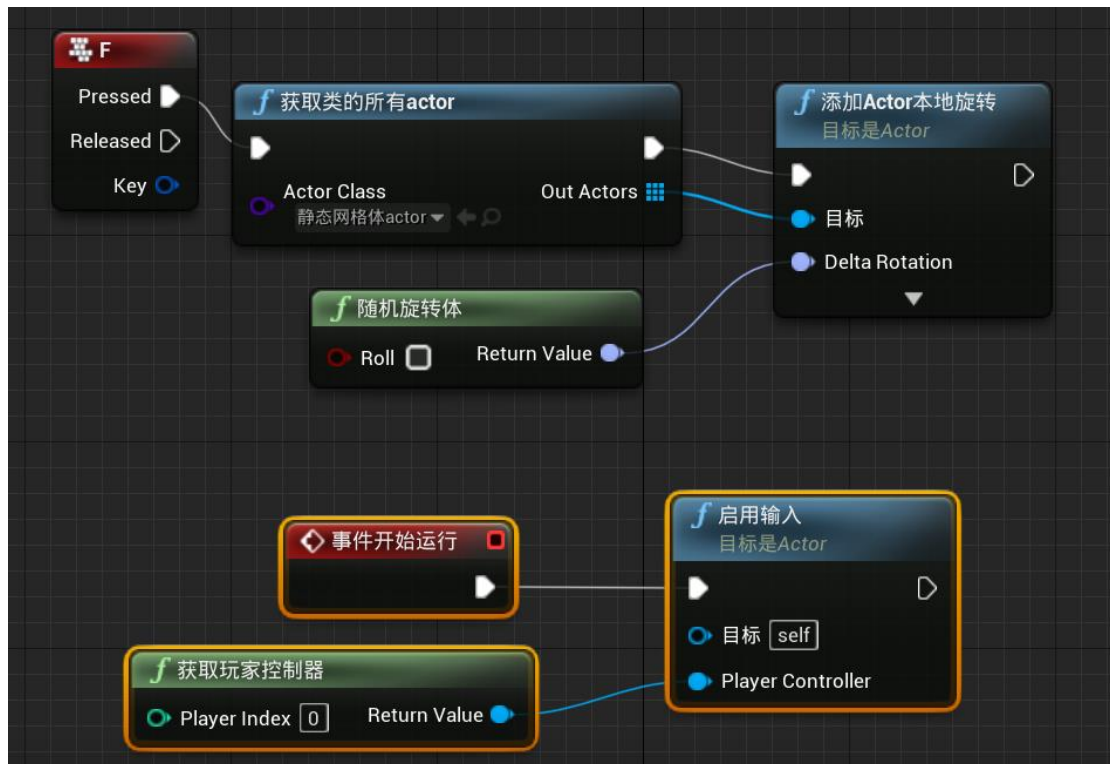


该节点通过设置 Actor Class，获取关卡场景中所有属于该类的对象，返回值是一个对象数组。由于之前在游戏场景里新增的 4 个 SM\_MatPreviewMesh\_02 静态模型实例对象属于 StaticMeshActor 类，故设置 Actor Class 为 StaticMeshActor（静态网格体 actor）。



添加“随机旋转体（Random Rotator）”节点、“添加 Actor 本地旋转（Add Actor Local Rotation）”节点、“键盘个事件>>F”节点，并添加启用玩家输入的节点模块。

把 MyArray 类拖动至游戏场景中。



游戏运行时，每当按下 F 键，4 个静态对象会随机旋转到某个角度。



游戏运行后会有一堆关于对象的移动性必须为“可移动”的警告，原因是不但 4 个 SM\_MatPreviewMesh\_02 静态模型实例对象属于 StaticMeshActor 类，游戏中的墙壁、地板、台阶都属于 StaticMeshActor 类，而这些对象都是不可移动的。



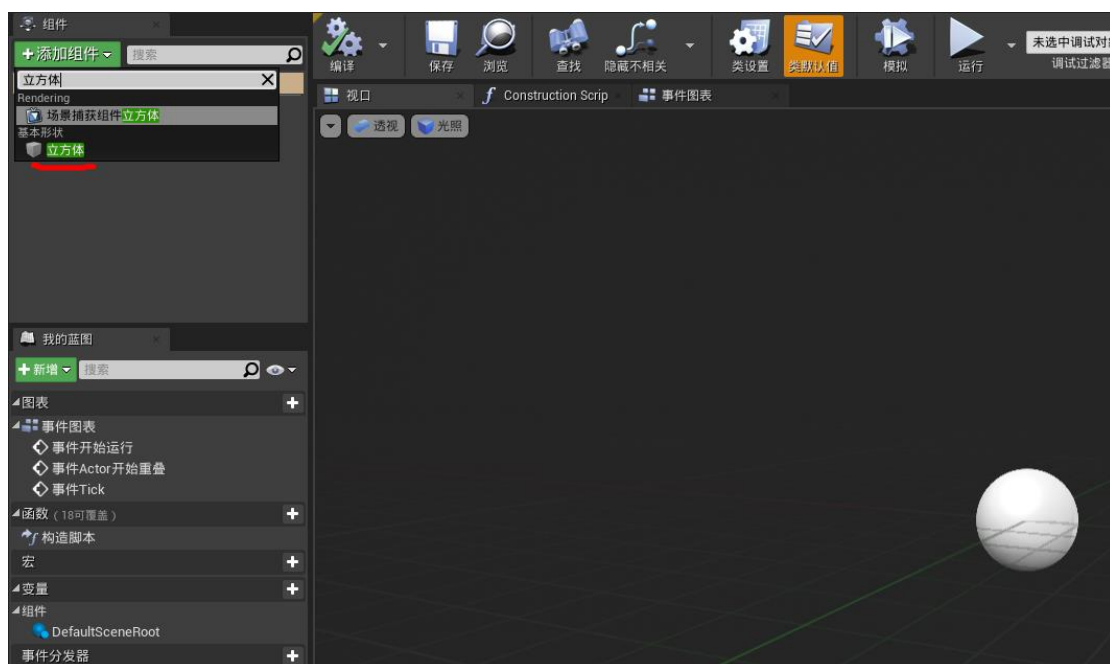
```
• 登陆的服务器
• PIE总开始时间: 0.111秒。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Linear_Stair_StaticMesh: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Bump_StaticMesh: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.LeftArm_StaticMesh: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.RightArm_StaticMesh: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Ramp_StaticMesh: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.CubeMesh_5: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Floor_1: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Wall7_4: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Wall19: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Wall10: StaticMeshComponent0的移动性必须为“可移动”。
! 如果需要移动, 则/Game/ThirdPersonBP/Maps/UEDP1E_0_ThirdPersonExampleMap.ThirdPersonExampleMap.PersistentLevel.Wall11: StaticMeshComponent0的移动性必须为“可移动”。
```

所以想要实现指向性更为明确的“获取类的所有 Actor (Get all actor of class)”节点效果，必须使用自定义类，以避免获取到其他无关对象。恰好 VariableWorkShop 蓝图中已经加入了 SM\_MatPreviewMesh\_02 作为组件，可以利用该蓝图类更完美（不报错）地实现随机旋转的功能（作业第 3 题）。

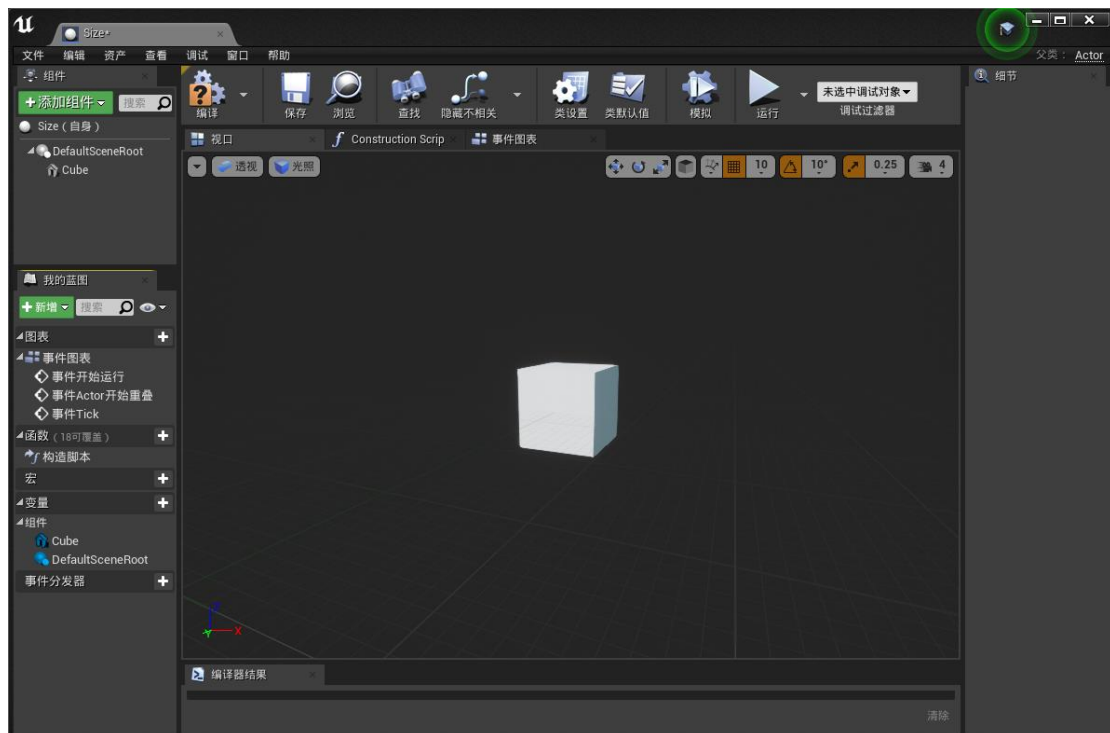
## (6) 显示 3D 控件 (Show 3D Widget) 属性

本节中将对象的属性值以 3D 控件的形式呈现在游戏场景中，使得变量在游戏场景中变得可视化。显示 3D 控件是向量 (Vector) 和变换 (Transform) 类型变量特有的属性，利用它可以在场景中自定义一些几何题的外形。以下将实现一个按键，当拖动按键时可以改变立方体的高度。

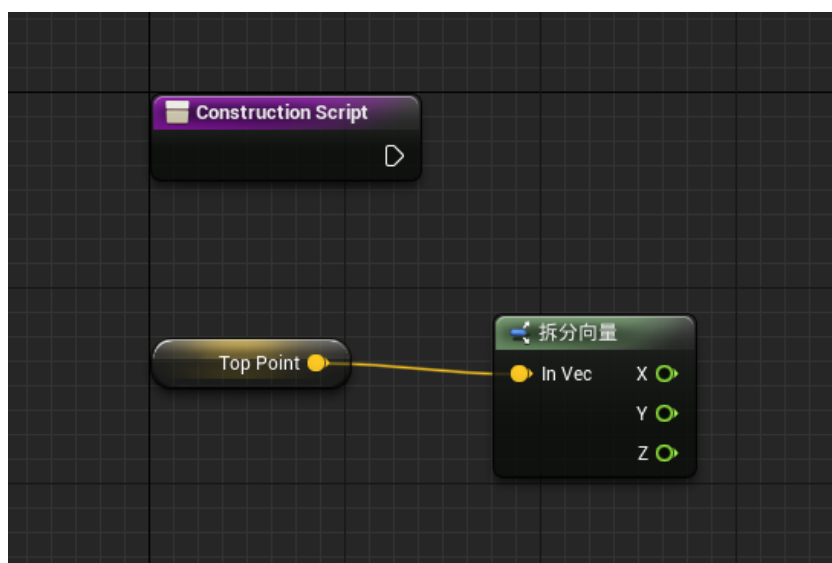
创建一个以 Actor 为父类的蓝图，命名为 Size。打开 Size 蓝图，在视口面板中添加一个“立方体”组件。



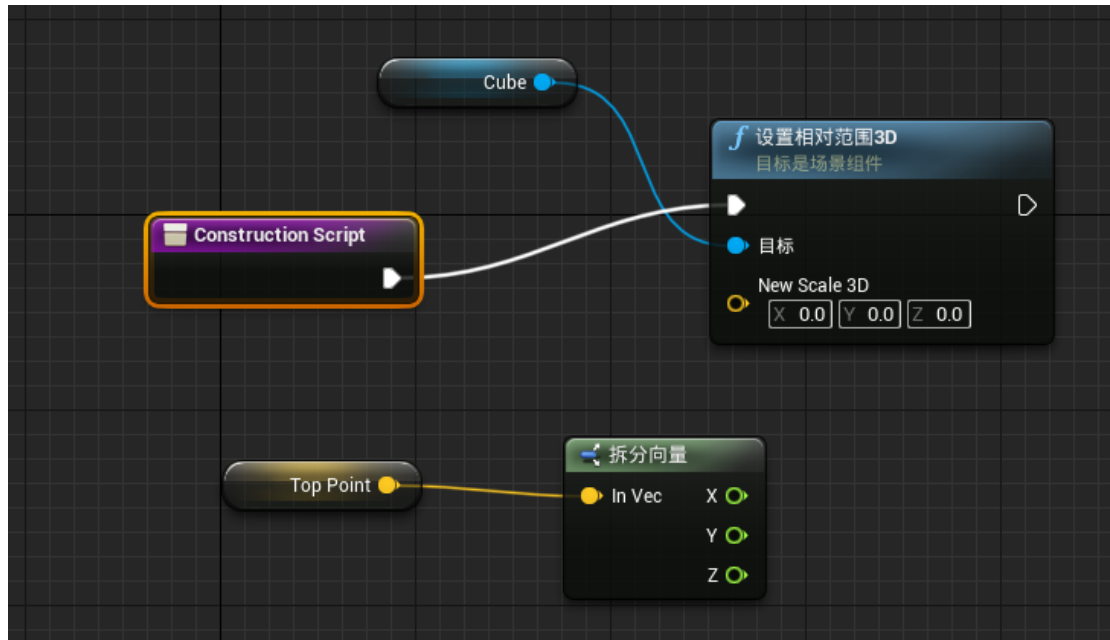




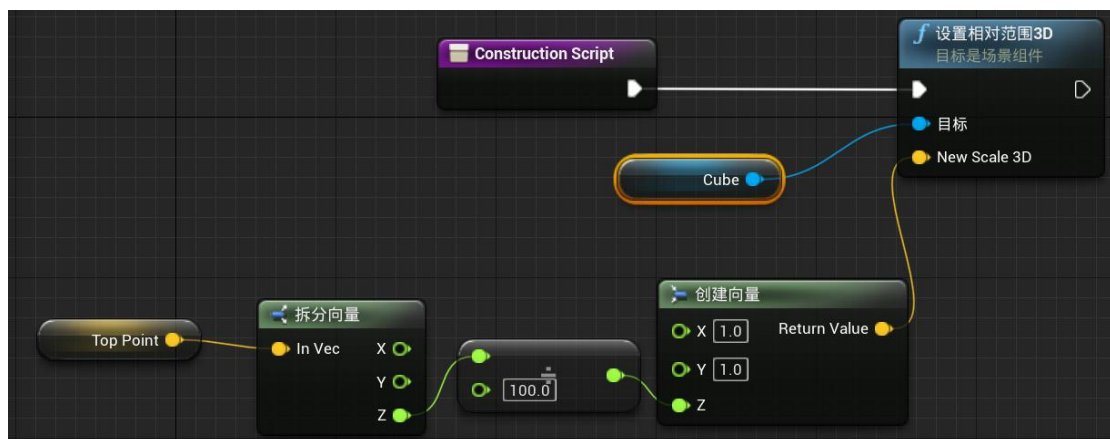
在 Size 蓝图内继续创建一个向量 (Vector) 类型变量，命名为 TopPoint，该变量将成为调整立方体高度的调节控件。切换到 Construction Script 面板，获取 TopPoint 变量，添加“拆分向量 (Break Vector)”节点，该节点可以分解出 TopPoint 的 X、Y、Z 分量。由于期望改变的是立方体的高度，也就是改变 Z 轴方向的数值，因此 TopPoint 会通过 Z 轴方向上的上下移动来改变立方体的高度，所以只需要改变 TopPoint 的 Z 轴坐标。



拖动“Construction Script”节点的执行引脚，搜索“设置相对范围 3D (Cube)” (Set Relative Scale 3D(Cube)) ”节点，该节点可改变指定对象的相对缩放值（相对于视口中的对象大小）。其中 New Scale 3D 参数用于指定赋予对象的缩放值。

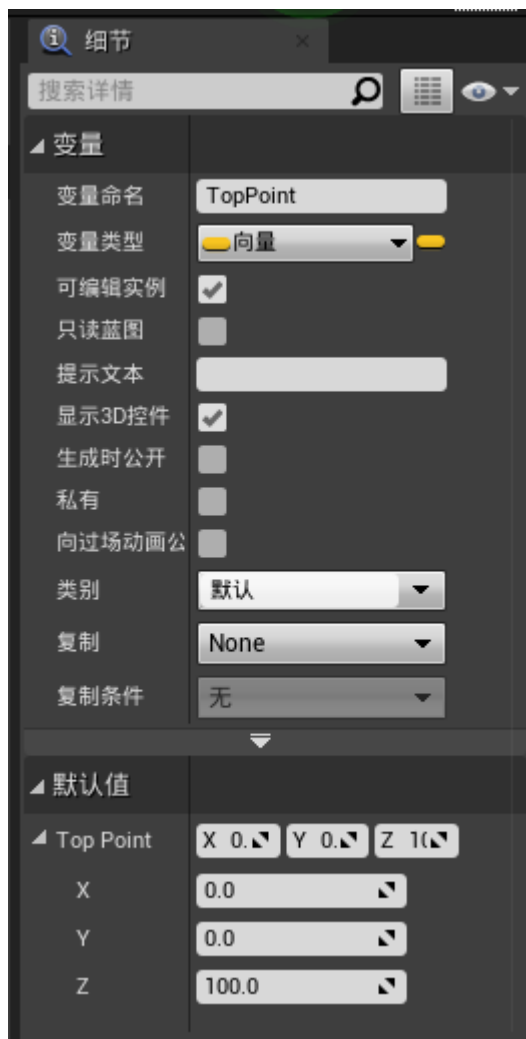


添加“创建向量 (Make Vector)”节点，该节点用于把 X、Y、Z 值组合成向量。其中 X 和 Y 的值保持不变，Z 的值由 TopPoint.Z 得到。当两者的比例是 1: 1 时，可以直接把 TopPoint.Z 作为创建向量中的 Z 值；可以根据情况缩小或扩大这个比例，例如当将 TopPoint.Z 缩小 100 倍后，再赋给立方体的 Z 轴方向的缩放值。

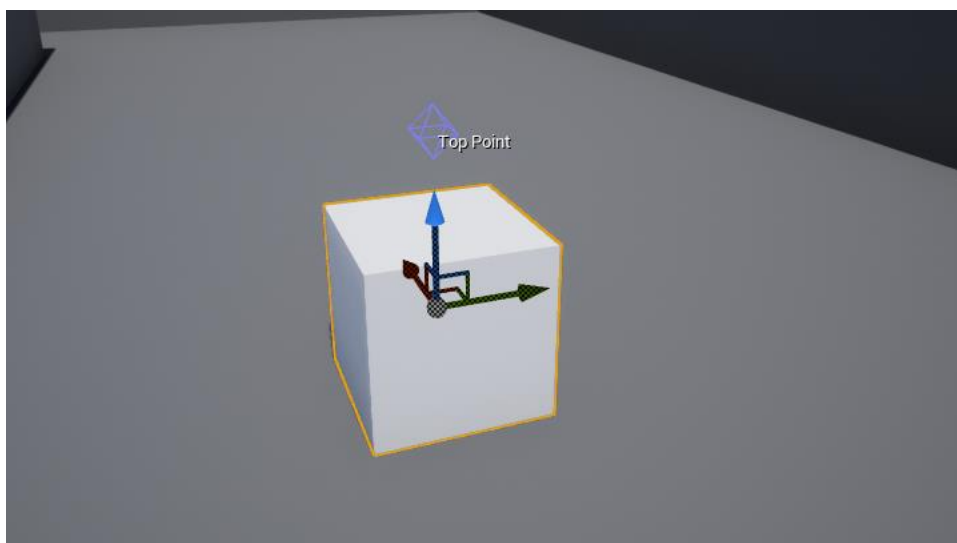


修改 TopPoint 变量的细节面板，勾选“可编辑实例 (Editable)”和“显示 3D 控件 (Show 3D Widget)”，把默认值中的 Z 值修改为 100。显示 3D 控件为 true

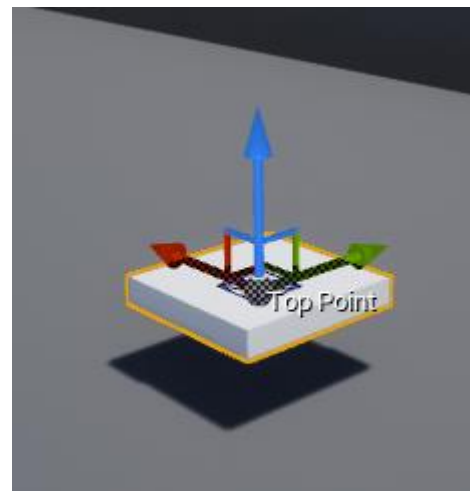
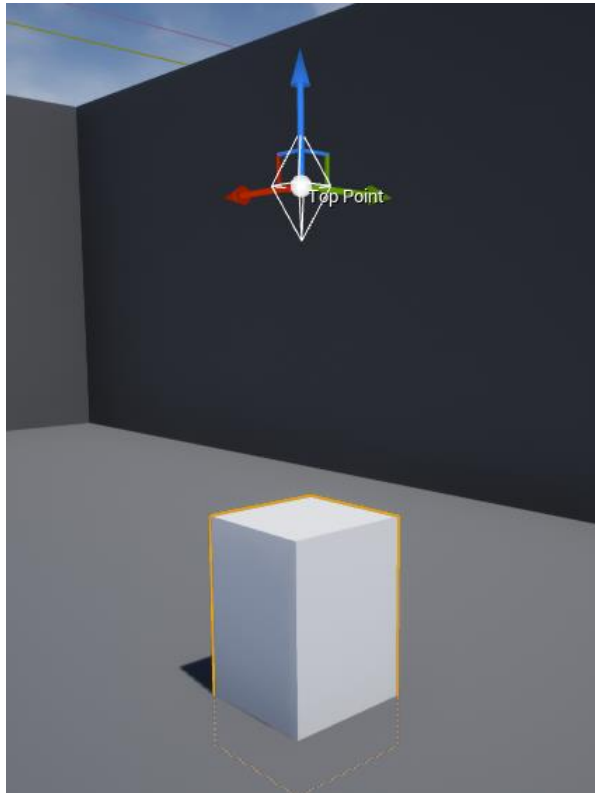
时，用户可以在游戏界面中调整该变量的值（可编辑实例必须勾选）。



把 Size 蓝图拖动到游戏场景中，可以看到 Size 蓝图对象除了一个立方体外，还有一个紫色控件，这就是以 3D 控件形式显示的 TopPoint 变量。(作业第 4 题)



在场景选中 TopPoint 控件，改变其 Z 轴的坐标值，立方体也会随着其位置的变化而线性缩放，改变高度。

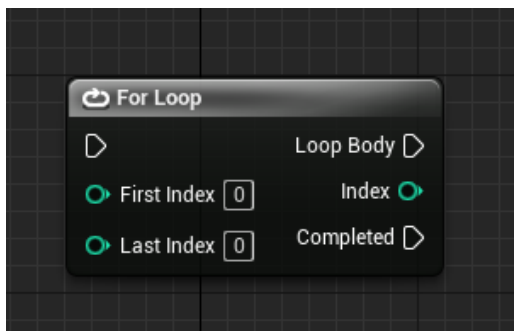


## 6、For 循环和 While 循环

本节介绍 4 种循环节点，分别是 ForLoop、ForLoopWithBreak、ForEachLoop 和 WhileLoop 节点。

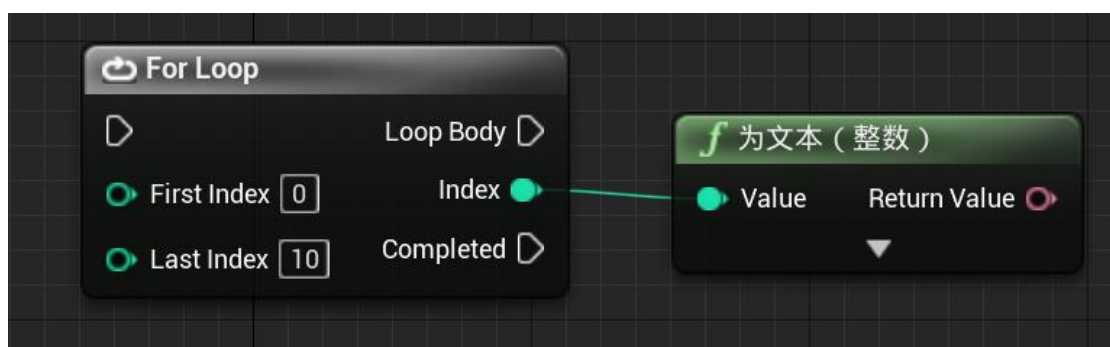
### (1) ForLoop 节点

创建一个以 Actor 为父类的蓝图，命名为 Loops。在事件图表面板中，添加 ForLoop 节点。



这是标准的循环语句，当前循环变量 Index 从 First Index 循环到 Last Index，每次循环在 Loop Body 输出执行脉冲。每次执行完毕 Loop Body 后，Index 的值加一。当 Index 大于 Last Index 时，循环结束。当循环结束后，Complete 执行引脚会触发执行。

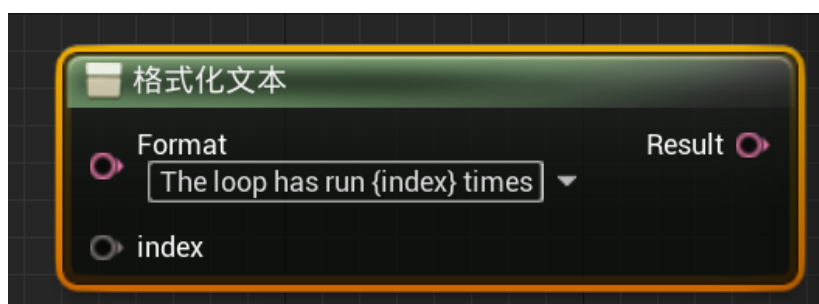
添加“为文本（整数）（To Text(int)）”节点，把循环变量 Index 的值转换为文本类型。



如果在打印 Index 的基础上，还想丰富打印内容，则添加“格式化文本 (Format Text)”节点，该节点有一个 Text 类型的参数 Format，用于指定文本格式，右侧 Text 类型的 Result 引脚返回格式化后的文本。



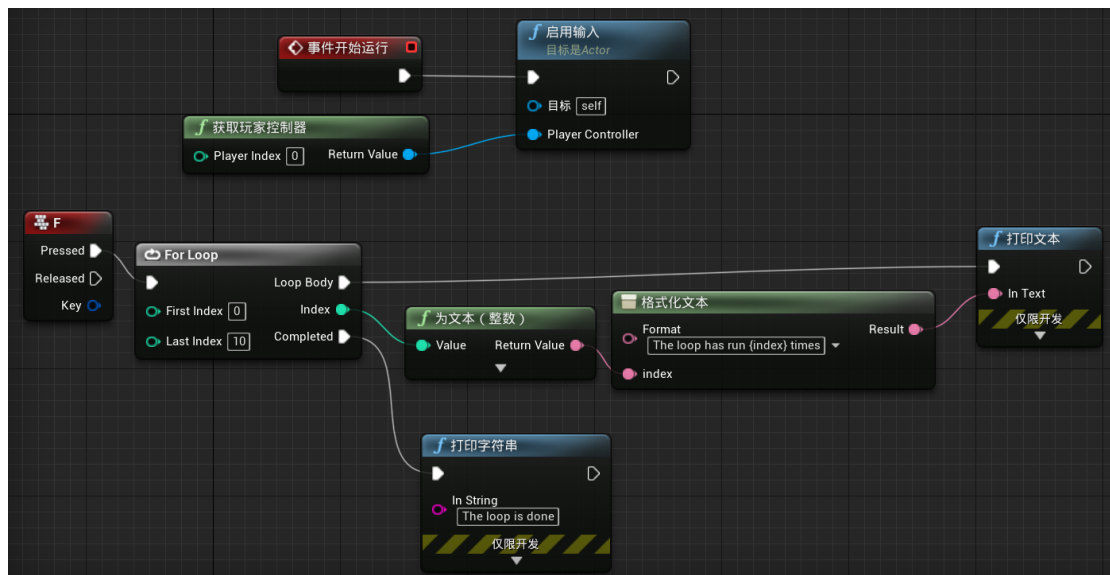
在 Format 右侧文本框中输入“The loop has run {index} times”，{...}中的文字会被作为参数自动添加到“格式化文本”左侧作为输入引脚。



连接蓝图，把循环变量的值作为 index 参数值。



补充完整蓝图，然后把 Loops 蓝图拖放到游戏场景中。



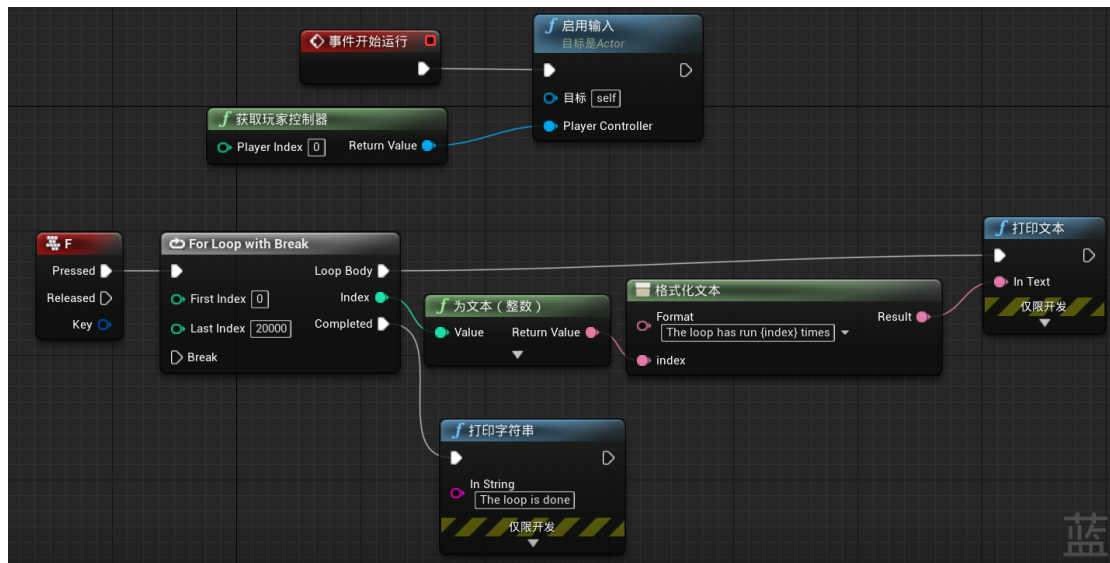
游戏运行时，当按下 F 后，屏幕上会显示相应的文字。



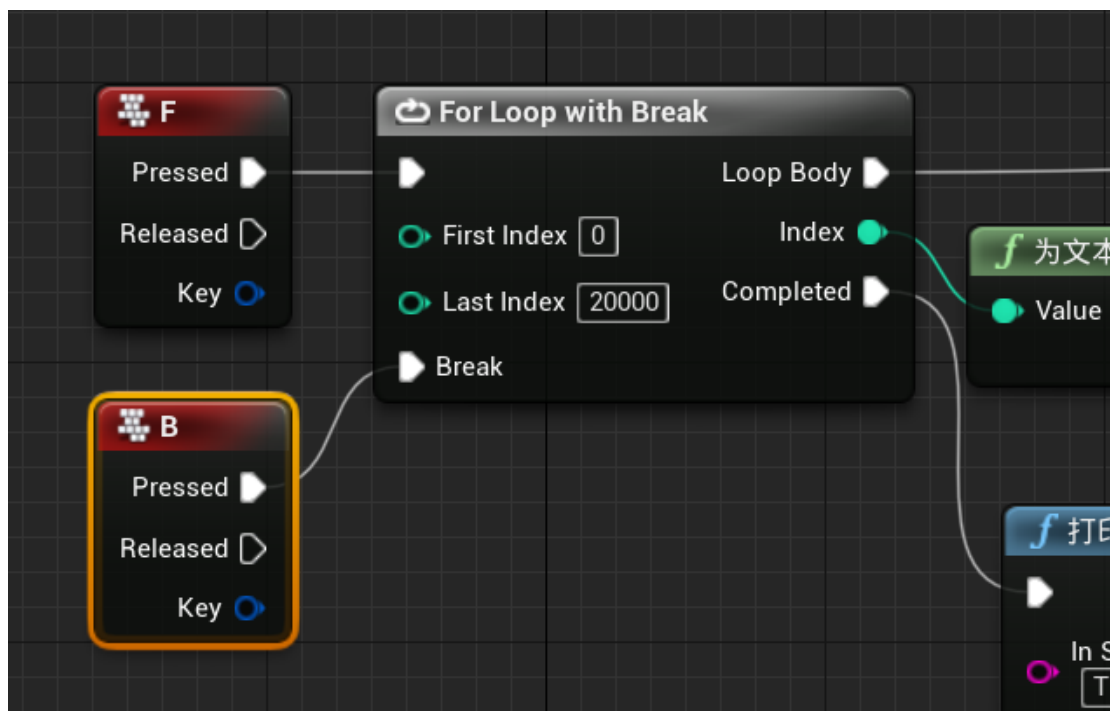
## (2) ForLoopWithBreak 节点

一个可以设置循环中断条件的循环节点。打开 Loops 蓝图，把 ForLoop 节点

删掉，改为 ForLoopWithBreak 节点。把节点的 Last Index 改为 20000。

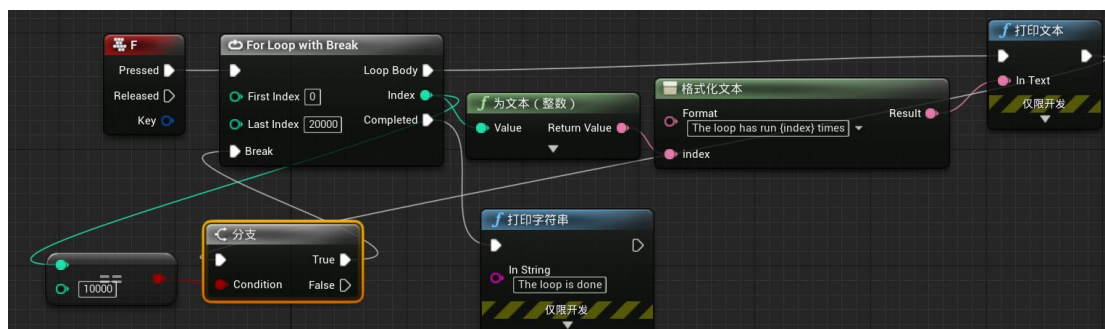


增加功能：在循环过程中按下 B 键，循环提前终止。但如果只是简单添加按键 B 和 Break 引脚相连，游戏运行时按下 B 键并不能中断循环。原因是当循环执行的过程中，游戏不会响应任何按键事件（而且循环执行得太快了）。



但是如果不是按键事件，可以在中断 ForLoopWithBreak 循环。例如用  $index == 100$  作为中断条件。

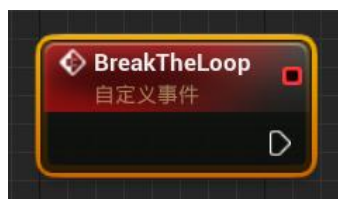




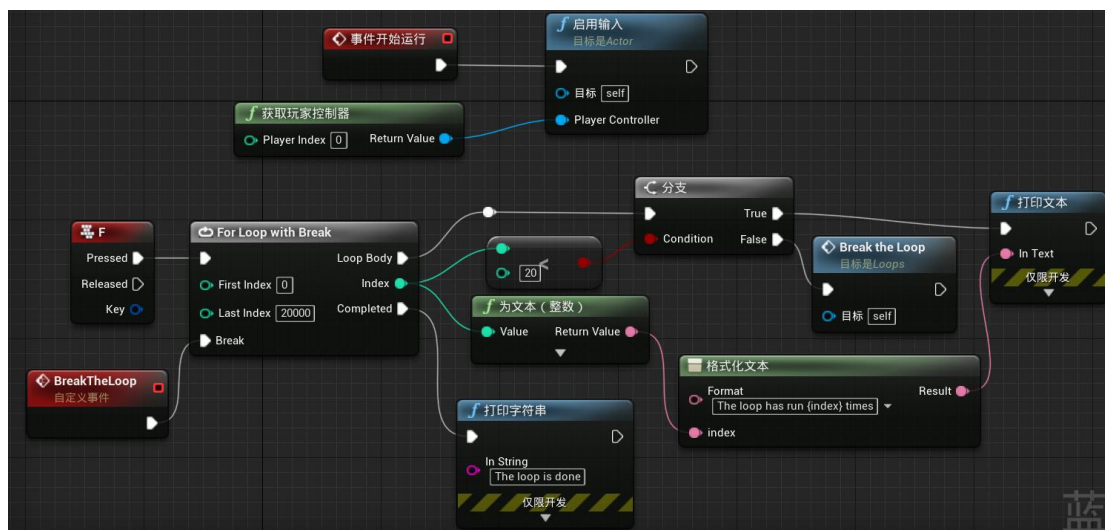
如果要在循环过程中响应按键事件中中断循环，那么需要用到“自定义事件 (Custom Event)”节点。



可以看到，“自定义事件”节点没有直接触发执行的执行引脚。把节点名字改为 BreakTheLoop。

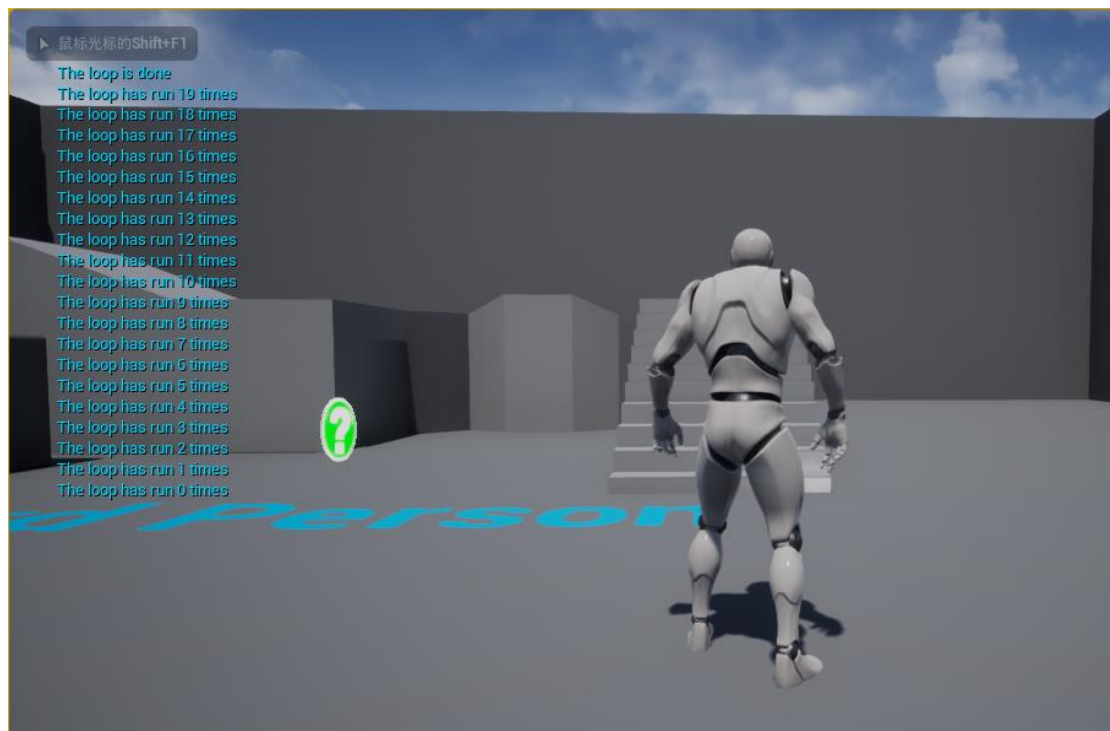


自定义事件可以在别处触发，则该节点会输出执行引脚。根据循环变量 index 的值，当  $index < 20$  时，正常输出文本，否则触发自定义事件 BreakTheLoop。



确保 Loops 蓝图已放置在游戏场景中，执行游戏，按下 F 键。屏幕上输出至

“The loop has run 19 times”后，随即输出“The loop is done”。说明循环提前被中止。



作业：1、利用课件在 VariableWorkShop 蓝图中已经定义好的字符串数组 MyArray，修改蓝图，按 E 的时候输出第一项 Alice，再按 E 的时候输出第二项 Bob，第三下按 E 时输出第三项 Tom，然后第四项又再次输出 Alice，以此类推。

2、利用课件在 VariableWorkShop 蓝图中已经定义好的字符串数组 MyArray，修改蓝图，按 E 时输出数组里所有的项。

3、课件里把 4 个 SM\_MatPreviewMesh\_02 静态模型实例对象放置在场景中，并用蓝图控制对象的旋转。现在改成把 4 个 VariableWorkShop 蓝图对象放置场景中，并用蓝图控制对象的旋转。

4、课件中的 Size 蓝图中用一个变量 TopPoint 控制立方体 Z 轴方向的缩放，修改蓝图，让变量 TopPoint 可以控制立方体 X、Y、Z 轴方向的缩放。

