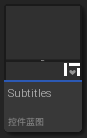
### 字幕

这里，我们将设置字幕

首先在资源浏览器中创建Widget Blueprint，命名为Subtitles



打开后首先创建一个画布面板

然后我们要在画布面板的下方中间展示字幕

首先给这个画布面板的字幕位置加一个背景模糊，锚点为下方全铺



然后给背景模糊加一个图像，通过图像的阴影设置来添加文字的阴影背景

然后添加文本，作为画布面板的子组件，层级最高。设置文本块时变量。锚点为下方全铺

最终结果如图所示





接下来创建两个动画：FadeIn和TextFade

首先是FadeIn，将轨道追踪画布面板，进行渲染不透明度的追踪

在0秒，不透明度为0

在0.25秒，不透明度为1

然后是TextFade，将轨道追踪TextBlock，进行渲染不透明度的追踪

他的工作方式是我们快速的在这里输入我们的文本块然后我们将改变到我们的下一行字幕，然后再反馈给我们

在0秒，不透明度为1

在0.1秒，不透明度为0

在0.2秒，不透明度为1

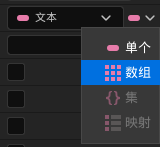
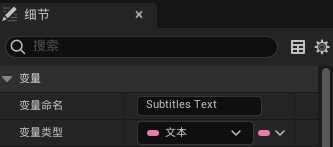
接下来进入事件图表

创建几个变量：

Text型数组变量Subtitles Text，表示字幕系统将要输出的内容，按行以一维数组方式存储

Int型Subtitle number，表示当前text在Subtitles Text数组中的索引

将单个类型转成数组类型，需要在细节中找到“变量--变量类型”，点击右侧的下箭头，选择数组



然后创建Event Construct节点

**Event Construct**节点

//在下方的slate控件构造后调用。由于从层级中添加和移除，此事件可能多次调用，具体取决于slate对象的使用方式

//如果需要一个真实的“创建后即调用一次”事件，则使用OnInitialized

//目标是用户控件

//Cosmetic.This event is only for cosmetic,non-ganmeplay actions.



Event Construct之后Play Animation,播放的动画是FadeIn

在一帧的延迟后，我们需要SetText（Text）

在Play Animation后，首先创建Delay节点，时间为0

然后将Subtitles Text变量拉入事件图表创建节点

然后给Subtitles Text，Get( a copy )，将得到的copy进行SetText（Text）

此时我们可以注意到，Get节点的下方有一个数字0，表示所Get的数组的索引，并将索引的内容作为a copy进行参数传递



们希望设置的针对每个关卡进行的系统级提示语。所以Subtitles Text的内容应当是在关卡蓝图中进行设计。我们便不在字幕系统的蓝图中进行设计

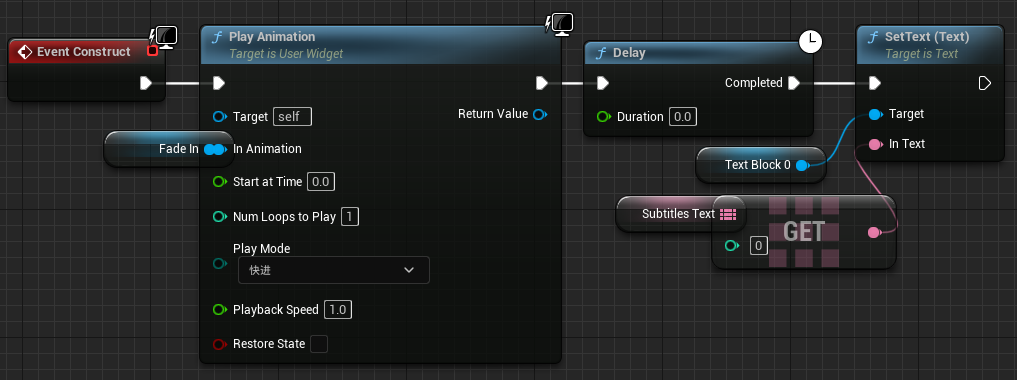
然后我们需要得到我们前面设置的文本块变量

在事件图表中右键输入get textblock后应当就能找到



从文本块变量引出连接找到SetText（Text）节点，将其进行连接

最终结果如图所示

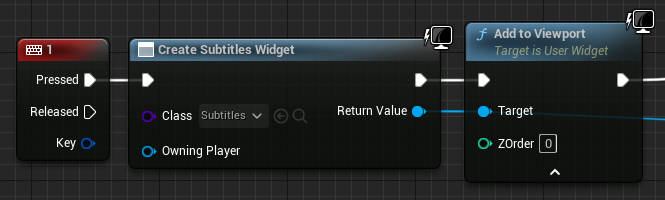


之后进入到关卡蓝图中，我们简单的展示一下是如何工作的

创建键盘1事件节点（鼠标右键后直接摁键盘上的1就能找到）



然后我们需要将Subtitles的内容加入到viewport



之后我们需要Set Subtitles Text节点

**Set Subtitles Text**节点

//Set the value of variable Subtitles Text



我们需要通过用Make Array节点来给Subtitles Text赋值

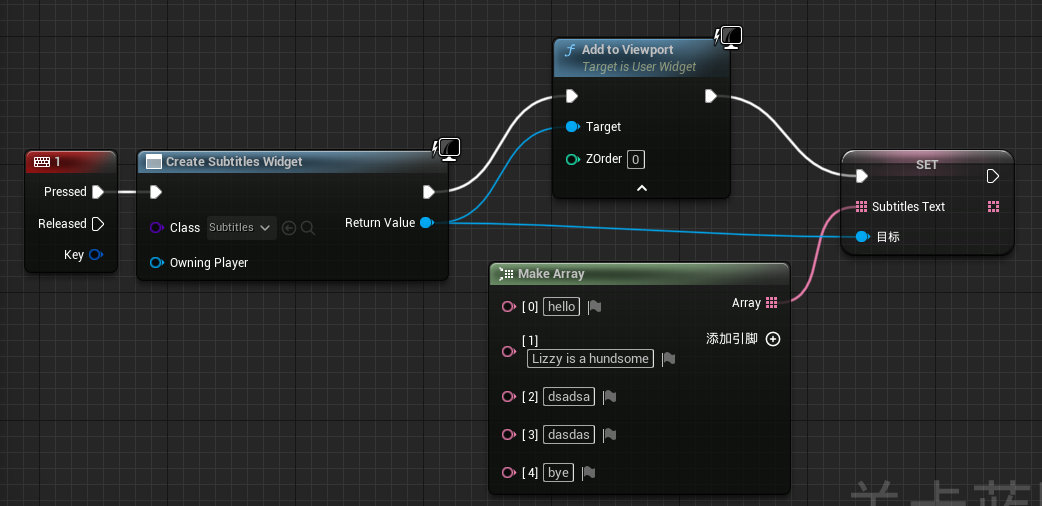
**Make Array**节点

//从一系列项目中创建数组



点击添加引脚可以增加数组的数量，并通过在方块中赋值来给SubtitlesText赋值



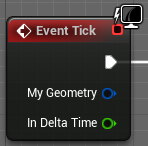


之后我们可以进入关卡中,摁1即可观察字幕

现在我们回到Subtitles

现在我们要实现的是：通过对每一帧的跟踪（Event Tick），来判断是否有某个键盘输入，如果为True，则切换Subtitle Text（因为Subtitle Text是以数组的形式保存，即只要为True，就显示下一个索引的Subtitle Text）

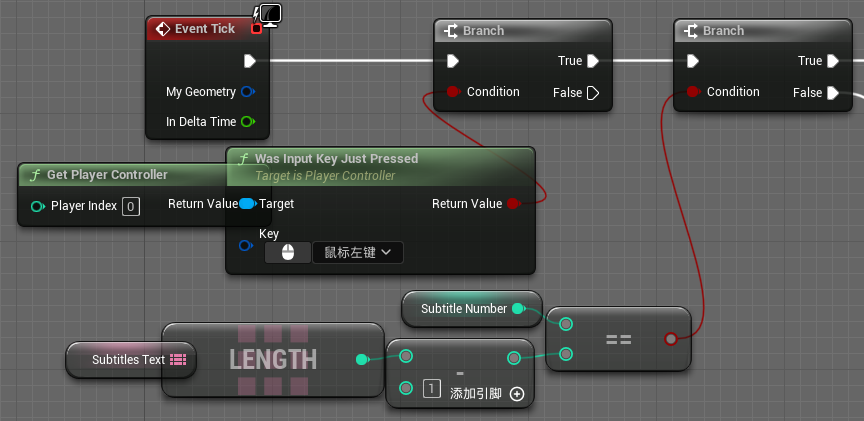
首先创建Event Tick节点，来实现每一帧的检查



然后通过GetPlayerController节点来获取游戏控制器的输入，并将输入传递给Was Input Key Just Pressed.这里我们选择Key为鼠标左键



将返回值通过branch进行判断传递。如果JustPressed为True，则再次进入分支，来判断Subtitle number和Subtitles Text的值是否相等，也就是判断SubtitlesText是否显示完成。如果为true，就播放字幕结束动画；如果为false，就设置下一条字幕进行显示



在为True的情况下，首先翻转播放FadeIn的动画，然后在0.1秒的延迟后将该widget进行移除，也就是Remove from Parent节点

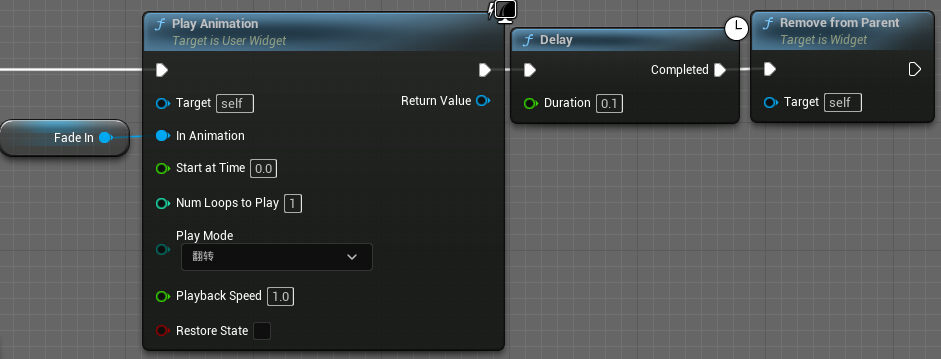
**Remove from Parent**节点

//从父控件移除控件。如果次控件被添加到玩家的画面或视口，其他也将从这些容器中移除

//目标是控件

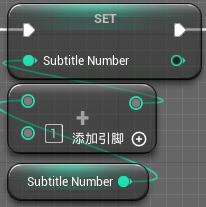


最终结果



如果字幕数组还没有播放完，则进入DoOnce节点，来保证每个索引的Text只展示一次，并完成一个字幕的展示之后重置DoOnce。期间Subtitle Number自增1

DoOnce的整个流程包括对Subtitle number的自增



还有TextFade的PlayAnimation，并在0.1秒的延迟后SetText（Text）。SetText是将SubtitleText的第Subtitle number赋值给Target：Text Block 0

在上述内容完成之后，会在0.1秒的延迟后重置DoOnce节点

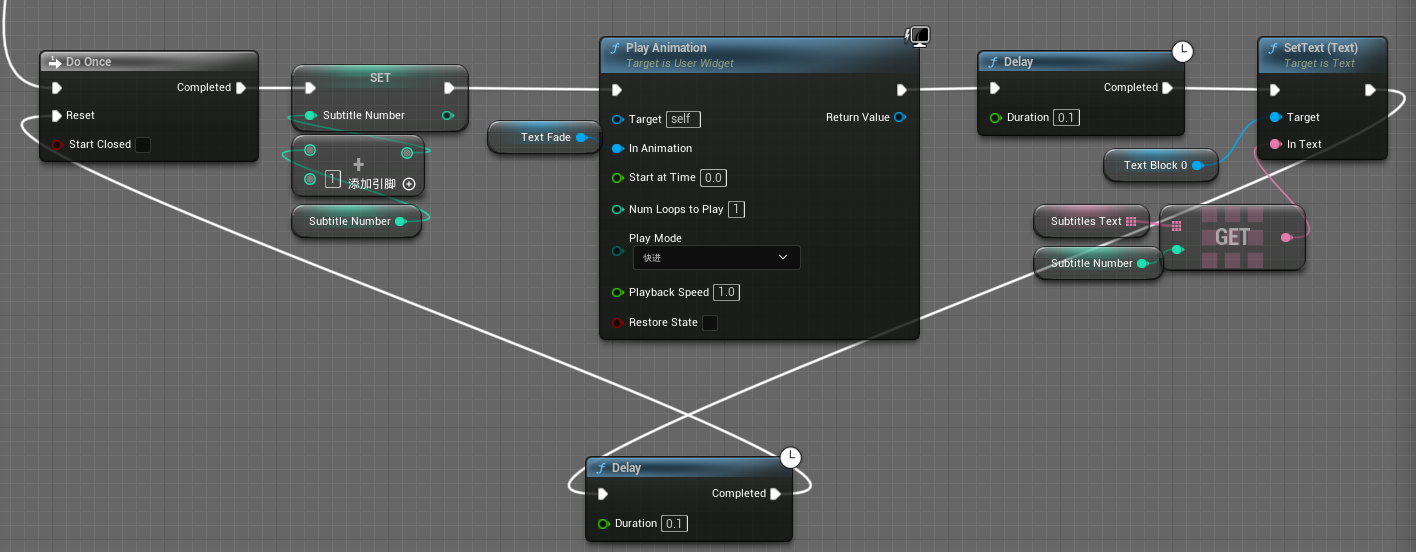
**Set Subtitle Number**节点

//Set the value of variable Subtitle number

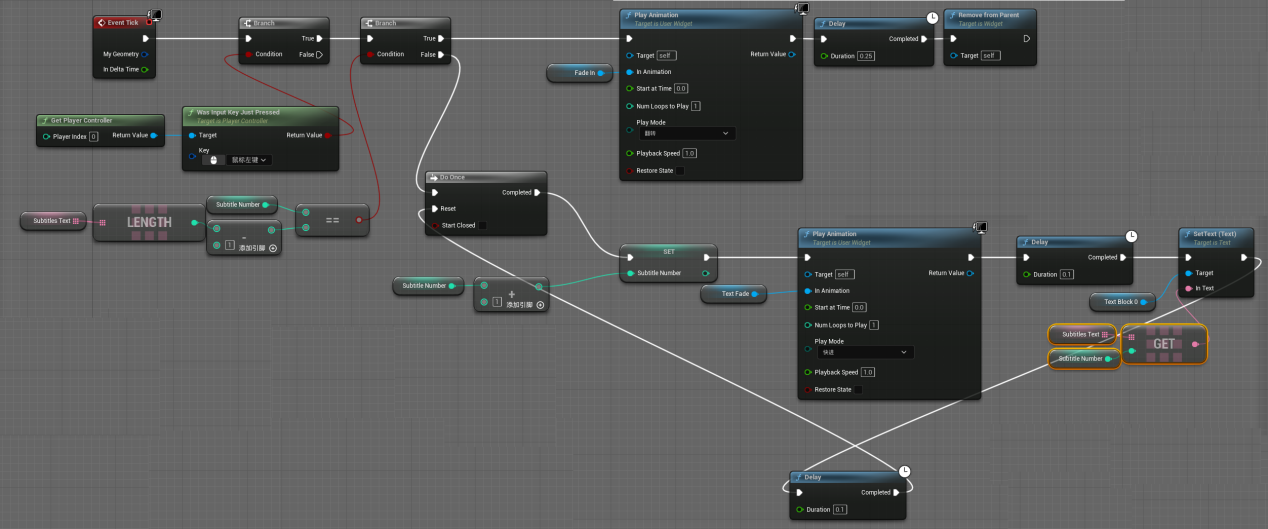


创建节点可以通过在变量中将变量拉入事件图表中，然后选择Set

最终节点族



最终结果



简单总结一下

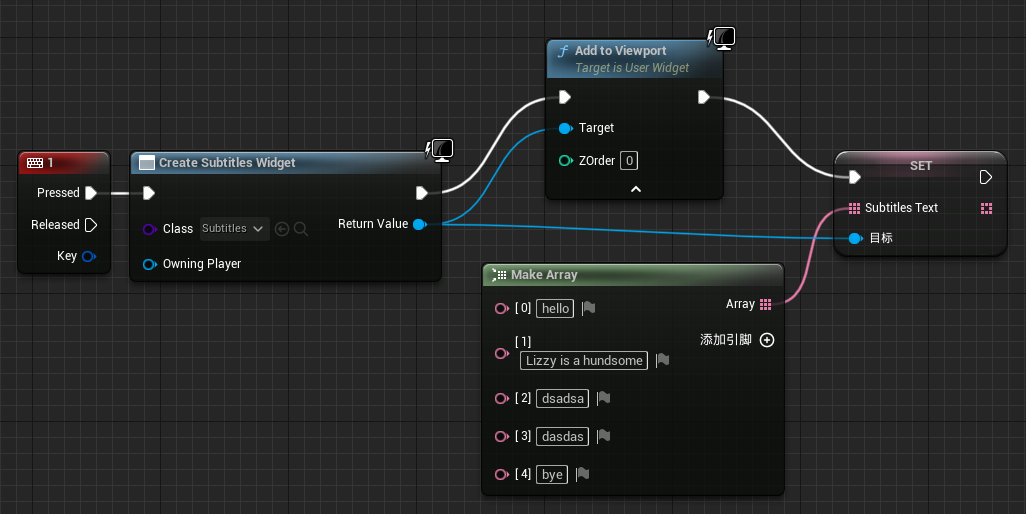
在关卡蓝图中，我们进行了SubtitleText的Set，在Subtitles蓝图中，我们实现了将SubtitleText给TextBlock的赋值

现在创建两个关卡，将新创建的两个关卡和正在编辑的关卡放在内容文件夹下的Level文件夹中

我们在此之前编辑的都是map1关卡。当然，新创建的map2和map3关卡在刚进入他们的视图中的时候是一片黑暗，需要进行简单的场景创建。这里不讨论

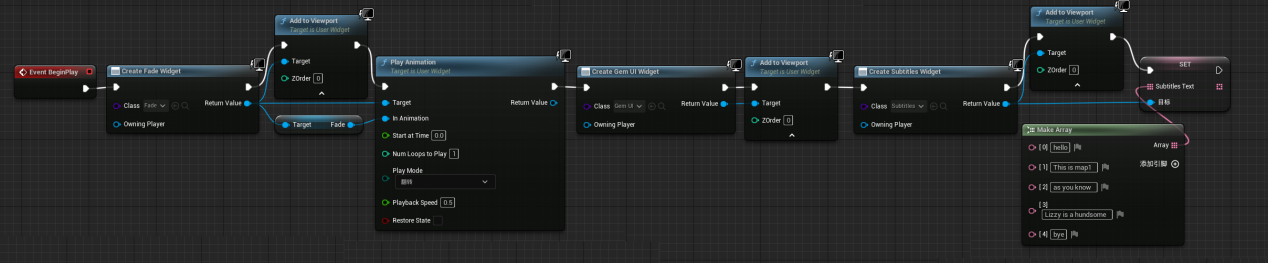
在创建新关卡时，需要实现新关卡的GemUI和ExitDoor的创建，才能实现通关和进入下一关的判定。因为通关开门的判定在Gem蓝图中

回顾map1的关卡蓝图，有关SubtitleText的节点族我们是设置成如下形式



实际上，我们可以直接将除了键盘输入部分的节点族连接至EventBeginPlay节点族的后面。这样显示字幕就不用摁数字1了

最后的关卡蓝图如图所示



每个关卡有特定的字幕。每个关卡的字幕在其各自的关卡蓝图中设置。

创建好新关卡后，若想连接各个关卡，只需要设置每个ExitDoor的NextLevel变量的参数，参数值为关卡名