欢迎继续我们的学习。在上一课的内容中，我们学习了如何持续生成隧道。在这一课的内容中，我们将学习如何添加更多的隧道。

添加更多隧道

首先要做的就是创建一个函数，让其生成一定数量的隧道。

打开BP\_TunnelSpawner，然后创建一个新的函数，将其命名为SpawnInitialTunnels。

为了生成特定数量的隧道，我们需要使用ForLoop节点。该节点可以让所连接的节点运行指定的次数。为此，让我们添加一个ForLoop节点，然后将其连接到Spawn Initial Tunnels的入口节点。

为了让ForLoop节点执行n次，我们需要将Last Index设置为n-1。

对本教程而言，我们需要生成三个隧道。为了执行三次循环，需要将Last Index的值设置为2。

注意：

如果没有手动设置First Index或Last Index，那么默认值是0。

当游戏开始的时候，玩家角色将始终在某个隧道前开始。为此，我们需要将首个隧道的位置设置为玩家角色的位置。

生成首个隧道

为了判断首个隧道是否已经生成，我们需要检查NewestTunnel是否已设置。如果还没有，那么首个隧道就还没有生成。这是因为只有当游戏生成了一个隧道后，才会设置NewestTunnel。

为了方便检查，需要在ForLoope节点之后添加一个IsValid节点。

接下来，需要获取到NewestTunnel的引用，并将其连接到IsValid节点的Input Object端口。



如果NewestTunnel还没有被设置，那么Is Not Valid端口之后的节点就会执行，否则就不会。

添加以下节点，并将其连接到IsValid节点的Is Not Valid端口：

通过以上的操作，就可以在玩家角色所在的位置生成一个隧道。

接下来，我们需要生成后续的隧道。

生成后续的隧道

添加一个SpawnTunnelAtSpawnPoint节点，然后将其连接到IsValid节点的Is Valid端口。



至此，我们最终的graph如下图所示：



总结一下：

1.ForLoop节点将总共执行三次

2.在第一次循环执行时，将在玩家角色的位置生成一个隧道

3.在随后的循环中，将在最新隧道的SpawnPoint生成隧道。

接下来在蓝图编辑器中切换到Event Graph视图，然后删除其中的SpawnTunnel节点。

接着在Event BeginPlay节点之后添加SpawnInitialTunnels节点。

现在，当游戏开始的时候，它将自动生成三个隧道。

点击蓝图编辑器上的Compile 按钮，然后返回主编辑器，并按下Play按钮来测试游戏。现在隧道变得更长了~

不过到目前为止游戏一点挑战性都没有，在下一课的内容中，我们将学习添加一些障碍物。

好了，这一课就到这里了，休息一下~