10-22

对象池模式

当角色现需要一次释放许多火球，并在之后销毁他们时，如果真的生成并销毁，那对cpu的开销将是巨大的。

所以我们使用对象池模式，在游戏一开始的时候就创建这些对象，只是是否使用他们的时候对他们的“是正在使用”属性进行修改。这样可以极大地减少cpu的开销，不过会增大内存的开销。

现在我们对角色的部分技能进行修改。

角色的一技能是释放一个扇形的区域的水球散弹。四技能是水球激光，五技能是水雨。我们实现技能是通过SpawnActor来实现的，水球技能是生成BP\_水球，水雨技能是生成BP\_水雨。



所以我们先关注水球技能。

按照PooledActor要求，应该首先修改BP-水球，添加一个bool型变量，命名为“正在使用”。

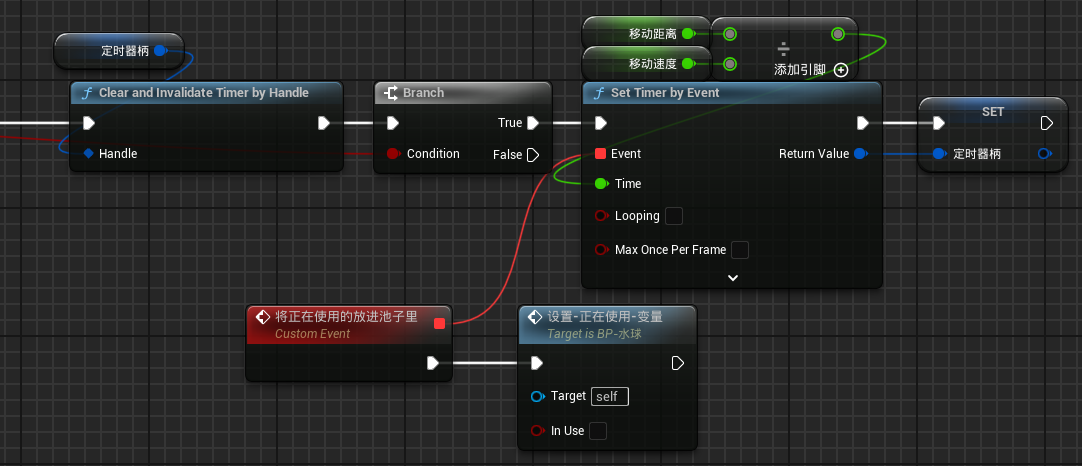
但是我们实现通用性效果，不在BP-水球上修改，创建BP-池对象。

添加一个bool型变量，命名为“正在使用”。添加float型变量，命名为存活时间

同时给他创建一个函数（或者自定义事件），命名为“设置变量-正在使用”，添加bool型形参InUse，对正在使用变量进行赋值。然后，如果他被使用，我们就启用碰撞，否则就禁用他。如果没有使用，我们也会隐藏它，或者以其他方式展示它。也有许多其他内容，具体见图片。



同时我们也要将计时器提升为变量，在我们创建一个新的之前，清除它并使其无效。



同时在EventBeginPlay时调用该自定义事件（）确保任何Actor都是从不被使用开始的。



上面是实现了PooledActor，也就是对象池里面的对象。接下来实现对象池。我们通过ActorComponent实现ActorPool

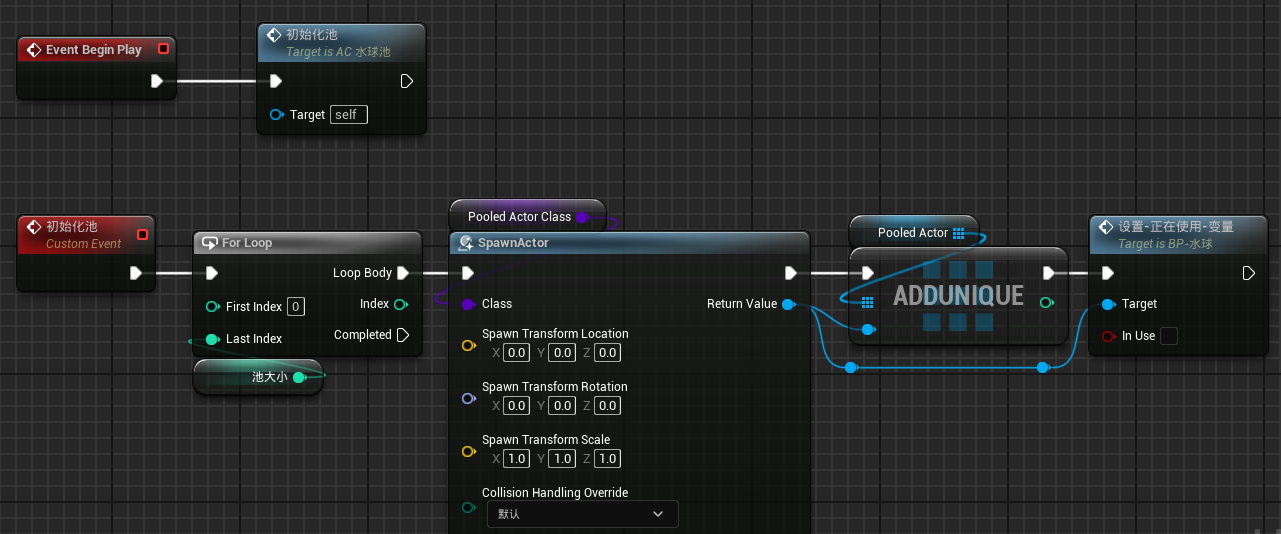
创建AC\_对象池，添加变量PooledActorClass，类型选择为BP-池对象的类引用，仅勾选可编辑实例。

添加变量“池大小”，int型，可编辑实例。

添加变量PooledActor，类型选择为BP\_水球的对象引用的数组。

接下来实现ActorPool的功能

首先是初始化池

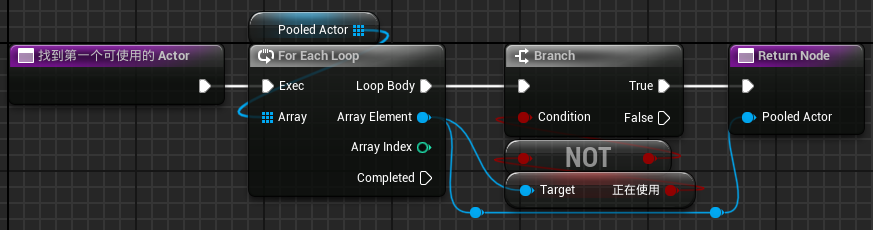


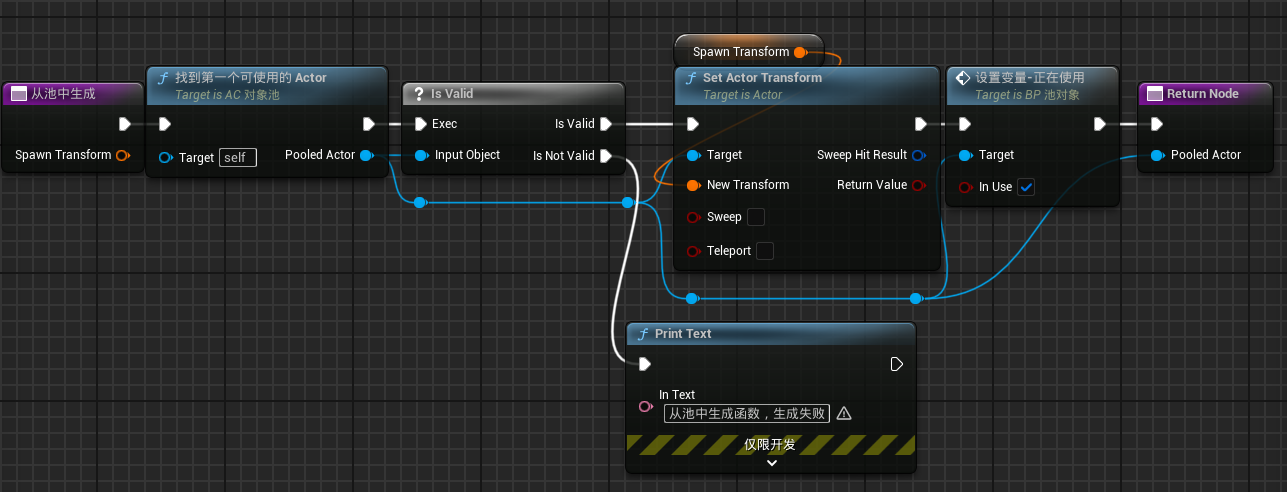
然后实现两个函数帮助我们在池中检索参与者

创建函数，命名为“找到第一个可使用的Actor”，返回值为PooledActor，类型为BP\_池对象的对象引用。

创建函数，命名为“从池中生成”。这个函数负责将参与者设置为正在使用。函数的输入为Transform（变换）类型的形参，返回值为PooledActor，类型为BP\_池对象的对象引用。

下面就来实现这些函数。





接下来对BP\_水球进行设置

首先修改水球的父类为BP\_池对象。

这之后需要对水球进行较多的修改。

将水球的存活时间（来自父类的变量）设置为移动距离/移动速度

在DestroyActor时不destroy，而是设置“正在使用”变量

然后是设置速度。设置速度不应该在水球的EventBeginPlay时设置。

这个AC\_水球池可以更广泛的应用到所有PooledActor类上，使其更具有通用性。我们可以改名为AC\_对象池