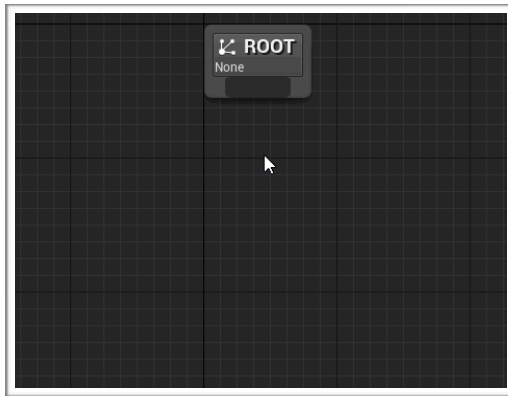


欢迎继续我们的学习。

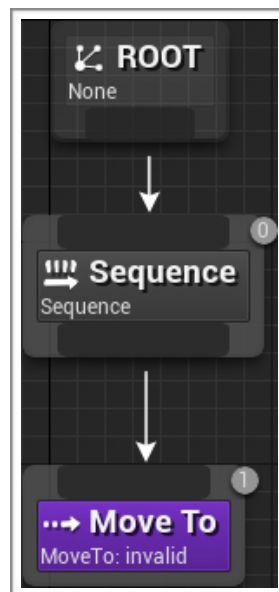
在这一课的内容中，我们首先需要让Pawn角色移动到某个随机的位置。

移动到某个随机位置

创建一个Sequence节点，然后将其连接到Root节点上。



接下来我们需要移动Pawn。创建一个MoveTo节点，然后将其连接到Sequence节点。该节点的作用是将Pawn移动到某个特定的位置或角色。



接下来创建一个Wait节点，然后将其连接到Sequence节点。注意这里要把Wait节点放在MoveTo节点的右边。在行为树中节点的顺序很重要，因为子节点将从向右执行。

注意：有一种简单的方法来确认节点的执行顺序，那就是查看每个节点右上角的数字编号。数字越小的节点越早被执行。

恭喜你，现在我们已经创建了自己的第一个行为树！它的作用是让Pawn角色移动到某个特定的位置，然后等待5秒。

为了移动Pawn角色，我们需要指定一个特定的位置。而MoveTo节点只支持从blackboards中接收数值。因此，我们需要创建一个blackboard。

创建blackboard

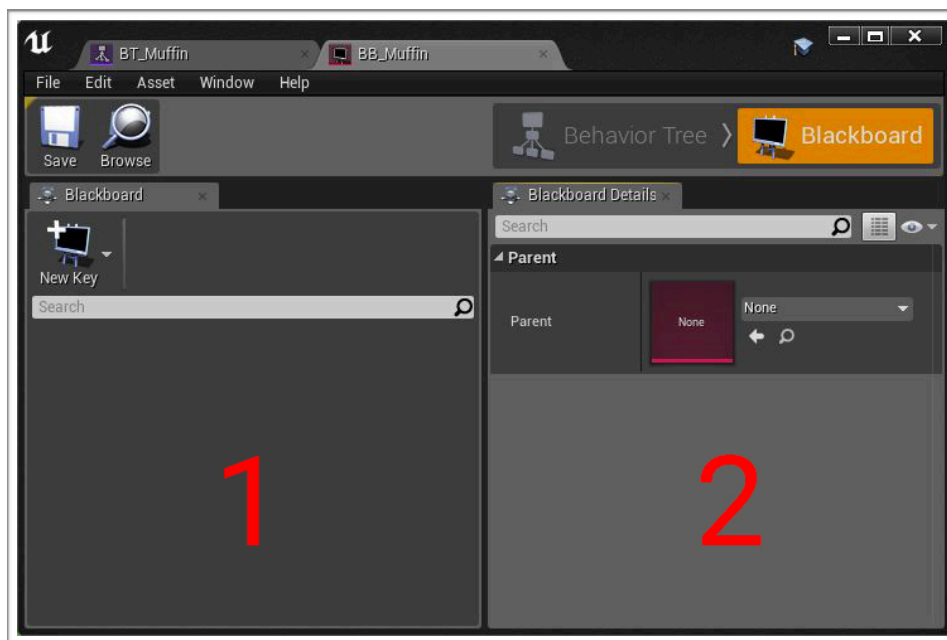
blackboard是一种游戏资源，它的唯一作用就是保存变量（在虚幻4中用keys表示）。我们可以把blackboard看做AI的存储记忆。

尽管blackboard不是必需的，不过它提供了一种很便捷的保存和读取数据的方式。之所以说blackboard很方便，是因为行为树中的很多节点只接受blackboard中的key。

为了创建blackboard，我们需要返回Content Browser,然后选择Add New\Artificial Intelligence\Blackboard。将其命名为BB_Muffin，然后将其打开。

Blackboard编辑器

Blackboard编辑器包含了两个面板：



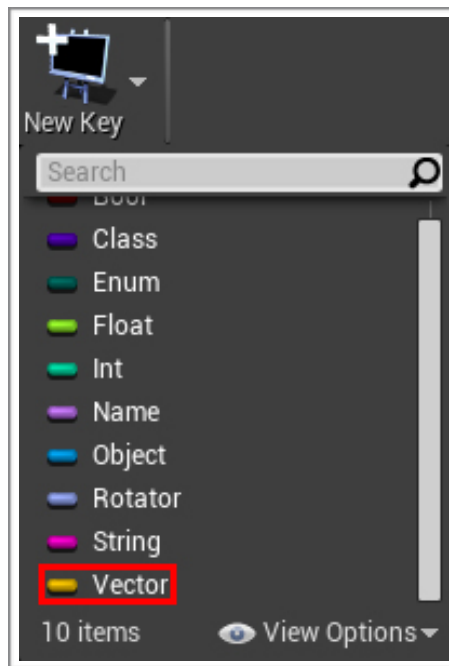
1.Blackboard:该面板将显示一系列的key

2.Blackboard Details:该面板将显示所选中的key的属性

接下来我们需要创建一个key，以保存目标位置。

创建目标位置的key

考虑到我们存储的位置信息处在3D空间，我们需要将其保存为一个vector变量。点击New Key，然后选择Vector，将其命名为TargetLocation。



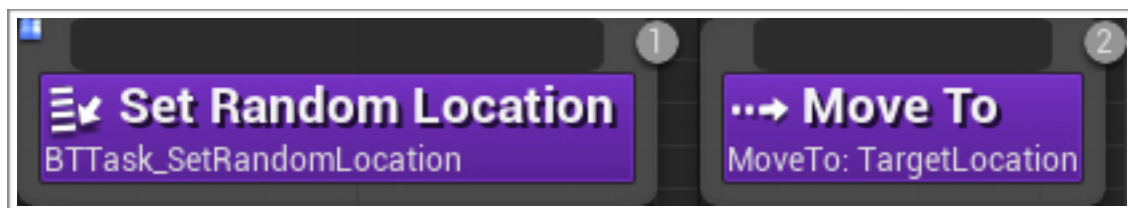
接下来，我们需要创建一个随机的位置，并将其保存在blackboard中。为此，我们将用到第三种行为树节点：service。

Services

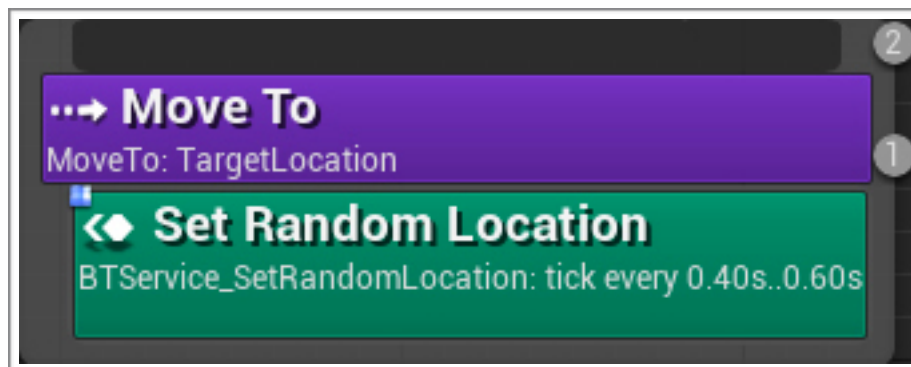
和task类似，service用于完成某件任务。但是和task不同的是，task通常是让Pawn角色执行某个动作，而service则通常是用于执行检查或者更新blackboard。

services并非单独的节点，它们通常和tasks或composites关联在一起。通过使用services，可以让行为树的组织更加有条理。

当我们只使用task时，行为树会是这样的：



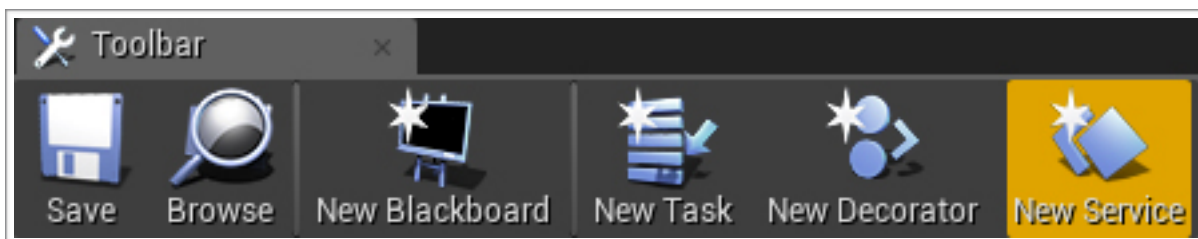
而当使用service时，行为树会是下面这样的：



好了，接下来我们将创建一个service，用于生成一个随机的位置。

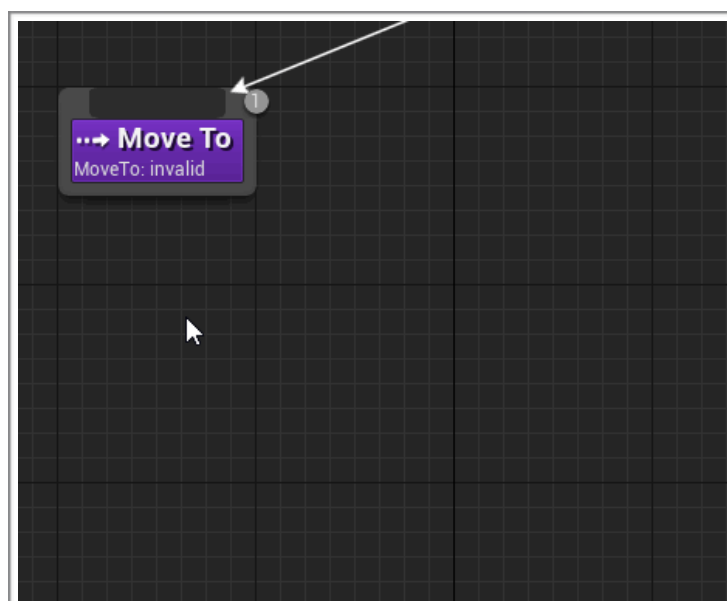
创建Service

返回BT_Muffin， 点击New Service。



这样我们就创建了一个新的service,并且会自动将其打开。

将其命名为BTService_SetRandomLocation。（注意：需要返回Content Browser来重命名）。该service只会在Pawn角色需要移动的时候执行。为此，我们需要将其关联到MoveTo节点上。打开BT_Muffin， 右键单击MoveTo， 选择Add Services\BTService Set Random Location。



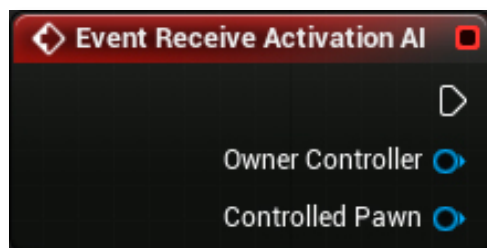
现在，当MoveTo节点激活的时候，BTService_SetRandomLocation也会激活。

接下来我们需要生成一个随机的目标位置。

创建随机的位置

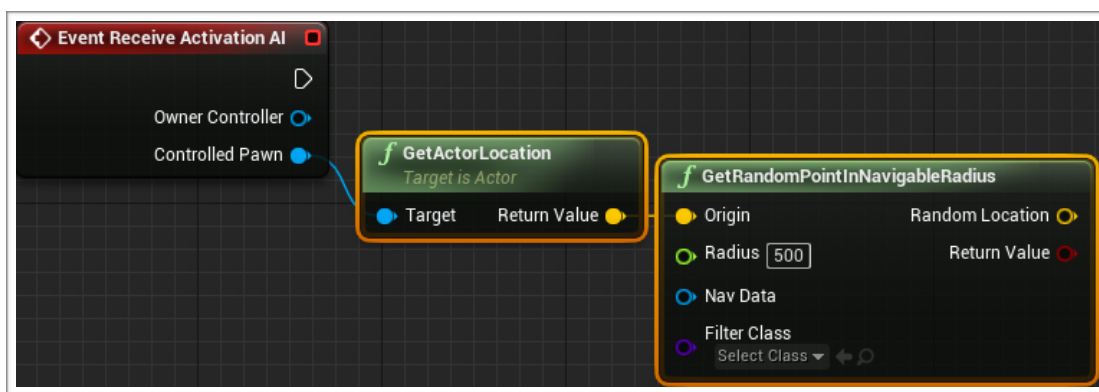
打开BTService_SetRandomLocation。

为了知道service何时激活，我们需要创建一个Event Receive Activation AI节点。这样，当service的父节点（所关联的节点）激活的时候，就会执行该节点。



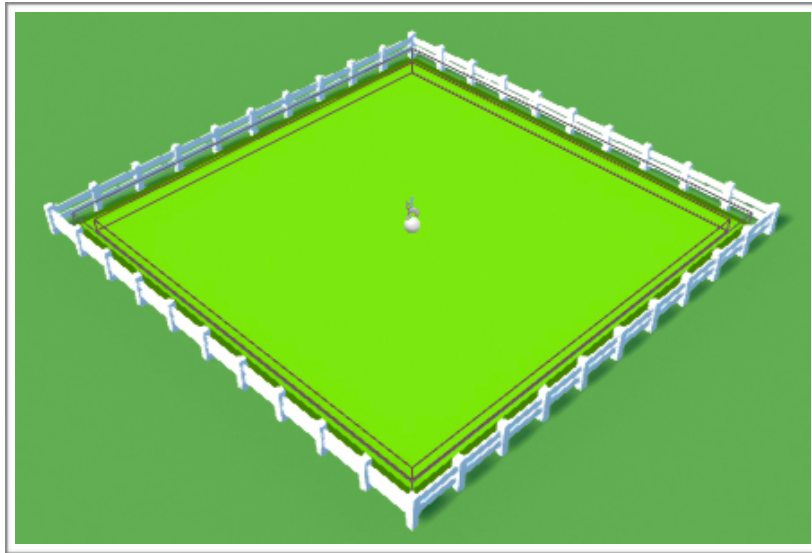
注意：除了Event Receive Activation AI节点，还有另外一个Event Receive Activation节点也可以完成同样的任务。两者的区别在于Event Receive Activation AI节点也提供了Controlled Pawn。

为了生成随机位置，还需要添加如下的高亮节点。注意需要将Radius设置为500。



这样，我们就获得了距离Pawn角色500个单位内的随机可寻路的位置。

注意：GetRandomPointInNavigableRadius使用寻路数据（称之为NaMesh）来判断某个点是否可达。在本教程中已经创建好了NavMesh，大家在Viewport中选中Show\Navigation，就可以看到NavMesh。



如果我们希望创建自己的NavMesh，那么可以创建一个Nav Mesh Bounds Volume。使用缩放工具，从而让它涵盖我们希望到达的区域范围。

好了，本课的内容就先到这里了。
我们下一课继续~

讨论群-笨猫学编程QQ群：
375143733

答疑论坛：
<http://www.vr910.com/forum.php?mod=forumdisplay&fid=52>

知乎专栏：
<https://zhuanlan.zhihu.com/kidscoding>

新浪博客：
<http://blog.sina.com.cn/eseedo>

Github：
<https://github.com/eseedo>

个人网站：
<http://icode.ai/>