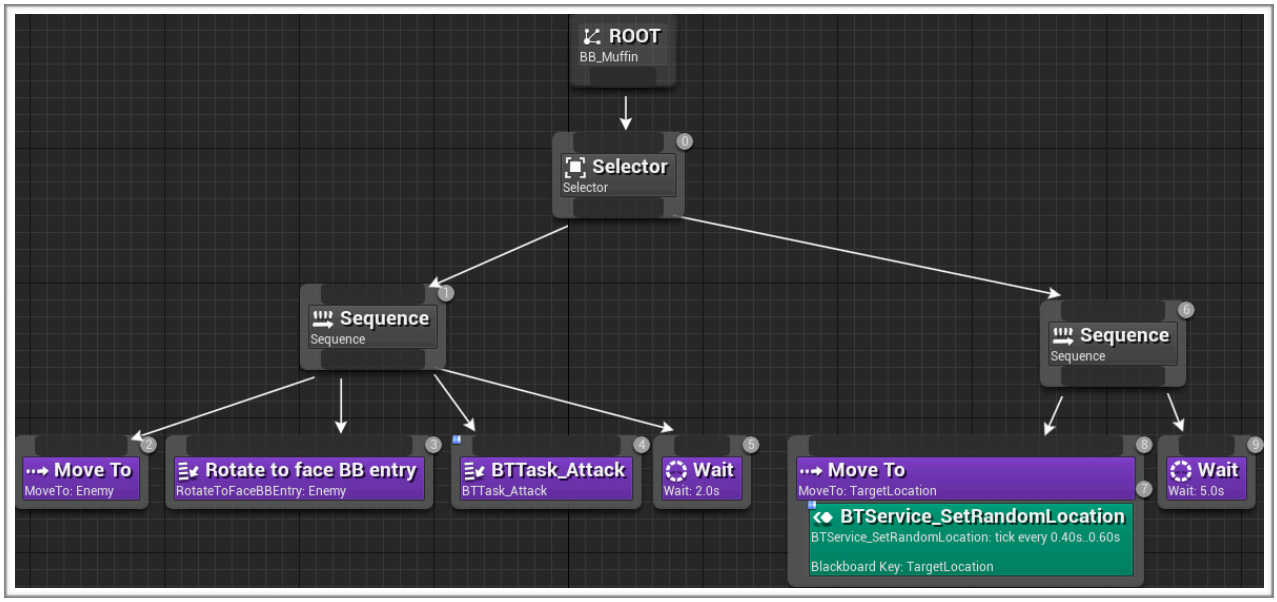
欢迎继续我们的学习。

本课也将是人工智能部分的最后一课。在本课的内容中，我们将把攻击和漫游的子行为树整合在一起。

将子行为树整合在一起

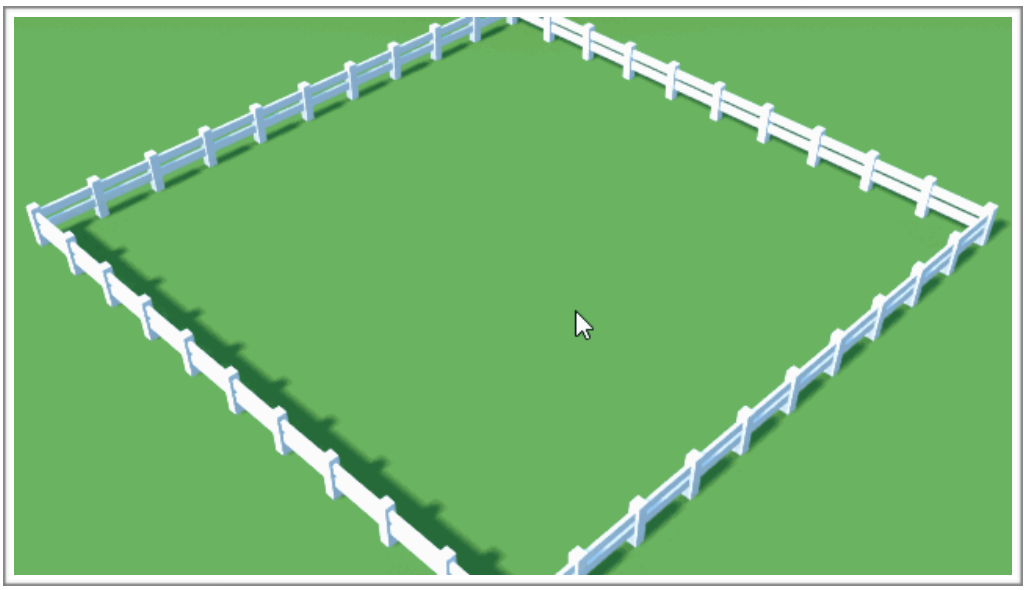
为了整合子行为树，我们需要用到Selector 这个composite。和Sequence节点类似，Selector中的子节点同样是从左向右执行的。不过不同的是，Selector将在子节点顺利完成之后停止，而不是在失败之后停止。通过使用Selector，我们可以确保行为树每次只会执行一种子行为树。

打开BT_Muffin，然后在Root节点之后创建一个Selector。随后使用以下的方式来连接子行为树：



通过以上的设置，一次只会执行一个子行为树。下面是每个子行为树的执行方式：
1.Attack: Selector将首先执行攻击子行为树。当完成所有任务后，Sequence也将顺利完成。Selector会检测到这一点，然后停止执行对应的Sequence。这样就不会紧接着执行漫游的子行为树。
2.Roam:Selector将优先执行攻击子行为树。在执行攻击的子行为树时，如果Enemy没有被设置，那么MoveTo节点就会执行失败。此时该部分的Sequence将执行失败，因此Selector将会执行另一个子行为树，也就是漫游。

返回主编辑器，点击Play。生成一些松饼，并测试一下效果。



不过问题来了，为什么松饼并不会立即攻击其它的松饼?
在传统的行为树中，将会从根节点每帧开始执行。这就意味着每次都会先尝试攻击子行为树，然后才是漫游子行为树。这就意味着如果Enemy的值发生变化，那么行为树的执行也会发生变化。

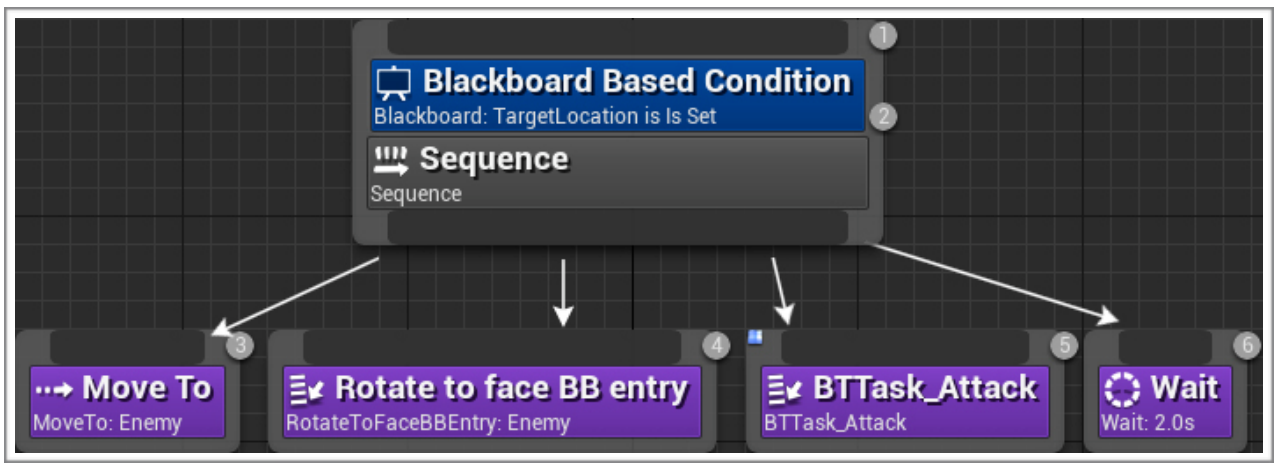
不过虚幻4中的行为树工作原理并非如此。在虚幻4中，总是从上一次执行的节点处开始。因为AI Perception不能立即感知其它的角色，所以漫游子行为树会开始执行。然后行为树在切换到攻击子行为树之前，需要等待漫游的子行为树结束。

为了修复这个问题，我们需要用到最后一种类型的节点：decorators。

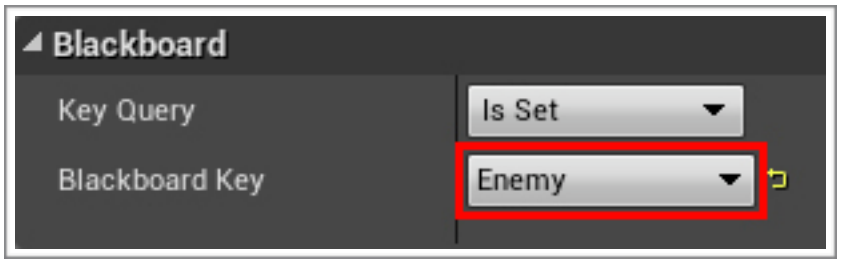
创建Decorator

和service类似，decorator将会关联到task或composite.通常来说，我们会使用decorator来执行检查。如果检查的结果是true，那么decorator将会返回true，反之则返回false。通过这种方式，我们可以控制decorator的父节点是否可以执行。

decorator还可以用来中止(abort)某个子行为树。这就意味着一旦Enemy被设置后，可以停止之前的漫游子行为树。这样就可以让松饼在检测到敌人之后立即开始攻击。
为了使用abort，我们可以使用Blackboard decorator。打开BT_Muffin，然后右键单击攻击子行为树的Sequence节点。选择Add Decorator\Blackboard。这样就可以在Sequence节点上关联一个Blackboard decorator。



接下来选中Blackboard decorator，然后在Details面板中将Blackboard Key设置为Enemy。



这里将会检查Enemy是否被设置。如果还没有被设置，那么decorator就会执行失败，并导致Sequence执行失败。这样漫游子行为树就会被执行。

为了中止漫游子行为树，我们需要用到Observer Aborts设置。

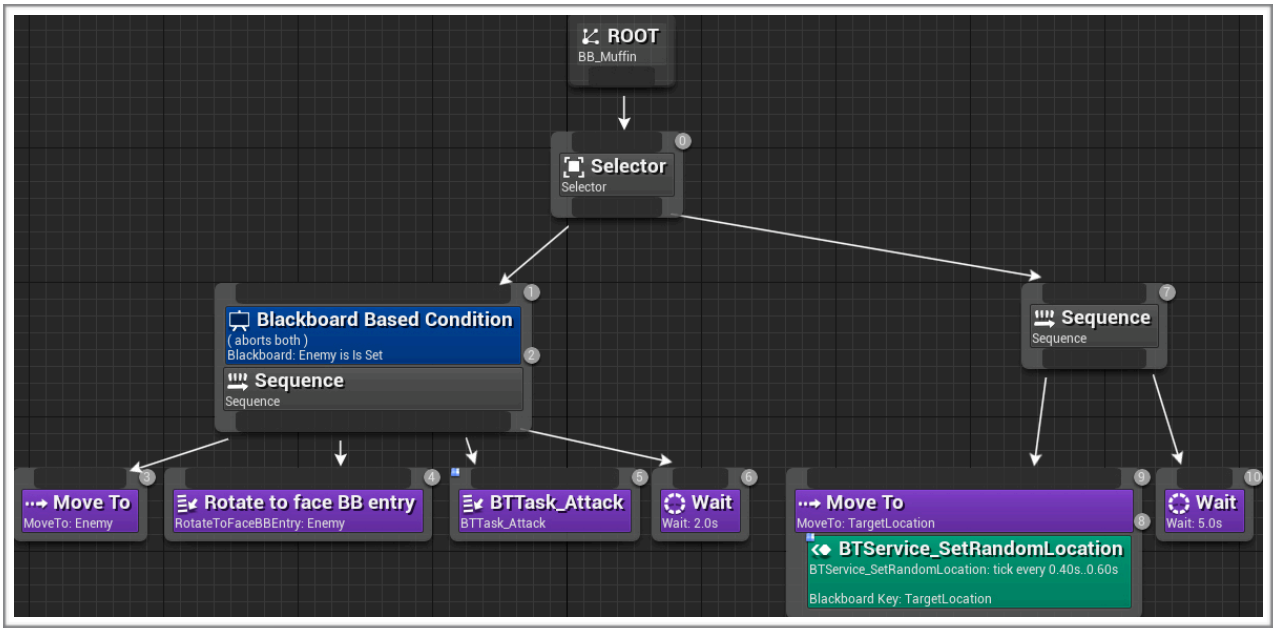
使用Observer Aborts

当所选的blackboard key被改变时，Observer aborts就会中止子行为树。有两种类型的aborts:
1.Self
这种设置将会在Enemy变得无效时让攻击子行为树中止自身。如果Enemy在攻击子行为树完成之前死了，就会发生这种情况。
2.Lower Priority
使用这种设置，当Enemy被设置时将中止低优先级的子树。因为漫游子树在攻击子树之后，因此其优先级更低。

这里将Observer Aborts设置为Both，这样两种类型的Abort都会被启用。

现在，当没有敌人的时候，攻击子树就会立即切换为漫游子树。同样的，当检测到敌人的时候，漫游子树也将立即切换为攻击子树。

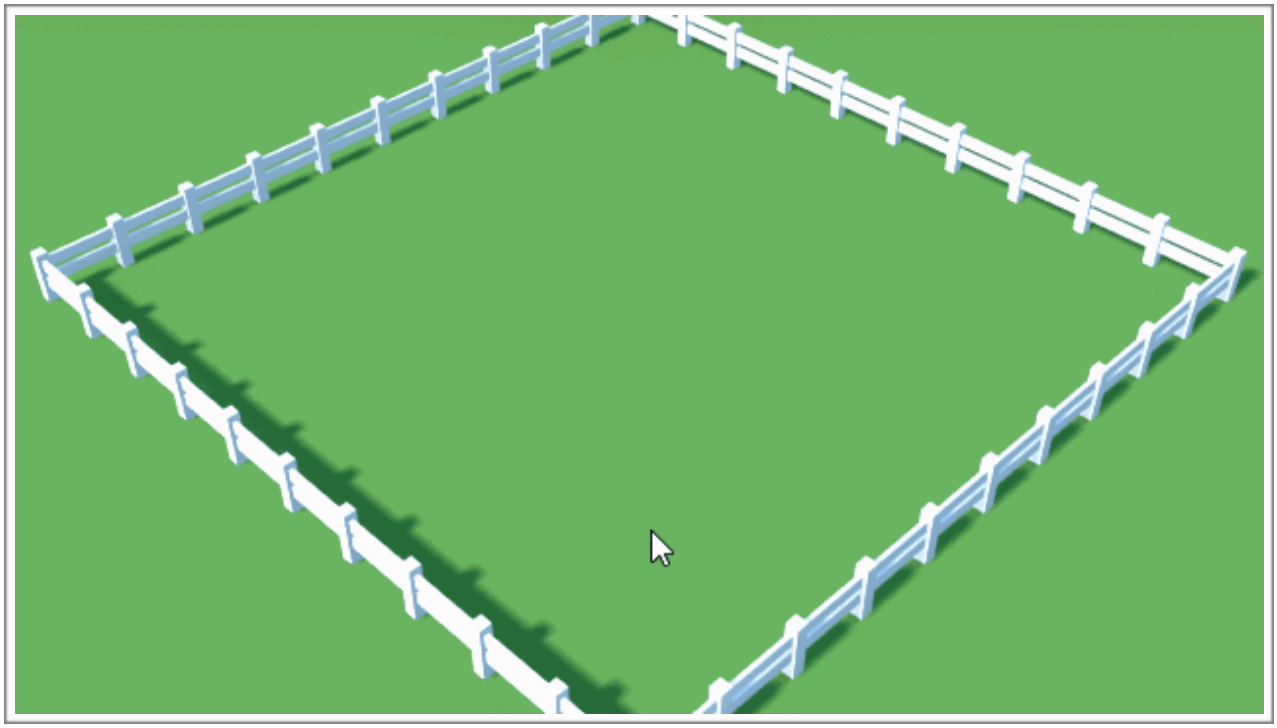
完整的行为树如下图所示：



这里大概解释下攻击子树的执行方式：
1.当Enemy被设置时，Selector将执行攻击子树
2.当Enemy被设置时，Pawn角色将会转向敌人，并向敌人靠近
3.随后角色将执行攻击
4.最后角色将等待两秒

接下来大概解释下漫游子树的执行方式：
1.当攻击子树执行失败的时候，Selector将会执行漫游子树。这里，执行失败的情况就是没有设置Enemy
2.BTService_SetRandomLocation将会生成一个随机的位置
3.Pawn角色将会向所生成的位置靠近
4.随后角色将等待5秒

关闭BT_Muffin，然后点击Play预览游戏。
在场景中生成一些松饼，然后看它们欢乐的战斗~



好了，关于游戏中的人工智能，就讲到这里。

在最后一部分的内容中，我们将学习如何创建一个简单的FPS射击游戏~

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/