

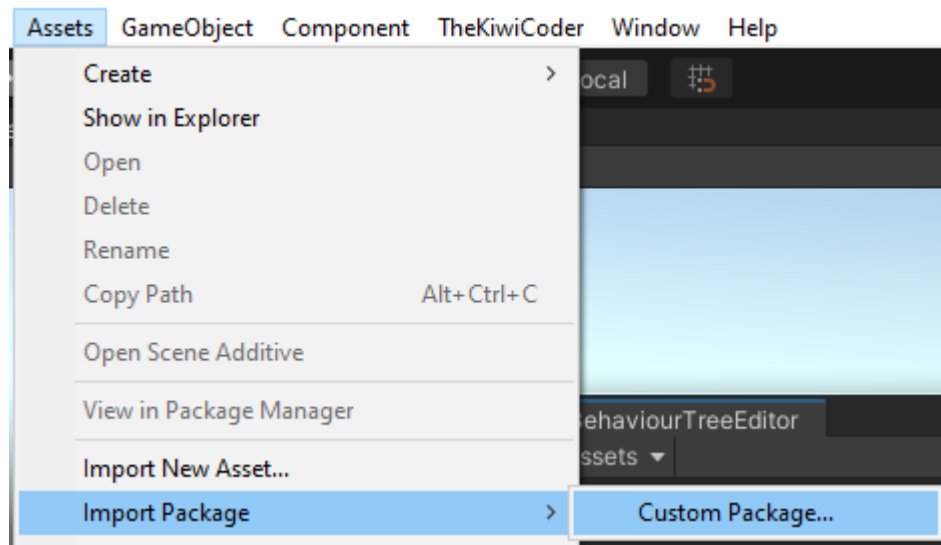
行为树编辑器

目录

1. [安装包](#)
2. [概述](#)
3. [动作节点](#)
4. [复合节点](#)
5. [装饰节点](#)
6. [编辑](#)
 1. [导航](#)
 2. [添加节点](#)
 3. [创建节点](#)
 4. [链接节点](#)
 5. [检查员视图](#)
 6. [黑板视图](#)
 7. [资产菜单](#)
7. [设置](#)

安装包

下载 zip 文件后，从资产菜单中导入包：资产->导入包->定制包装 并选择行为树.unitypackage 文件。

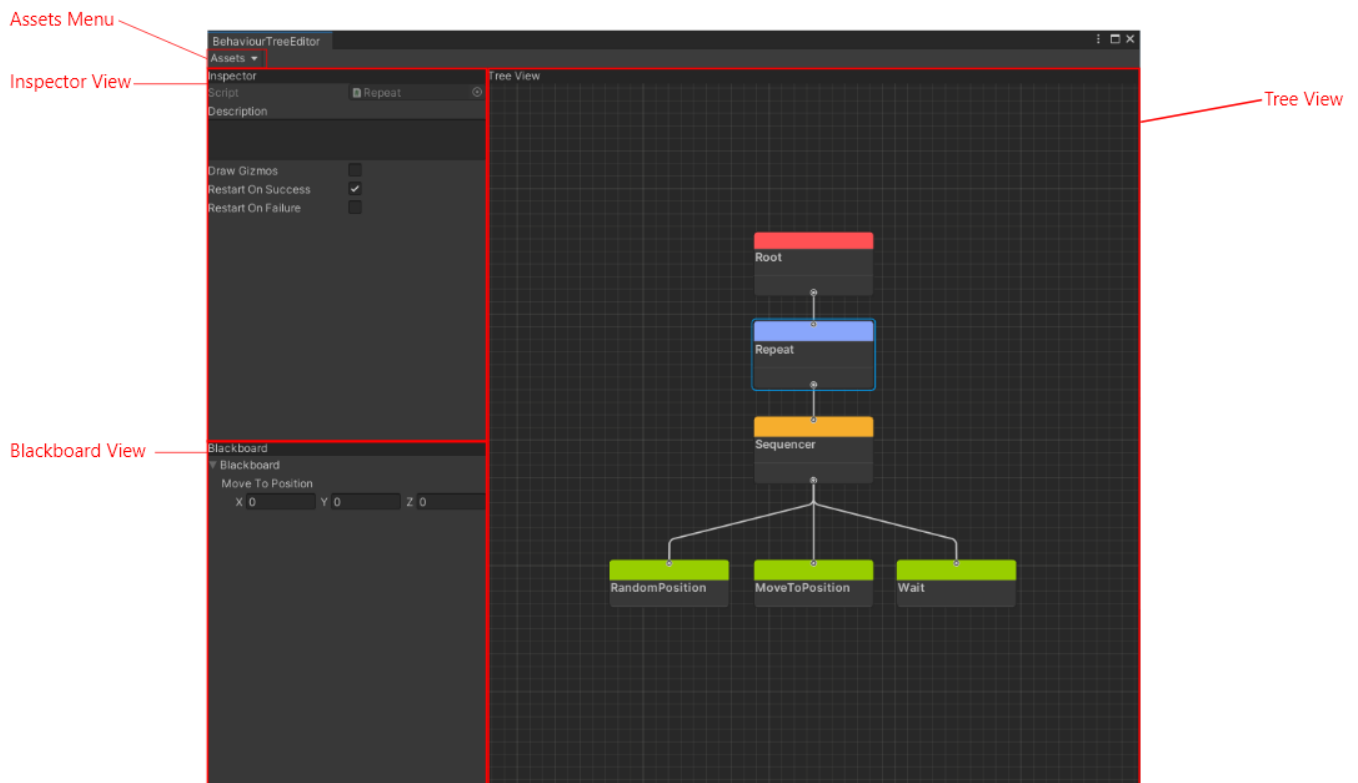


所有文件都将安装到 资产/KiwiCoder/行为树

可以在以下位置找到示例场景和行为树 资产/KiwiCoder/行为树/示例

笔记： 在撰写本文时用于测试的版本是 Unity 2020.3.13f1 LTS。

概述



行为树非常擅长创建简单和复杂的人工智能行为。行为树的主要优点之一是它们能够将复杂的行为分解为非常简单的单元，称为动作，然后可以以各种方式组合、重用和重新排列以创建新的行为。

行为树从根节点开始自上而下执行，向下遍历到它的子节点。树中的每个节点都可以返回三种状态之一；失败、成功或运行。由父节点决定如何解释这些返回状态，并决定在发生故障时下一步做什么。

动作节点

动作节点构成了树的大部分。他们总是在树的叶子上，没有孩子。这些节点负责改变世界的状态，并最终代表代理的能力。代理可以做的任何事情都应该表示为一个动作。您将创建许多新类型的动作来表示游戏中特定于代理的行为。

动作的粒度可以大也可以小，但保持动作尽可能小以增加整个树的重用通常是个好主意。一旦你为你的 AI 构建了一个体面的动作库，它们就可以组合在一起以创建复杂的 AI 行为。

包中包含的一些示例操作节点类型是：

- Log - 将消息记录到控制台 Wait - 等待一段时间
- Breakpoint - 执行时暂停编辑器
- RandomFailure - 将失败返回给它的父级，随机变化值介于 0 和 1 之间 RandomPosition - 在最小最大范围内生成一个随机位置并将该值分配给黑板。
- MoveToPosition - 从黑板上读取一个位置，并将导航网格代理目标设置为该位置。

复合节点

复合节点代表树的控制流，可以有很多子节点。由每个复合节点类型决定子节点的执行顺序。复合节点还监视其子节点的故障状态。有很多方法可以处理故障和执行顺序，这就是将每个复合节点与其他节点区分开来的原因。行为树中重复出现的两种主要复合节点类型是排序器和选择器（又名回退），但是通常需要许多其他复合节点类型来创建复杂的逻辑。一个例子是能够同时执行多个子节点的并行节点。

包中包含的复合节点类型有：

合成的 节点 类型	执行顺序	成功	失败
-----------------	------	----	----

合成的 节点 类型	执行顺序	成功	失败
音序器	从左到右一次执行一个孩子	当所有 孩子们 返回 成功	当一个孩子 返回失败
选择器	从左到右一次执行一个孩子	当一个 孩子 回报 成功	当所有孩子 返回失败
随机的 选择器	随机选择一个孩子执行	当。。。的时候 孩子 回报 成功	当孩子 返回失败
平行线	同时“一次”执行所有子级。多个孩子可以同时处于运行状态。	当所有 孩子们 返回 成功	当一个孩子 返回失败。 剩下的孩子 被中止。
打断 选择器	与 Selector 类似，但每个刻度都会不断重新评估子级。如果具有更高优先级的孩子成功，则当前正在运行的孩子将被中止。	当所有 孩子们 返回 成功	当一个孩子 返回失败

装饰节点

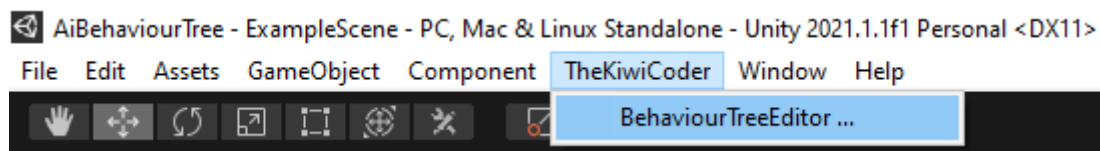
装饰器节点只有一个子节点，用于隐藏或修改其子节点的返回状态。例如，它们对于翻转孩子的返回状态非常有用，类似于“！”大多数编程语言中的运算符。然而，它们也能够其中包含复杂的逻辑，例如在多次返回失败时重新启动子项。或者让一个跑了太久的孩子流产。

包中包含的装饰器节点类型是：

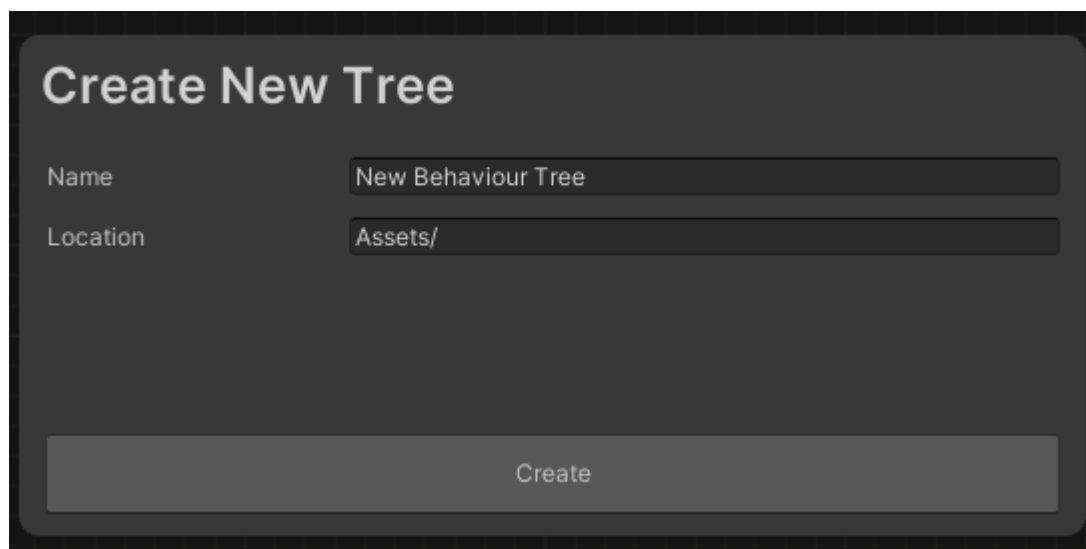
- 失败 - 总是向父级返回失败成功 - 总是向父级返回
- 成功
- Inverter - 将孩子的返回状态从成功反转为失败，反之亦然重复 - 如果它的孩子失
- 败，它会再试一次，并将运行返回给它的父母。超时 - 如果在给定的超时期限后尚
- 未完成，则中止它的子项。

编辑

安装包后，可以通过菜单访问BehaviourTreeEditor



首次打开行为树编辑器时，会出现创建新行为树的提示。为树选择一个名称并选择一个位置，然后按“创建”。行为树也可以通过标准项目窗口创建菜单创建。



导航

节点可以选择目录，也可以选择框选并在画布周围拖动。

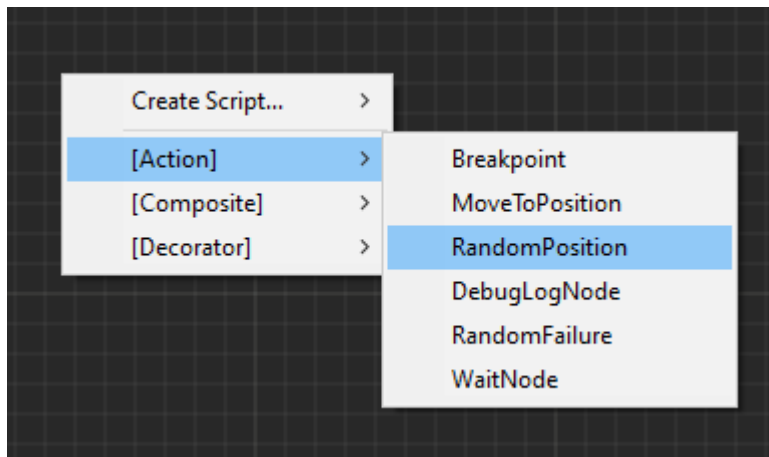
选择和整子树，双击它的父级，子级将被添加到选择中。在空白处单击以再次取消选择。

键盘快捷键：

关键代码	行动
删除	删除当前选中的节点
一种	将画布上的所有节点框起来
哦	框画画布原点
[框住当前选择的子节点

添加新节点

通过右键单击画布中的任意位置并选择要添加的节点类型，可以通过上下文菜单将新节点添加到树中。节点分为三个子菜单：操作、组合和装饰器。

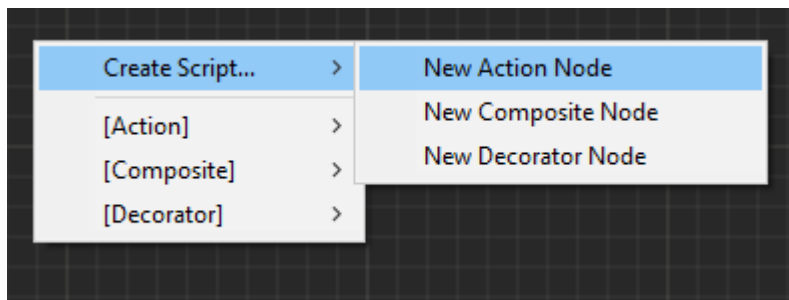


创建新节点

内置的节点类型只能让你到此为止。当您设计和创建自己的节点类型时，行为树的真正威力就来了。节点只是标准的 c# 脚本，它们继承自三种基本类型之一，**动作节点**，**复合节点**，或者 **装饰节点**。

创建一个新的脚本可以手动完成，只要它继承了前面提到的基本类型，那么它们就会被编辑器自动选取和注册，并会出现在上下文菜单中。

为了加快进程，也可以从上下文菜单中创建新的节点脚本。这些使用在以下位置找到的脚本模板[资产/TheKiwiCoder/BehaviourTree/ScriptTemplates](#)



链接节点

要将一个节点添加为另一个节点的子节点，请拖动 **输出** 父节点的 **输入** 子节点的。只注意**复合节点** 类型可以有多个孩子。

检查员视图

检查器视图显示当前选定节点的所有公共属性。要在检查器中显示节点的属性，请确保选择节点标题上方的节点中间。所有节点都有一个描述字段，可以在检查器中设置。只需开始在描述框中输入，文本就会出现在节点的标题下。

黑板视图

黑板视图显示黑板的所有公共属性。这些值可以由树的任何级别的单个节点读取和写入。请注意，此黑板不是通用的，需要对其进行修改以包含对您的游戏有意义的属性。存在用于说明目的的示例属性。

- MoveToPosition - Vector3 写入 随机位置 节点，并从移动到位置 节点

资产菜单

资产菜单下拉菜单显示位于项目中的所有行为树。使用它可以在项目中的不同树之间快速跳转。还有一个额外的菜单选项可以从这里创建一个新的行为树。

设置菜单

行为树编辑器有各种“硬编码”设置。这些可以通过“行为树”类别下的标准项目设置菜单访问。

