**混合树**

游戏动画中的一项常见任务是在两个或更多相似运动之间进行混合。也许最熟知的示例就是根据角色的速度来混合行走和奔跑动画。另一个示例是角色在奔跑期间转向时向左或向右倾斜。

区分过渡与混合树十分重要。虽然两者都用于创建平滑动画，但它们用于不同种类的情况。

* \_\_过渡\_\_用于在给定时间内从一个动画状态平滑过渡到另一动画状态。过渡作为[动画状态机](file:///E:\\UnityDocumentation\\Manual\\AnimationStateMachines.html)的一部分指定。如果过渡很快，从一个运动到完全不同运动的过渡通常没有问题。
* \_\_混合树\_\_允许通过不同程度合并多个动画来使动画平滑混合。每个运动对最终效果的影响由一个\_混合参数\_控制，该参数只是与 Animator Controller 相关联的数字[动画参数](file:///E:\\UnityDocumentation\\Manual\\AnimationParameters.html)之一。为了使混合后的运动合理，要混合的运动必须具有相似的性质和时机。混合树是动画状态机中的一种特殊状态类型。

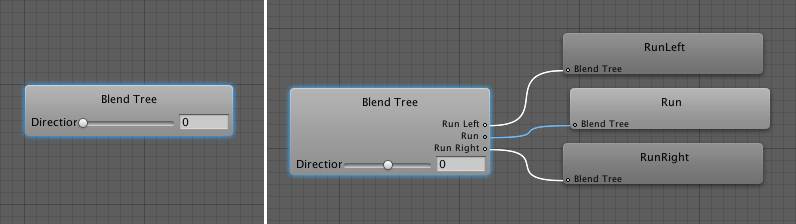
例如，类似运动可能是各种行走和奔跑动画。为了使混合取得良好效果，剪辑中的动作必须在标准化时间内的相同点进行。例如，可将行走和奔跑动画对齐，使得脚与地面的接触时刻发生在标准化时间内的相同点（例如，左脚接触点为 0.0，右脚接触点为 0.5）。由于使用标准化时间，因此剪辑长度不同并不要紧。

**使用混合树**

要开始使用新混合树，您需要：

1.右键单击 Animator Controller 窗口上的空白区域。 1.从显示的上下文菜单中，选择 **Create State > From New Blend Tree**。 1.双击混合树 (Blend Tree) 以进入混合树视图 (Blend Tree Graph)。

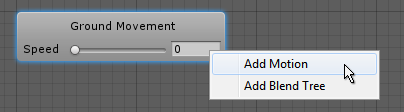
Animator 窗口现在显示整个混合树的图示，而 Inspector 显示当前选定节点及其直接子节点。

Animator 窗口显示整个混合树的图示。左边的混合树只有根混合节点（尚未添加子节点）。右边的混合树有一个根混合节点以及三个动画剪辑作为子节点。

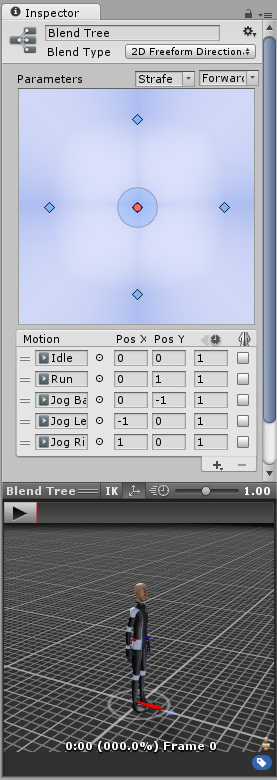
要将动画剪辑添加到混合树，可选择该混合树，然后单击 Inspector 的 Motion 字段内的加号图标。

添加任何运动之前在 Inspector 中显示的混合节点。加号图标用于添加动画剪辑或子混合树。

或者，也可通过右键单击混合树并从上下文菜单进行选择来添加动画剪辑或子混合节点：

右键单击混合树节点时的上下文菜单。

使用动画剪辑和输入参数来设置混合树时，Inspector 窗口通过图形显示动画如何随着参数值变化而进行组合（拖动滑动条时，树根中的箭头会改变其着色以显示其控制的动画剪辑）。

以五个动画剪辑设置的一个 2D 混合树在 Inspector 中预览的情况

您可以选择混合树图中的任何节点，然后在 Inspector 中检查它。如果所选节点是动画剪辑，则会显示该动画剪辑的 Inspector。如果该动画是从模型导入的，设置将处于只读状态。如果节点是混合节点，则会显示混合节点的 Inspector。

您可以从 *Blend Type* 菜单选择 [1D](file:///E:\UnityDocumentation\Manual\BlendTree-1DBlending.html) 或 [2D](file:///E:\UnityDocumentation\Manual\BlendTree-2DBlending.html) 混合；这两个类型之间的差异在本部分的相应页面中有介绍。

**混合树和根运动**

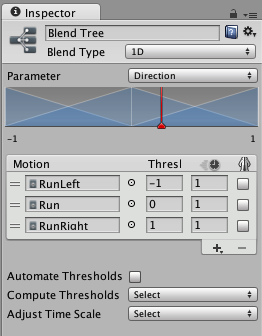
动画之间的混合是使用线性插值处理的（即，每个动画的量是由混合参数加权的单独动画的平均值）。但应注意，根运动\_不是\_以相同方式插值的。请参阅有关[根运动](file:///E:\\UnityDocumentation\\Manual\\RootMotion.html)的页面，详细了解这种情况可能给角色带来何种影响。

# 1D 混合

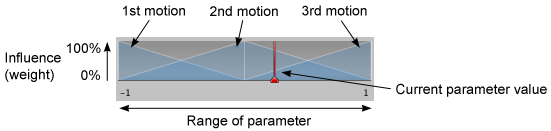
**混合节点\_\_的**Inspector\_\_ 中的第一个选项是 **Blend Type**。此下拉选单用于选择不同混合类型之一；这些类型可以根据一个或两个参数进行混合。1D 混合根据单个参数来混合子运动。

设置混合类型后，首先需要选择[动画参数](file:///E:\UnityDocumentation\Manual\AnimationParameters.html)来控制此混合树。在此示例中，\_Direction\_ 参数在 –1.0（左）和 +1.0（右）之间变化，其中 0.0 表示无倾斜的直线奔跑。

然后，您可以通过单击“+”小按钮并从弹出菜单中选择 **Add Motion Field** 来添加单个动画。操作完成后，应如下所示：

具有三个动画剪辑的 1D 混合树。

参数在最小值和最大值之间变化时，Inspector 顶部的图显示每个子运动的影响。每个运动都显示为一个蓝色小金字塔（第一个和最后一个仅显示一半），如果您在其中一个上单击并按住鼠标左键，相应运动将在下面的运动列表中突出显示。每个金字塔的顶峰定义运动具有最大影响的参数值，表示其动画权重为 1，而其他动画的权重为 0。这也称为运动的\_\_阈值\_\_。

混合树 Inspector 顶部的图显示了子运动在参数值范围内的权重。

红色竖条表示\_\_参数 (Parameter)\_\_ 的值。如果在 **Inspector** 底部的 **Preview** 中按 **Play** 并左右拖动图中的红色条，可以看到参数值如何控制不同运动的混合。

## 参数范围

混合树使用的参数范围在该图下方显示为左右侧的两个数字。可通过单击数字并用鼠标向左或向右拖动来更改其中任意一个数字。请注意，这些值对应于运动列表中第一个和最后一个运动的阈值。

## 阈值

您可以通过单击图中的对应蓝色金字塔并将其向左或向右拖动来更改运动的阈值。如果未启用“Automate Thresholds”开关，还可通过在 Threshold 列的数字字段中输入数字来编辑运动列表中某个运动的阈值。

运动列表下方是 Automate Thresholds 复选框。启用此复选框将在参数范围内均匀分布运动的阈值。例如，如果有五个剪辑且参数范围为 –90 到 +90，则阈值将按顺序设置为 –90、–45、0、+45 和 +90。

**Compute Thresholds** 下拉选单将根据您选择的数据设置阈值；这些数据是从动画剪辑的根运动中获得的。可供选择的数据是速度 (Speed)、速度 (Velocity) x、y 或 z 以及以角度或弧度表示的角速度 (Angular Speed)。如果您的\_\_参数\_\_对应于这些属性之一，则可使用 Compute Thresholds 下拉选单来计算阈值。

| **属性：** | **功能：** |
| --- | --- |
| **Speed** | 根据速度（速度的量级）设置每个运动的阈值。 |
| **Velocity X** | 根据 velocity.x 设置每个运动的阈值。 |
| **Velocity Y** | 根据 velocity.y 设置每个运动的阈值。 |
| **Velocity Z** | 根据 velocity.z 设置每个运动的阈值。 |
| **Angular Speed (Rad)** | 根据角速度（弧度/秒）设置每个运动的阈值。 |
| **Angular Speed (Deg)** | 根据角速度（角度/秒）设置每个运动的阈值。 |

例如，假设您有一段速度为每秒 1.5 个单位的行走动画、一段速度为每秒 2.3 个单位的慢跑动画以及一段速度为每秒 4 个单位的奔跑动画，则从下拉选单中选择 **Speed** 选项将基于这些值为三段动画设置参数范围和阈值。所以，如果将速度参数设置为 3.0，则会混合慢跑与奔跑，并略微偏向慢跑。

# 2D 混合

**混合节点\_\_的**Inspector\_\_ 中的第一个选项是 **Blend Type**。此下拉选单用于选择不同混合类型之一；这些类型可以根据一个或两个参数进行混合。2D 混合类型根据两个参数来混合子运动。

不同的 2D 混合类型具有适用的不同用途。它们的不同之处在于如何计算对每个运动的影响。

\_\_2D Simple Directional\_\_：最好在运动表示不同方向（例如“向前走”、“向后退”、“向左走”和“向右走”或者“向上瞄准”、“向下瞄准”、“向左瞄准”和“向右瞄准”）时使用。根据需要可以包括位置 (0, 0) 处的单个运动，例如“空闲”或“瞄准”。在 Simple Directional 类型中，在同一方向上\_不\_应该有多个运动，例如“向前走”和“向前跑”。

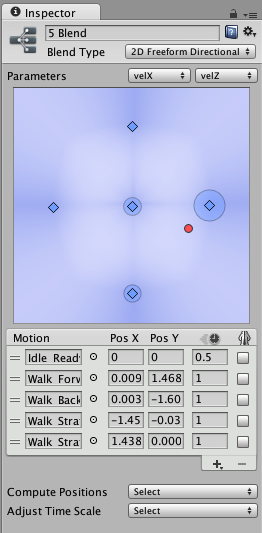
\_\_2D Freeform Directional\_\_：运动表示不同方向时，也使用此混合类型，但是您可以在同一方向上有多个运动，例如“向前走”和“向前跑”。在 Freeform Directional 类型中，运动集应始终包括位置 (0, 0) 处的单个运动，例如“空闲”。

\_\_2D Freeform Cartesian\_\_：最好在运动不表示不同方向时使用。凭借 Freeform Cartesian，X 参数和 Y 参数可以表示不同概念，例如角速度和线速度。一个示例是诸如“向前走不转弯”、“向前跑不转弯”、“向前走右转”、“向前跑右转”之类的运动。

\_\_Direct\_\_：此类型的混合树让用户直接控制每个节点的权重。适用于面部形状或随机空闲混合。

设置混合类型后，首先需要选择两个[动画参数](file:///E:\UnityDocumentation\Manual\AnimationParameters.html)来控制此[混合树](file:///E:\UnityDocumentation\Manual\class-BlendTree.html)。在此示例中，参数是 velocityX\_（扫射）和 velocityZ\_（前进速度）。

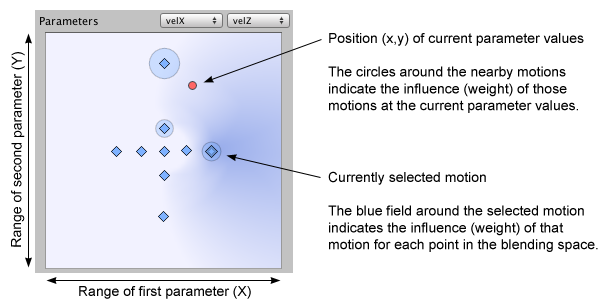
然后，您可以通过单击 **+ > Add Motion** 字段并将动画剪辑添加到混合树来添加单个动画。操作完成后，应如下所示：

具有五个动画剪辑的一个 2D 混合节点。

2D 混合中的位置类似于 1D 混合中的阈值，不同之处是有两个值而不是一个值，对应于两个参数。它们沿水平 X 轴的位置对应于第一个参数，沿垂直 Y 轴的位置对应于第二个参数。向前走动画可能的 velocityX 为 0 而 velocityZ 为 1.5，因此应将这些值输入该运动的 Pos X 和 Pos Y 数字字段中。

## 2D 混合图

Inspector 顶部的图显示了 2D 混合空间中的子运动的位置。这些运动以蓝点显示。未分配动画剪辑或混合树的运动对混合没有影响，并显示为灰点。您可以通过单击图中的运动点来选择该运动。选择运动后，该运动对混合空间中每个点的影响便会可视化为蓝色区域。此区域位于该运动的位置正下方时最强，此情况下运动具有完全影响，意味着其动画权重为 1 而其他动画的权重为 0。逐渐远离该位置后，此运动的影响将减弱，而其他运动的影响将增强。

混合节点 Inspector 顶部的图显示了子运动在参数值范围内的权重。

红点表示两个\_\_参数\_\_的值。如果在 **Inspector** 底部的 **Preview** 中按 **Play** 并在图中四处拖动红点，可看到参数值如何控制不同运动的混合。在图中，您还可以看到每个运动的影响（表示为围绕每个运动的圆圈）。您将看到，如果将红点移动到表示运动的其中一个蓝点上，则该运动的圆圈将获得其最大半径，而其他所有运动的圆圈将缩小到看不见。在几个运动之间的位置，附近的多个运动将对混合产生影响。如果选择其中一个运动以查看该运动的蓝色影响区域，您可以看到在四处移动红点时，运动的圆形大小与影响区域在不同位置的强度完全对应。

未选择任何运动时，该图显示所有影响区域的混合，在单个运动主导的位置，颜色会更蓝，而在许多运动对混合有影响的位置，蓝色会更淡。

## 位置

您可以通过单击图中的对应蓝点并将其四处拖动来更改运动的位置。此外，还可以通过在 Pos X 和 Pos Y 列的数字字段中输入数字来编辑运动列表中某个运动的位置坐标。

**Compute Positions** 下拉选单将根据您选择的数据设置位置（这些数据是从动画剪辑的根运动中获得的）。可供选择的数据是速度 (Speed)、速度 (Velocity) x、y 或 z 以及以角度或弧度表示的角速度 (Angular Speed)。如果您的一个或两个\_\_参数\_\_对应于其中一个属性，可以使用 Compute Positions 下拉选单来计算 Pos X 和/或 Pos Y。

| **属性：** | **功能：** |
| --- | --- |
| **Velocity XZ** | 根据运动的 velocity.x 设置每个运动的 Pos X，根据 velocity.z 设置 Pos Y。 |
| **Speed And Angular Speed** | 根据运动的角速度（弧度/秒）设置每个运动的 Pos X，根据速度设置 Pos Y。 |

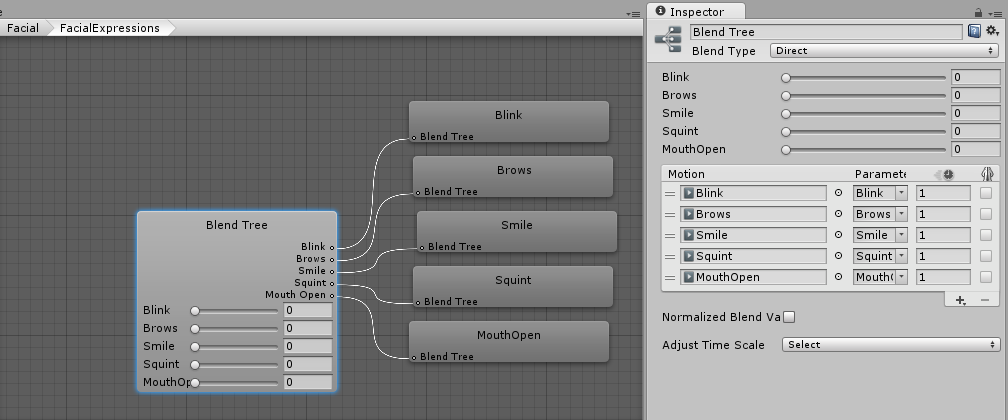
此外，可通过选择 **Compute Position > X Position From** 和/或 **Compute Position > Y Position From** 来一次仅自动计算其中一项而使其他项保持不变，从而进行混合与匹配。

| **属性：** | **功能：** |
| --- | --- |
| **Speed** | 根据速度（速度的量级）设置每个运动的 Pos X 或 Pos Y。 |
| **Velocity X** | 根据运动的 velocity.x 设置每个运动的 Pos X 或 Pos Y。 |
| **Velocity Y** | 根据运动的 velocity.y 设置每个运动的 Pos X 或 Pos Y。 |
| **Velocity Z** | 根据运动的 velocity.z 设置每个运动的 Pos X 或 Pos Y。 |
| **Angular Speed (Rad)** | 根据角速度（弧度/秒）设置每个运动的 Pos X 或 Pos Y。 |
| **Angular Speed (Deg)** | 根据角速度（度/秒）设置每个运动的 Pos X 或 Pos Y。 |

例如，假设参数对应于横向速度和前进速度，并有一段平均速度为 (0, 0, 0) 的空闲动画、一段速度为 (0, 0, 1.5) 的行走动画以及两段速度分别为 (–1.5, 0, 0) 和 (1.5, 0, 0) 的扫射动画。从下拉选单中选择 Velocity XZ 选项将根据这些速度的 X 和 Z 坐标来设置运动的位置。

# 直接混合

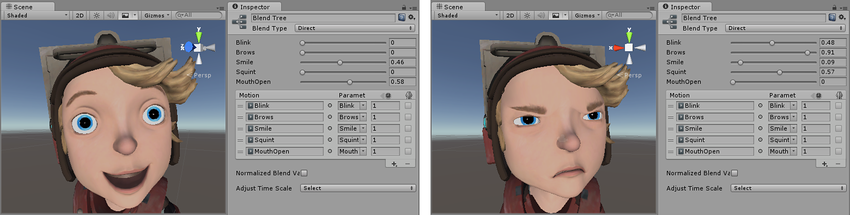
使用直接混合树可将 Animator 参数映射到 BlendTree 子项的权重。如果您想要精确控制所混合的各种动画而不是使用一个或两个参数来间接混合它们（1D 和 2D 混合树便是这种情况），这将非常有用。

一个已分配五个动画剪辑的直接混合树。

设置直接混合树时，检视面板允许您将动作添加到动作列表。然后，应为每个动作分配一个相应参数以直接控制其在树中的混合权重。[在此处阅读有关创建 Animator 参数的更多信息](file:///E:\UnityDocumentation\Manual\AnimationParameters.html)。

实际上，此直接模式只是绕过交叉渐变或各种 2D 混合算法（Freeform Directional、Freeform Cartesian 等），允许您实现所喜欢的任何代码来控制混合动画的混合。

在混合面部表情的混合形状动画时，或者在将附加动画混合在一起时，这可能特别有用。

每个剪辑的混合权重可以任意混合。

# 其他混合树选项

以下选项对于 1D 和 2D 混合都通用。

## 时间标度

您可以使用动画速度数字字段（顶部带有时钟图标的列）来更改动画剪辑的“自然”速度，因此可以使用值 2.0 来使行走速度翻倍。**Adjust Time Scale > Homogeneous Speed** 按钮可重新缩放剪辑的速度，使它们与所选的参数最小值和最大值相对应，但保持它们最初具有的相同\_相对\_速度。

请注意，\_\_Adjust Time Scale\_\_ 下拉选单仅在所有运动都是动画剪辑而非子混合树时才可用。

## 镜像

您可以通过启用最右侧的镜像切换开关来为运动列表中的任何人形动画剪辑生成镜像。此功能可让您使用原始形式和镜像版本的相同动画，但无需占用两倍的内存和空间。

# 动画混合形状

## 准备原始作品

在 Maya 中设置好混合形状后：

* 将选择的内容导出到 FBX，确保选中动画复选框并选中变形模型下的混合形状。
* 将 FBX 文件导入 Unity（Assets > Import New Assets > [文件名].fbx）。
* 将资源拖动到 Hierarchy 窗口中。如果在层级视图中选择对象并查看检视面板，则会在 SkinnedMeshRenderer 组件下看到列出的混合形状。在此位置，您可以调整混合形状对默认形状的影响，0 表示混合形状没有影响，100 表示混合形状具有最大影响。

## 在 Unity 中创建动画

也可以使用 Unity 中的 Animation 窗口来创建混合动画，步骤如下：

* 在 Window > Animation 下打开 Animation 窗口。
* 在窗口左侧，单击“Add Curve”，然后添加 Blend Shape，它将位于 Skinned Mesh Renderer 下。

从此处，您可以操作关键帧和混合权重来创建所需动画。

完成动画编辑后，可在 Editor 窗口或 Animation 窗口中单击 Play 以预览动画。

## 脚本访问

也可以使用 GetBlendShapeWeight 和 SetBlendShapeWeight 等函数来通过代码设置混合权重。

您还可以通过访问 blendShapeCount 变量以及其他有用的函数来检查 Mesh 上有多少混合形状。

以下代码示例连接到具有 3 个或更多混合形状的游戏对象时，随着时间推移，会将默认形状与其他两个混合形状混合在一起：



# Animator Override Controller

Animator Override Controller 是一种资源，它可让您扩展现有 Animator Controller，从而替换使用的特定动画但保留其原始结构、参数和逻辑。

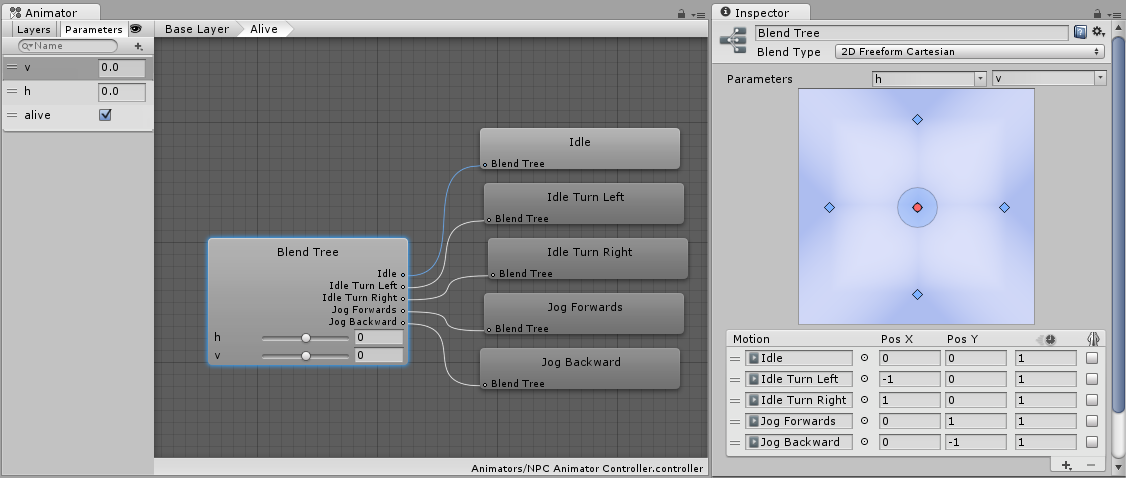
因此，您可以创建同一个基本状态机的多个变体，但是每个变体使用不同的动画集。例如，您的游戏可能有各种各样的 NPC 类角色生活在世界上，但每种类型（小妖精、恶魔、小精灵等）都有自己独特的行走、空闲、坐定等动画。

通过创建一个包含所有 NPC 类型角色逻辑的“基础”Animator Controller，即可为每种角色类型创建一个重写，并放入各自的动画文件。

以下是一个用于演示的典型Animator Controller资源：



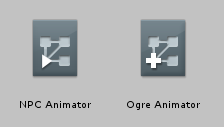
这代表一个Animator Controller，其中包含一个简单状态机，由混合树控制四个方向的动画，还有一个空闲动画，如下所示：



为了扩展此通用 NPC 状态机来使用仅适用于（比如）恶魔型角色的独特动画，您可以创建一个Animator Override Controller，并放入恶魔的动画剪辑以替换原始动画剪辑。恶魔可能有不同的空闲和四处游走方式，可能采取更缓慢、更狂暴和更有力的动作。但是，通过使用Animator Override Controller，如何在运动状态之间进行过渡和混合的基本逻辑可被具有不同动画集的不同角色所共享，从而减少构建和修改状态机本身所需的工作量。

要创建新的Animator Override Controller，请使用 **Assets > Create** 菜单，或 Project 视图中的 **Create** 按钮，然后选择 Animator Override Controller。

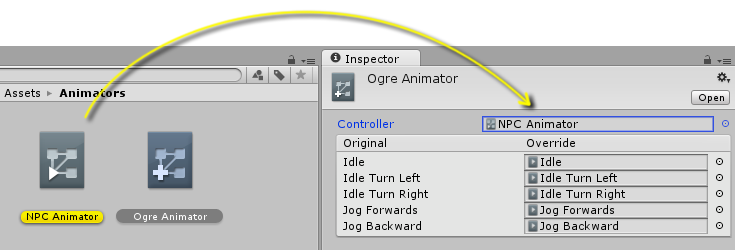
Animator Override Controller具有与Animator Controller非常相似的图标，不同之处在于它在图标角上带有“加号”而不是“播放”符号：

比较图标：并排展示 Animator Controller 和 Animator Override Controller 资源

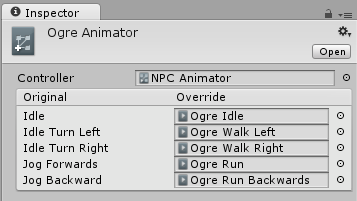
在检视面板中选择新的 Animator Override Controller 时，最初会处于未分配状态，如下所示：

Animator Override Controller 中未分配 Animator Controller

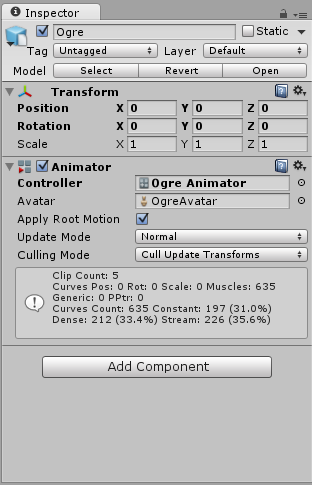
要开始使用Override Controller，必须将原始控制器资源分配给检视面板中的新Override Controller。完成此操作后，原始控制器中使用的所有动画将在 Override Controller 的检视面板中显示为列表：

将现有控制器拖入 Animator Override Controller 的检视面板

然后，您可以分配新动画剪辑以重写原始剪辑。在此示例中，已使用动画的“恶魔”(Ogre) 版本重写所有剪辑。

已分配新剪辑的 Override Controller

现在，此 Override Controller 可在Ogre角色的游戏对象的 Animator 组件中使用，就像它是 Animator Controller 一样。它将使用与原始 Animator Controller 相同的逻辑，但播放的是分配的新动画而不是原始动画。

在Animator组件中将Override Controller用于游戏对象