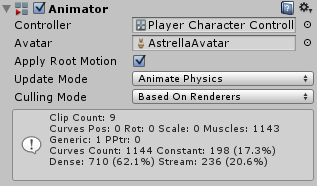
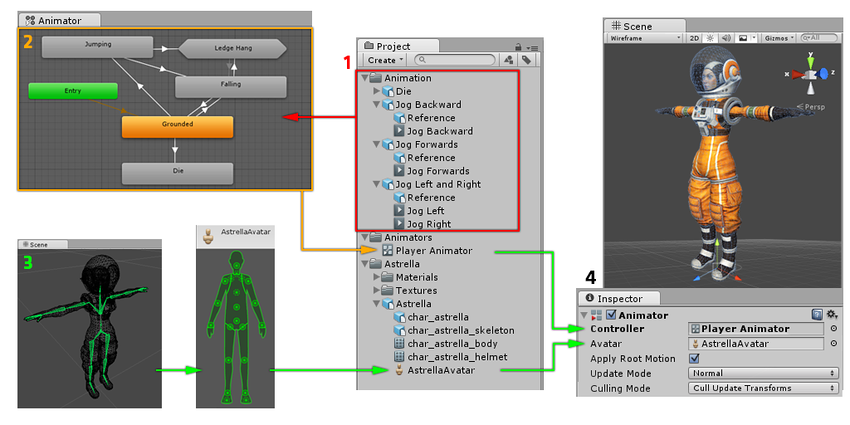
**Animator 组件 (Animator Component)**

Animator 组件用于将动画分配给场景中的游戏对象。Animator 组件需要引用 [Animator Controller](file:///E:\UnityDocumentation\Manual\class-AnimatorController.html)，后者定义要使用哪些动画剪辑，并控制何时以及如何在动画剪辑之间进行混合和过渡。

如果游戏对象是具有[Avatar](file:///E:\UnityDocumentation\Manual\class-Avatar.html)定义的人形角色，还应在此组件中分配Avatar，如下所示：

已分配控制器和 Avatar 的 Animator 组件。

下图显示了如何将各种资源（动画剪辑、Animator Controller 和 Avatar）一起汇集在游戏对象的 Animator 组件中：

此图显示了动画系统的各个部分如何连接在一起

## 属性

| **属性：** | | **功能：** |
| --- | --- | --- |
| **Controller** | | 附加到此角色的 Animator Controller。 |
| **Avatar** | | 此角色的[Avatar](file:///E:\UnityDocumentation\Manual\class-Avatar.html)。（如果 Animator 用于对人形角色进行动画化） |
| **Apply Root Motion** | | 我们应该从动画本身还是从脚本控制角色的位置和旋转。 |
| **Update Mode** | | 此选项允许您选择 Animator 何时更新以及应使用哪个时间标度。 |
|  | **Normal** | Animator 与 Update 调用同步更新，Animator 的速度与当前时间标度匹配。如果时间标度变慢，动画将通过减速来匹配。 |
|  | **Animate Physics** | Animator 与 FixedUpdate 调用同步更新（即，与物理系统步调一致）。如果要对具有物理交互的对象（例如可四处推动刚体对象的角色）的运动进行动画化，应使用此模式。 |
|  | **Unscaled Time** | Animator 与 Update 调用同步更新，但是 Animator 的速度忽略当前时间标度而不顾一切以 100% 速度进行动画化。此选项可用于以正常速度对 GUI 系统进行动画化，同时将修改的时间标度用于特效或暂停游戏。 |
| **Culling Mode** | | 您可以为动画选择的剔除模式。 |
|  | **Always Animate** | 始终进行动画化，即使在屏幕外也不要剔除。 |
|  | **Cull Update Transforms** | 未显示渲染器时，禁用变换组件的重定向、IK（反向动力学）和写入。 |
|  | **Cull Completely** | 未显示渲染器时，完全禁用动画。 |

## 动画曲线信息

Animator 组件底部的信息框为您提供 Animator Controller 使用的所有剪辑中所用数据的明细。

动画剪辑包含“曲线”形式的数据；曲线表示值如何随时间变化。这些曲线可描述对象的位置或旋转、人形动画系统中肌肉的弯曲或者剪辑内的其他动画值（例如改变的材质颜色）。

下表说明了每个数据项代表的内容：

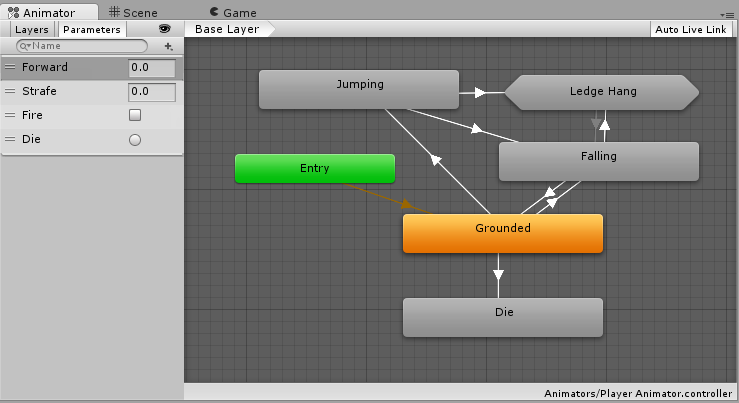
| **标签** | **描述** |
| --- | --- |
| **Clip Count** | 分配给此 Animator 的 Animator Controller 使用的动画剪辑总数。 |
| **Curves (Pos, Rot & Scale)** | 用于动画化对象位置、旋转或缩放的曲线总数。这些曲线用于不属于标准人形骨架的动画对象。在对人形 Avatar 进行动画化时，这些曲线会显示额外非肌肉骨骼（例如尾巴、飘逸的布料或悬垂的吊坠）的计数。如果您有人形动画并发现意外的非肌肉动画曲线，表示动画文件中可能有不必要的动画曲线。 |
| **Muscles** | 此 Animator 用于人形动画的肌肉动画曲线数量。这些是用于对标准人形 Avatar 肌肉进行动画化的曲线。除了 Unity 的标准 Avatar 中所有人形骨骼的标准肌肉运动外，此处还包括用于存储根运动位置和旋转动画的两条“肌肉曲线”。 |
| **Generic** | 由 Animator 用于动画化其他属性（如材质颜色）的数字（浮点）曲线数量。 |
| **PPtr** | 精灵动画曲线（由 Unity 的 2D 系统使用）的总数 |
| **Curves Count** | 动画曲线的合计总数 |
| **Constant** | 优化为常量（不变）值的动画曲线数量。如果动画文件包含了具有不变值的曲线，Unity 会自动选择此项。 |
| **Dense** | 使用“密集”数据（通过线性内插的离散值）存储方法进行优化的动画曲线数量。与“流”方法相比，此方法使用的内存少得多。 |
| **Stream** | 使用“流”数据（这些值具有用于曲线插值的时间和切线数据）存储方法的动画曲线数量。与“密集”方法相比，此数据占用的内存多得多。 |

如果导入动画剪辑时在[动画导入引用](file:///E:\\UnityDocumentation\\Manual\\class-AnimationClip.html)中将“Anim Compression”设置为“Optimal”，Unity 将使用启发式算法来确定最好使用密集还是流方法来存储每条曲线的数据。

# Animator Controller

Animator Controller 允许您为角色或对象安排和维护一组动画剪辑以及关联的动画过渡。在大多数情况下，拥有多个动画并在满足某些游戏条件时在这些动画之间切换是很常见的。例如，只要按下空格键，就可以从行走动画剪辑切换到跳跃动画剪辑。但是，即使您仅有一个动画剪辑，仍需要将其放入 Animator Controller 以便将其用于游戏对象。

Animator Controller 引用其中所用的动画剪辑，使用\_\_状态机\_\_来管理各种动画剪辑和它们之间的过渡；状态机可视为动画剪辑和过渡的一种流程图，或者是在 Unity 中使用可视化编程语言编写的简单程序。可在[此处](file:///E:\UnityDocumentation\Manual\AnimationStateMachines.html)找到有关状态机的更多信息。

简单的Animator Controller

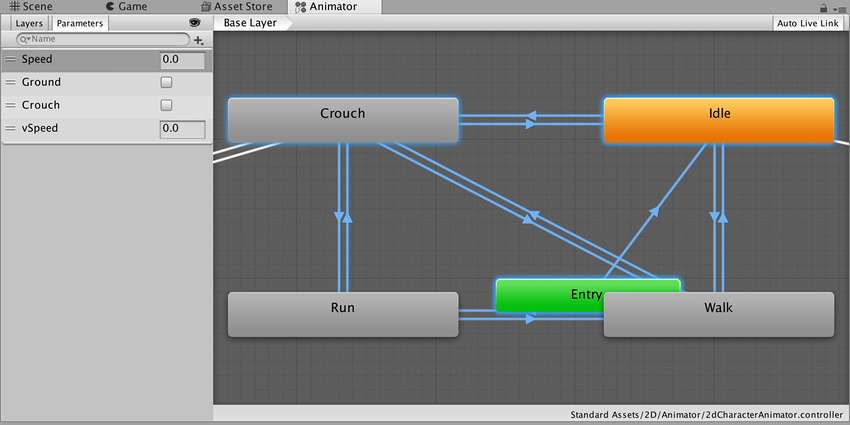
开始使用 Animation 窗口对游戏对象进行动画化时，或者将动画剪辑附加到游戏对象时，Unity 会自动创建 Animator Controller。

要手动创建 Animator Controller，请右键单击 Project 窗口，然后单击 **Create > Animator Controller**。

## 导航

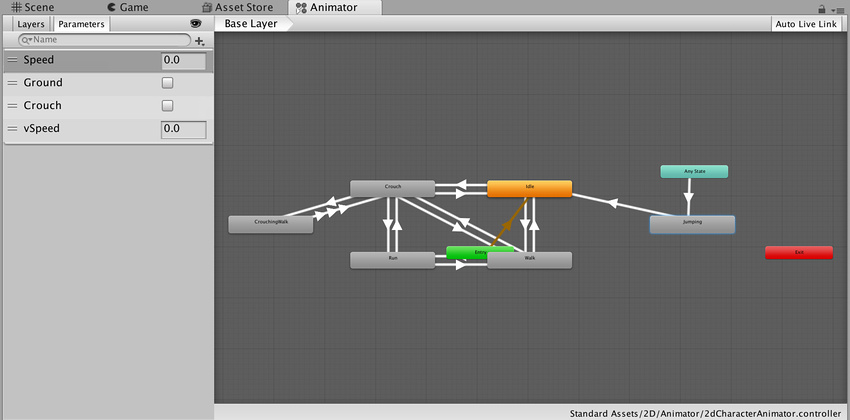
使用鼠标滚轮或类似功能可放大和缩小 Animator Controller 窗口。

要聚焦 Animator Controller 窗口中的某一项，请选择一个或多个状态（单击要选择的状态或在其周围拖动选框），然后按 F 键来放大所选项。

聚焦所选状态

按 A 键可将所有动画状态调整到 Animator Controller 视图中。

Unity 将保存您的选择。按 A 和 F 键可在所选动画状态与整个 Animator Controller 之间切换。

按 A 键时，Unity 将所有状态自动调整到 Animator Controller 视图中

在播放模式期间，Animator 会平移视图，以便始终能看到正在播放的当前状态。Animator Controller 遵循基础层和子状态机的独立缩放因子，而窗口将自动平移以确保活动状态的可见性。

要在播放模式下修改缩放，请遵循以下步骤：

* 在 Animator Controller 窗口中启用 **Auto Live Link**
* 单击 Play 按钮以进入播放模式
* 单击 Pause
* 在 Animator Controller 中，选择要放大的一个或多个状态
* 按 F 键来放大所选项
* 再次单击 Play 按钮以恢复播放模式

请注意，Animator Controller 在激活时会平移到每个状态。

Animator 平移到活动状态

# 创建AnimatorController

### Animator Controller

可从\_\_Animator Controller\_\_视图（菜单：\_\_Window > Animator Controller\_\_）中查看并设置角色行为。

可通过多种方式创建Animator Controller：

* 从 \_\_Project 视图\_\_中，选择“**Create > Animator Controller**”。
* 在 Project 视图中右键单击并选择“**Create > Animator Controller**”。
* 从 Assets 菜单中，选择“**Assets > Create > Animator Controller**”。

随后将在磁盘上创建 .controller 资源。在 **Project Browser** 窗口中，该图标如下所示：

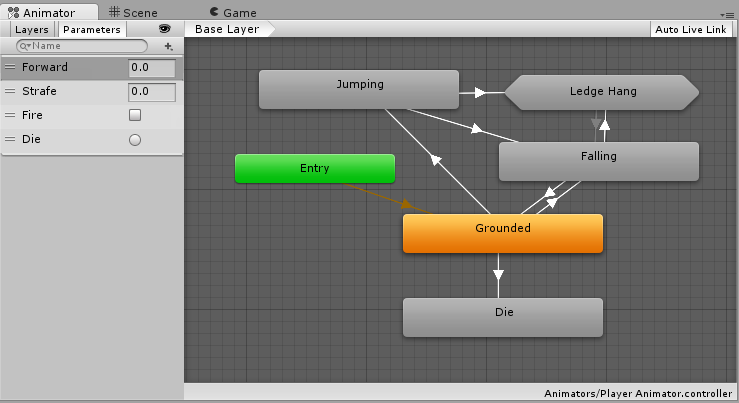
磁盘上的Animator Controller资源磁盘上的Animator Controller资源

### Animator 窗口

完成状态机设置后，您可以将控制器放在 \_\_Hierarchy 视图\_\_中具有 Avatar 的任何角色的 Animator 组件上。

Animator Controller 窗口包含：

* \_\_动画层小部件\_\_（左上角，请参阅[动画层](file:///E:\UnityDocumentation\Manual\AnimationLayers.html)）
* \_\_事件参数小部件\_\_（左上方，请参阅[动画参数](file:///E:\UnityDocumentation\Manual\AnimationParameters.html)）
* [状态机](file:///E:\UnityDocumentation\Manual\AnimationStateMachines.html)可视化。

Animator Controller 窗口

请注意，Animator Controller 窗口将始终显示最近选择的 .controller 资源的状态机（无论当前加载了什么场景）。

# 动画状态

\_\_动画状态\_\_是\_\_动画状态机\_\_的基本组成模块。每个状态包含一个单独动画序列（或[混合树](file:///E:\UnityDocumentation\Manual\class-BlendTree.html)）；此序列将在角色处于该状态时播放。游戏中的事件触发状态转换时，角色将处于新状态，随后会替换其动画序列。

在 **Animator Controller** 中选择某个状态时，Inspector 中将显示该状态的属性：



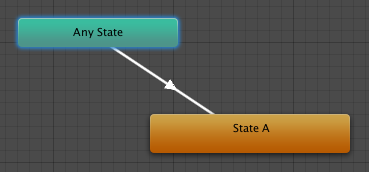
| **属性：** | **功能：** |
| --- | --- |
| **Speed** | 动画的默认速度 |
| **Motion** | 分配给此状态的动画剪辑 |
| **Foot IK** | 此状态下是否遵循 Foot IK。适用于人形动画。 |
| **Write Defaults** | AnimatorStates 是否为其运动执行未动画化的属性写回默认值。 |
| **Mirror** | 是否为状态生成镜像。这仅适用于人形动画。 |
| **Transitions** | 源自此状态的过渡的列表 |

以棕色显示的默认状态是状态机首次激活时所处的状态。必要时，可通过右键单击另一状态并从上下文菜单选择 **Set As Default** 来更改默认状态。每个过渡上的 Solo 和 Mute 复选框用于控制\_\_动画预览\_\_的行为，请参阅[本页面](file:///E:\UnityDocumentation\Manual\AnimationSoloMute.html)以了解更多详细信息。

可通过右键单击 **Animator Controller 窗口\_\_中的空白区域并从上下文菜单选择**Create State > Empty\_\_ 来添加新状态。或者，也可将动画拖到 Animator Controller 窗口中，从而创建包含该动画的状态。（请注意，只能将 Mecanim 动画拖到控制器中，非 Mecanim 动画将被拒绝。）状态还可包含[混合树](file:///E:\UnityDocumentation\Manual\class-BlendTree.html)。

### 任意状态

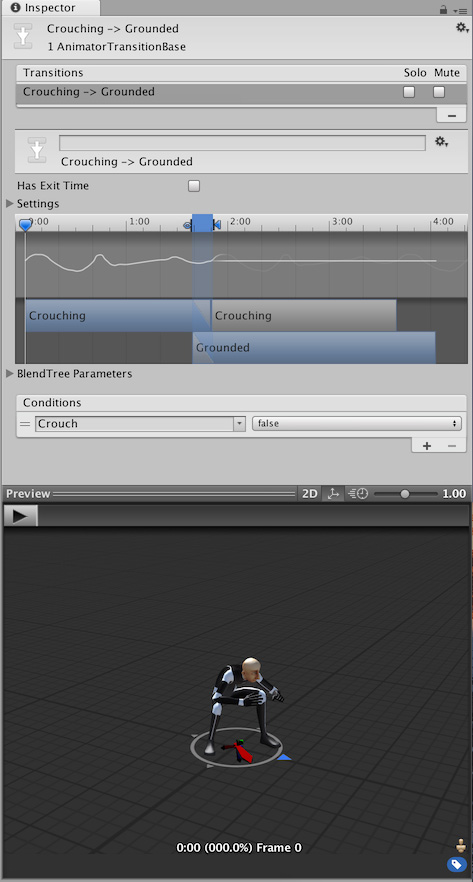
**任意状态 (Any State)** 是一个始终存在的特殊状态。此状态适用于您想要进入特定状态的情况（无论您当前处于哪个状态）。这是向状态机中的所有状态添加相同外向过渡的简便方法。请注意，\_\_任意状态\_\_的特殊含义意味着它不能成为过渡的端点（即，不能通过跳到“任意状态”选择下一步要进入的随机状态）。



# 动画过渡

动画过渡允许[状态机](file:///E:\UnityDocumentation\Manual\StateMachineBasics.html)从一个动画状态切换或混合到另一动画状态。过渡不仅定义状态之间的混合应该耗费多长时间，而且还定义它们应该在什么条件下激活。您可以设置仅在特定条件成立时才发生过渡。要设置这些条件，请在 Animator Controller 中指定参数值。

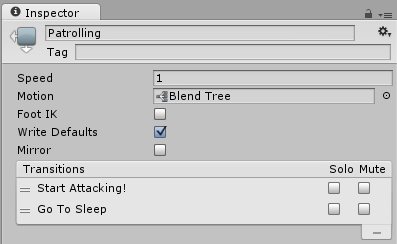
例如，您的角色可能具有“巡逻”状态和“睡眠”状态。您可以将巡逻和睡眠之间的过渡设置为仅在“alertness”参数值低于某个水平时才会发生。

在 Inspector中查看过渡的示例。

要为过渡命名，请在字段中输入名称，如下所示：

E:\UnityDocumentation\uploads\Main\AnimatorTransitionName.png

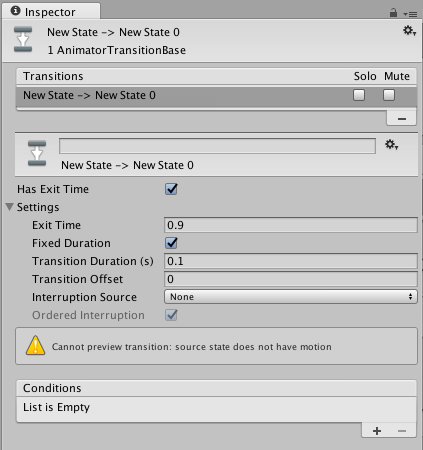
状态的 Inspector 窗口会显示该状态使用的过渡，如下所示：



在任何给定时间只能有一个激活的过渡。但是，当前激活的过渡可能会被另一个过渡中断（如果已将设置配置为允许此行为，请参阅以下的[过渡中断](file:///E:\UnityDocumentation\Manual\class-Transition.html#TransitionInterruption)）。

### 过渡属性

要查看某个过渡的属性，请单击 Animator 窗口中连接两个状态的过渡线。这些属性将显示在 Inspector 窗口中。



使用以下属性可调整过渡及其在当前状态和下一状态之间如何混合。

| **属性** | **功能** |
| --- | --- |
| **Has Exit Time** | **Exit Time** 是一种不依赖参数的特殊过渡。但是，它依赖状态的标准化时间。选中此选项可在 **Exit Time** 指定的具体时间进行过渡。 |
| **Settings** | 包含详细过渡设置的折叠菜单（如下所示）。 |
| **Exit Time** | 如果选中 **Has Exit Time\_\_，此值表示过渡可以生效的确切时间。该时间以标准化时间表示（例如，退出时间为 0.75 表示，在已播放 75% 动画的第一帧上，\_\_Exit Time** 条件为 true）。在下一帧上，该条件为 false。  对于循环动画，每个循环都会评估退出时间小于 1 的过渡，因此可使用此选项在每个循环的动画中以适当时机对过渡进行计时。  对 **Exit Time** 大于 1 的过渡仅评估一次，因此此类过渡可用于在固定循环次数后的特定时间退出。例如，过渡的退出时间为 3.5 的情况下，在三个半循环后对该过渡评估一次。 |
| **Fixed Duration** | 如果选中 **Fixed Duration** 复选框，则以秒为单位解读过渡时间。如果未选中 **Fixed Duration** 复选框，则过渡时间解读为源状态的标准化时间的一部分。 |
| **Transition Duration** | 相对于当前状态持续时间的过渡持续时间，以标准化时间或秒为单位（具体取决于 **Fixed Duration** 模式）。此时间在过渡图中显示为两个蓝色标记之间的部分。 |
| **Transition Offset** | 过渡到的目标状态的起始播放的时间偏移。例如，值为 0.5 表示目标状态在其时间轴的 50% 处开始播放。 |
| **Interruption Source** | 此选项用于控制该过渡可能中断的情况（请参阅下文的[过渡中断](file:///E:\UnityDocumentation\Manual\class-Transition.html#TransitionInterruption)）。 |
| **Ordered Interruption** | 确定当前过渡是否可在不考虑顺序的情况下被其他过渡中断（请参阅下文的[过渡中断](file:///E:\UnityDocumentation\Manual\class-Transition.html#TransitionInterruption)）。 |
| **Conditions** | 一个过渡可以具有单个条件、多个条件或根本没有条件。如果过渡没有条件，Unity Editor 只会考虑 **Exit Time\_\_，并在达到退出时间时发生过渡。如果过渡有一个或多个条件，则必须满足所有条件才能触发过渡。  一个条件包含：  - 一个事件参数（在条件中考虑其值）。 - 条件谓词（需要时填写，例如，浮点数的“小于”或“大于”）。 - 一个参数值（需要时填写）。  如果为该过渡选择**Has Exit Time\_\_ 并有一个或多个条件，请注意 Unity Editor 在 **Exit Time** 之后考虑条件是否为 true。这样可确保在动画的特定部分中进行过渡。 |

### 过渡中断

使用 **Interruption Source** 和 **Ordered Interruption** 属性可控制如何中断过渡。

在概念上，中断顺序就像这些过渡已排队，然后解析为一个有效过渡从第一个过渡插入到最后一个过渡。

#### Interruption Source 属性

[AnyState](file:///E:\UnityDocumentation\Manual\class-State.html) 中的过渡始终首先添加在队列中，然后其他过渡根据 **Interruption Source** 的值排队：

| **值** | **功能** |
| --- | --- |
| **None** | 不再添加任何过渡。 |
| **Current State** | 将当前状态的过渡排队。 |
| **Next State** | 使下一状态的过渡进行排队。 |
| **Current State then Next State** | 将当前状态的过渡排序，然后将下一状态的过渡排序。 |
| **Next State then Current State** | 将下一状态的过渡排队，然后将当前状态的过渡排队。 |

**注意**：这意味着，即使在 **Interruption Source** 设置为 **None** 的情况下，过渡也可以被其中一个 [AnyState](file:///E:\\UnityDocumentation\\Manual\\class-State.html) 过渡中断。

#### Ordered Interruption 属性

**Ordered Interruption** 属性将更改队列的解析方式。

根据该属性的值，对队列的解析将在以下列出的不同时间结束。

| **值** | **结束时间** |
| --- | --- |
| **选中** | 已找到有效过渡或当前过渡。 |
| **取消选中** | 已找到有效过渡。 |

只有 [AnyState](file:///E:\\UnityDocumentation\\Manual\\class-State.html) 过渡才能被自身中断。

要了解有关过渡中断的更多信息，请参阅 Unity 博客文章[状态机过渡中断 (State Machine Transition Interruptions)](https://blogs.unity3d.com/2016/07/13/wait-ive-changed-my-mind-state-machine-transition-interruptions/)。

### 过渡图

要手动调整以上列出的设置，可直接在字段中输入数字或使用过渡图。在操作视图元素时，过渡图会修改上述值。

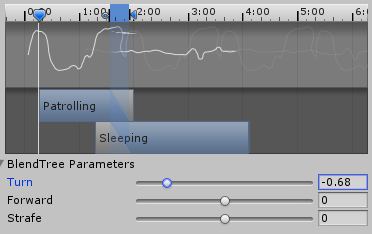
Inspector 中显示的过渡设置和过渡图

请根据以下说明更改图形视图中的过渡属性：

* 拖动 **Duration “out”** 标记可更改过渡的\_\_持续时间 (Duration)\_\_。
* 拖动 **Duration “in”** 标记可更改过渡的持续时间以及\_\_退出时间 (Exit Time)\_\_。
* 拖动目标状态可调整\_\_过渡偏移 (Transition Offset)\_\_。
* 拖动预览播放游标可在 Inspector 底部的预览窗口中快速预览动画混合（效果）。

### 混合树状态之间的过渡

如果属于此过渡的当前状态或下一状态是[混合树](file:///E:\UnityDocumentation\Manual\class-BlendTree.html)状态，则混合树参数将出现在 Inspector 中。通过调整这些值可预览在混合树值设置为不同配置时的过渡表现情况。 如果混合树包含不同长度的剪辑，您应该测试在显示短剪辑和长剪辑时的过渡表现。调整这些值不会影响过渡在运行时的行为方式；它们仅用于帮助您预览不同条件下的过渡表现情况。

当前状态或下一状态是混合树状态时显示的混合树参数预览控件。

### 条件

一个过渡可以具有单个条件、多个条件或根本没有条件。如果过渡没有条件，Unity Editor 只会考虑 \_\_Exit Time\_\_，并在达到退出时间时发生过渡。如果过渡有一个或多个条件，则必须满足所有条件才能触发过渡。

一个条件包含：

* 一个事件参数（在条件中考虑其值）。
* 条件谓词（需要时填写，例如，浮点数的小于或大于）。
* 一个参数值（需要时填写）。

如果为该过渡启用 **Has Exit Time** 并有一个或多个条件，仅在状态的退出时间之后才会检查这些条件。这样可确保仅在动画的特定部分中进行过渡。