

# 动画

---

本页介绍了如何在自上而下引擎中使用动画。

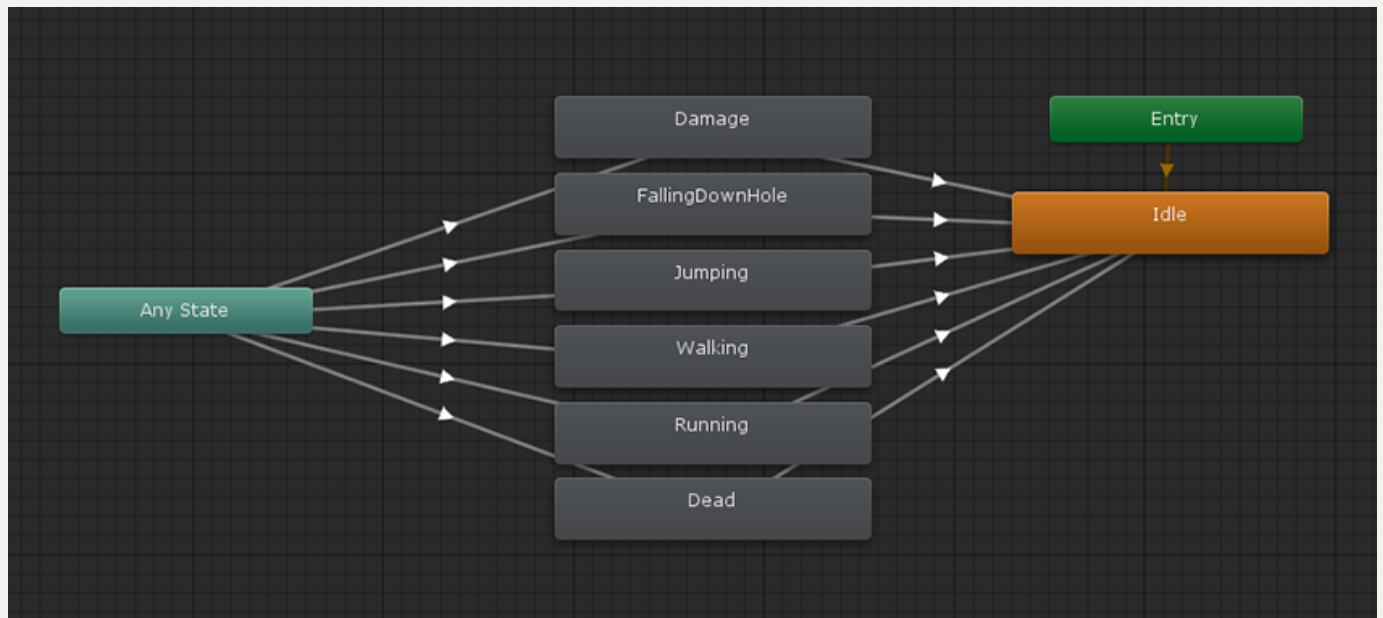
- [介绍](#)
- [动画控制器](#)
- [动画和脚本](#)
- [添加新动画](#)
- [动画参数](#)
- [二维细节](#)
- [调试动画](#)

## 介绍

自上而下引擎包括许多演示角色，它们都带有许多动画。在各种演示中，您会发现一些使用**spritesheets**制作动画的角色，一些使用**Mecanim**或**3D fbx** 动画模型。选择适合您的技能和需求的动画方法完全取决于您。但只要 Unity 支持，引擎应该涵盖您决定的任何方法。当涉及到动画时，它是不可知的，它的范围在发送和更新动画参数时结束。您决定如何使用它们取决于您，让您可以充分利用 Unity 动画系统的强大功能。

本页不会介绍如何创建动画或如何设置动画师。Unity 有很多关于这方面的文档，[去看看](#)吧。然而，它将涵盖自上而下引擎的细节以及它如何帮助您创建漂亮的动画角色。

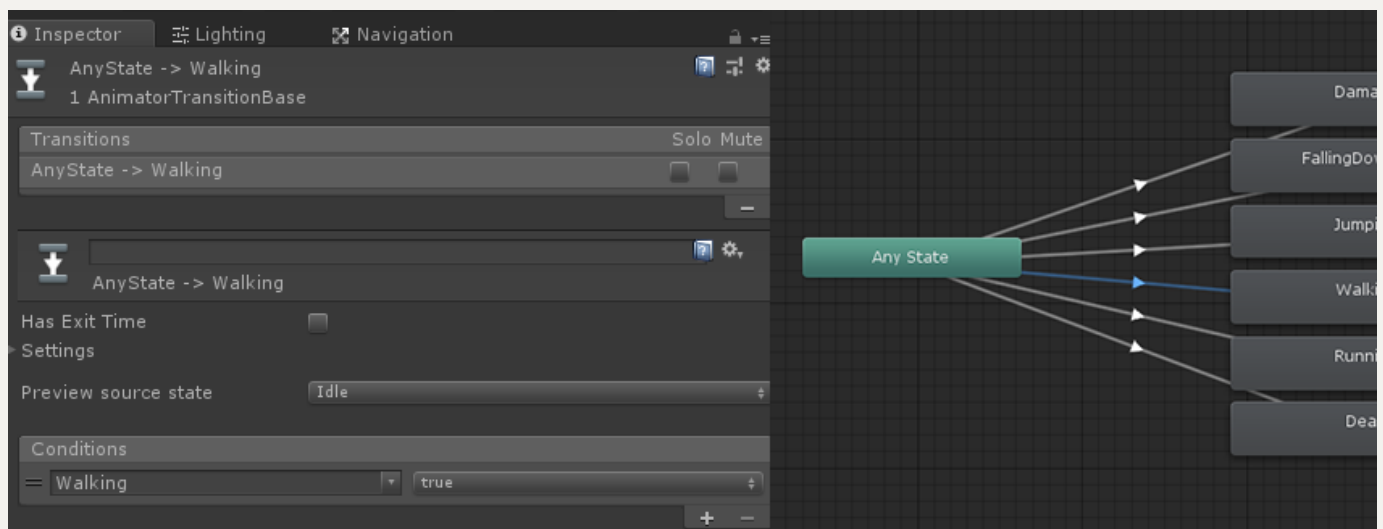
## 动画控制器



## 考拉角色动画师

在大多数情况下，您需要一个动画控制器来设置您的动画。资产包括一堆这些，我建议使用考拉作为您的起点，因为它包含大多数动画参数，因此您不必再次输入它们。您可以简单地复制它，然后将您的动画拖入其中，随时替换考拉的动画，并根据需要添加更多参数。

动画控制器由两大部分组成：一方面动画参数将通过角色和角色能力脚本每帧更新以反映角色的当前状态，另一方面是状态机，允许您确定每个动画应该在什么条件下播放以及如何从一个过渡到另一个。



## 过渡的例子

您会注意到演示动画师的工作流程非常简单，通常基于"任何状态"模式，适用于演示，但您可能希望针对每个特定角色进行更改。该引擎确实鼓励您创建自己的图形。每个角色都是独一无二的，并且在动画方面都有自己的需求，因此没有灵丹妙药，如果您想创建复杂的角色，您必须学习如何设置 Unity 动画过渡。您可以查看 Tie 动画器以了解更传统的动画器设置示例（与 Koala 的"任何状态"模式相反）。

或者，如果您对演示动画师设置之一感到满意，并且只想用您的动画替换动画，则可以使用[动画师覆盖控制器](#)（右键单击、创建、动画师覆盖控制器）。然后，从它的检查器中，选择您要覆盖的动画控制器，然后简单地将您的动画拖放到适当的插槽中。

## 动画和脚本

自上而下引擎的角色系统具有内置的动画界面，因此您不会在这方面浪费时间。每个能力都已经加载了相应的动画参数。要更新动画参数，您可以使用[Unity 的内置方法](#)或 TopDown 引擎提供的[方法](#)。这真的很简单，因为从任何能力中你只需要覆盖两个方法。让我们来看看它在 Dash 能力中是如何使用的：

首先，通常在能力类的顶部，我们为每个动画参数声明两个变量：

```
// animation parameters
protected const string _dashingAnimationParameterName = "Dashing";
protected int _dashingAnimationParameter;
```

**InitializeAnimatorParameters**：此方法"注册"参数，供以后使用。基本上它只是将该参数添加到一个列表中，在检查它在动画控制器中的存在之后，以避免在运行时出现潜在的错误。如果该参数不存在，更新请求将什么也不做，不会触发错误。这允许您在多个角色之间共享一个动画师，而无需将所有参数复制到所有控制器中。此方法仅在初始化时调用。它还负责将字符串散列为 int 以提高性能。

```
protected override void InitializeAnimatorParameters()
{
    RegisterAnimatorParameter(_dashingAnimationParameterName,
    AnimatorControllerParameterType.Bool, out
    _dashingAnimationParameter);
}
```

**UpdateAnimator**: 此方法每帧调用一次，将使用当前值更新动画器参数。在其中，您应该只调用 `MMAAnimator.UpdateAnimatorBool/Int/Trigger`。

```
public override void UpdateAnimator()
{
    MMAAnimatorExtensions.UpdateAnimatorBool(_animator,
        _dashingAnimationParameter, (_movement.CurrentState ==
        CharacterStates.MovementStates.Dashing),
        _character._animatorParameters);
}
```

## 添加新动画

要添加新动画，您要做的就是创建它，将其拖到角色的动画控制器中，然后创建过渡到它。如果它需要新的动画参数，请确保将它们都添加到动画控制器的参数列表中，并使用上述方法从现有或新功能在脚本中注册/更新它们。

## 动画参数

引擎定义了许多动画参数，供其各种能力和类使用。您的角色的动画师不需要所有这些，但拥有它们也无妨。您可以手动一个一个地添加它们，或者您可以向您的动画师添加一个 `CharacterAnimationParametersInitializer` 组件，然后按其上的 `AddAnimationParameters` 按钮，它会为您添加它们。完成此操作后，您可以安全地删除该组件。

这是引擎中已有的所有动画参数的完整列表：

参数名称	能力	类型	角色
活	特点	布尔值	如果角色当前还活着，则为真
接地	特点	布尔值	如果角色接触地面，则为真
极速	特点	漂	角色当前的x速度

		浮
y速度	特点	漂 角色当前的 y 速度 浮
速度	特点	漂 角色当前的z速度 浮
水平方向	CharacterOrientation2D	漂 字符在 x 轴上的方向 浮
垂直方向	CharacterOrientation2D	漂 字符在 y 轴上的方向 浮
相对前进速度	特点	漂 相对（相对于角色的前进）速度 浮
相对横向速度	特点	漂 相对（相对于角色的向前）横向速度 浮
相对前向速度归一化	特点	漂 相对前进速度，归一化 浮
相对横向速度归一化	特点	漂 相对横向速度，归一化 浮
RemappedForwardSpeedNormalized	角色定位3D	漂 在 0 和最大运行速度之间重新映射的前进速 浮 度值
重新映射的横向速度归一化	角色定位3D	漂 重新映射的横向速度值介于 0 和最大运行速 浮 度之间
Y旋转速度	角色定位3D	漂 模型的瞬时转速 浮
闲置的	特点	布 如果角色当前空闲，则为真 尔 值
激活	角色按钮激活	布 如果角色当前正在激活某些东西，则为真 尔 值
蹲伏	角色蹲伏	布 如果角色当前蹲下，则为真 尔 值
爬行	角色蹲伏	布 如果角色当前正在爬行，则为真 尔 值

伤害	健康	扳机	角色受到伤害时触发
潇洒	性格潇洒	布尔值	如果角色当前正在冲刺，则为真
短跑开始	性格潇洒	布尔值	如果角色在此帧开始破折号，则为真
死亡	健康	扳机	角色死亡时触发
方向	CharacterOrientation2D	漂浮	v1.3中引入，0：西，1：北，2：东，3：南
落下洞	CharacterFallDownHoles2D	布尔值	如果角色当前正在掉下一个洞，则为真
武器装备	角色手柄武器	布尔值	如果当前装备了武器，则为真
武器装备ID	角色手柄武器	整数	-1 如果没有装备武器，否则武器上指定的 WeaponAnimationID
跳跃	人物跳跃	布尔值	如果角色当前正在跳跃，则为真
击中地面	人物跳跃	布尔值	如果角色在这一帧刚刚落地，则为真
速度	人物动作	漂浮	角色当前的水平速度
步行	人物动作	布尔值	如果角色当前正在行走，则为真
跑步	角色奔跑	布尔值	如果角色当前正在运行，则为真

随机的	特点	漂 0f 和 1f 之间经常刷新的随机值，可用于为动画添加随机性
随机常数	特点	整 一个随机整数（介于 0 和 1000 之间），在开始生成，并且在此动画师的整个生命周期内保持不变，对于具有相同类型的不同字符很有用

除此之外，您还可以在 `CharacterHandleWeapon` 中找到许多动画参数，您可以直接从每个武器的检查器中设置它们的名称：

参数属性	能力	类型	角色
武器装备	角色 手柄 武器	布尔值	如果当前装备了武器，则为真
武器装备ID	角色 手柄 武器	整数	当前装备的武器ID
空闲动画参数	角色 手柄 武器	布尔值	如果武器空闲，则为真
开始动画参数	角色 手柄 武器	布尔值	如果武器正在启动，则为真
<code>DelayBeforeUseAnimationParameter</code>	角色 手柄 武器	布尔值	如果武器处于 <code>DelayBeforeUse</code> 状态，则为真
<code>DelayBetweenUsesAnimationParameter</code>	角色 手柄 武器	布尔值	如果在两次使用之间，则为真
停止动画参数	角色 手柄 武器	布尔值	当武器停止时为真
重新加载开始动画参数	角色 手柄 武器	布尔值	重新加载开始时为真

重新加载停止动画参数	角色 手柄 武器	布 尔 值	重新加载完成时为真
重载动画参数	角色 手柄 武器	布 尔 值	当武器重新加载时为真
单用途动画参数	角色 手柄 武器	布 尔 值	在使用武器的确切框架中为真（子弹被击中，剑击中 等）
使用动画参数	角色 手柄 武器	布 尔 值	当武器被积极使用时为真
装备动画参数	角色 手柄 武器	布 尔 值	装备武器时为真
组合进度	角色 手柄 武器	布 尔 值	如果正在使用组合武器，则 为真
武器角度动画参数	角色 手柄 武器	漂 浮	武器的当前角度（基于 WeaponAim）
<b>WeaponAngleRelativeAnimationParameter</b>	角色 手柄 武器	漂 浮	武器的当前角度，相对于角 色的面向方向（基于 WeaponAim）

## 二维细节

引擎中的动画对 2D 和 3D 的全局工作方式相同，但方向除外。在 3D 中，CharacterOrientation3D 能力将旋转您的角色模型。然而，在 2D 中，除了可选的翻转之外，如果您希望您的角色看起来像是在向上、向下、向左、向右等走动，您将需要动画。这是通过 CharacterOrientation2D 功能完成的，它提供了发送到的选项角色的动画师的水平和垂直方向。然后可以在混合树中使用它来定义应该播放什么动画。如果你想要一个例子，你可以查看 Grasslands 演示场景。该场景中的角色是 2D 精灵，带有助于空闲、行走和攻击的混合树，每个都根据角色的方向分为 4 个方向。您可以了解有关混合树的更多信息 [Unity 的文档](#)。



## 调试动画

如果您的动画没有触发，或者如果它们在您想要的时候没有触发，那么找出问题所在可能会有点令人困惑，尤其是当您是 Unity 动画师的新手时。如果是这种情况，您应该做的第一件事可能是阅读 Unity 在 Mecanim 上的资源，[例如这个](#)。

一旦您对基础知识有了更好的了解，您可以通过以下几种方法来查明问题：

您的角色根本没有动画 很可能您没有将动画师绑定到角色，因此处理角色逻辑的类和能力不知道要更新哪个动画师。选择您的 **Character**，打开它的 Character 检查器，并确保您将 Animator 对象拖到 **CharacterAnimator** 插槽中。

您的角色确实会设置动画，但是当您想要时，事情不会发生 引擎所做的只是发送和更新动画参数。如果这些确实更新，则引擎正在正确执行其工作。您可以通过按下 **play**，在运行时选择角色的动画师并执行一个动作（如跳跃）来仔细检查。如果 **jump** 参数按时变为 **true**（有时是一段长时间，有时根据参数只有一帧），引擎方面一切正常。该参数更改通常是即时的，只要您执行操作，只有少数例外。

如果引擎和逻辑方面一切正常，剩下要做的就是调整动画师的过渡。这可以像在任何 Unity 项目中一样完成，这里没有特定于引擎的内容。常见的违规者将是退出时间（持续时间和相关的复选框、过渡持续时间、中断源等），但不幸的是，没有通用规则，每个动画师都需要根据您的上下文和设置采用自己的方法。设置过渡可能非常复杂，有些人可能会觉得它不是很直观。Unity 有一些关于它的文档，您可能需要查看。