**重定向人形动画**

Mecanim 的最强大功能之一是重定向人形动画。这意味着您可以相对轻松地将同一组动画应用于各种角色模型。重定向只适用于已配置了 Avatar 的人形模型，因为这可以让我们在模型的骨骼结构之间建立对应关系。

**建议的层级视图结构**

使用 Mecanim 动画时，您可以期望场景包含以下元素：

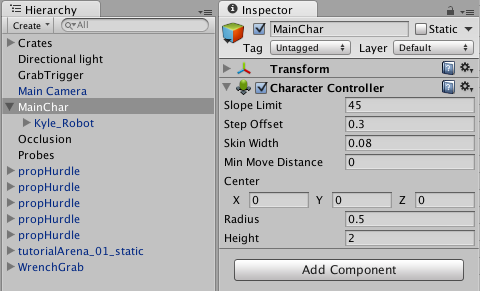
* 已导入的具有 Avatar 的角色模型。
* 引用 Animator Controller 资源的 Animator 组件。
* 从 Animator Controller 引用的一组动画剪辑。
* 角色的脚本。
* 角色相关组件，例如角色控制器。

项目还应包含具备有效 Avatar 的另一个角色模型。

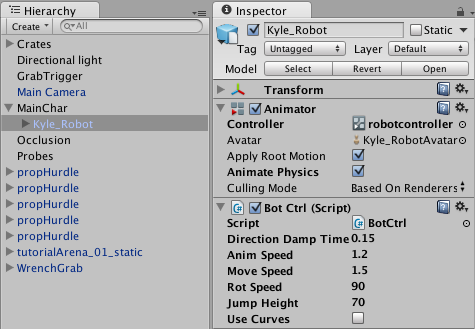
如果对术语有疑问，请参考[动画术语表](file:///E:\UnityDocumentation\Manual\AnimationGlossary.html)

建议的设置是：

* 在层级视图中，创建包含角色相关组件的游戏对象



* 将模型作为游戏对象的子代与 Animator 组件放在一起

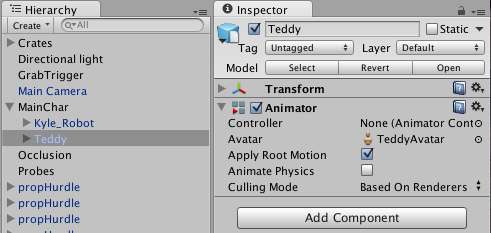


* 确保引用 Animator 的脚本正在寻找子代（而不是根）中的 Animator；使用 GetComponentInChildren<Animator>() 而不是 GetComponent<Animator>()。

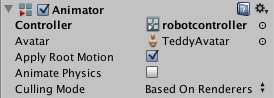


然后，为了在另一个模型上复用相同动画，您需要：

* 禁用原始模型
* 将所需模型作为游戏对象的另一子代拖入



* 确保新模型的 Animator Controller 属性正在引用同一控制器资源



* 调整顶级游戏对象上的角色控制器、变换和其他属性以确保动画与新模型顺利融合。
* 任务完成！



# 性能和优化

本页面提供一些提示来帮助您从 [Unity 的动画系统](file:///E:\UnityDocumentation\Manual\AnimationOverview.html)获得最佳性能，内容涵盖角色设置、动画系统和运行时优化。

## 角色设置

### 骨骼数量

在某些情况下，您需要创建具有大量骨骼的角色：例如，在需要大量可自定义的连接时。这些额外骨骼会增加构建大小，并且每个额外的骨骼都可能带来相对处理成本。例如，在\_\_通用\_\_模式下，已拥有 30 个骨骼的骨架上增加 15 个骨骼需要多花费 50% 的时间来解析。请注意，对于[通用](file:///E:\UnityDocumentation\Manual\GenericAnimations.html)和[人形](file:///E:\UnityDocumentation\Manual\ConfiguringtheAvatar.html)类型，您可以额外添加骨骼。如果未播放使用额外骨骼的动画，则处理成本应该可以忽略不计。如果连接不存在或为隐藏状态，此成本甚至会更低。

### 多个蒙皮网格

尽可能合并蒙皮网格。将一个角色拆分为两个[蒙皮网格渲染器](file:///E:\UnityDocumentation\Manual\class-SkinnedMeshRenderer.html)会降低性能。如果角色只有一种材质，那就更好，但在某些情况下，可能需要多种材质。

## 动画系统

### 控制器

未设置[控制器](file:///E:\UnityDocumentation\Manual\class-AnimatorController.html)的 [Animator](file:///E:\UnityDocumentation\Manual\class-Animator.html) 不会花时间执行处理。

### 简单动画

播放没有混合的单个[动画剪辑](file:///E:\UnityDocumentation\Manual\class-AnimationClip.html)会使 Unity 的速度比[旧版动画系统](file:///E:\UnityDocumentation\Manual\Animations.html)更慢。旧系统非常直接，对曲线采样并直接写入变换中。Unity 的当前动画系统具有用于混合的临时缓冲区，并会对采样曲线和其他数据进行额外复制。当前系统布局已针对动画混合和更复杂设置进行优化。

### 缩放曲线

动画化缩放曲线比动画化移动和旋转曲线的成本更高。为了改善性能，请避免使用缩放动画。

**注意：**这不适用于常量曲线（具有相同[动画剪辑](file:///E:\UnityDocumentation\Manual\AnimationClips.html)长度值的曲线）。常量曲线经过优化，成本低于比普通曲线。常量曲线的值与默认场景值相同时，常量曲线不会每帧都写入场景。

### 层

大多数时间，Unity 都在估算动画，并将[动画层](file:///E:\UnityDocumentation\Manual\AnimationLayers.html)和[动画状态机](file:///E:\UnityDocumentation\Manual\AnimationStateMachines.html)的开销保持在最低水平。向 Animator 添加另一层（无论同步与否）的成本取决于层播放的动画和混合树。层的权重为零时，Unity 会跳过层更新。

### 人形动画类型与通用动画类型

以下提示可帮助您选择具体类型：

* 导入人形动画时，如果不需要 IK（反向动力学）目标或手指动画，请使用 Avatar 遮罩 (class-AvatarMask) 将它们移除。
* 使用通用类型时，使用根运动比不使用根运动的成本更高。如果动画没有使用根运动，请确保未指定根骨骼。

### 场景级别优化

可进行许多优化，一些有用的提示如下：

* 使用哈希而不是字符串来查询 Animator。
* 实现一个小的 AI 层来控制 Animator。您可以让它为 OnStateChange、OnTransitionBegin 和其他事件提供简单回调。
* 使用状态标记可轻松地将 AI 状态机与 Unity 状态机匹配。
* 使用其他曲线来模拟事件。
* 使用其他曲线来标记动画；例如，与[目标匹配](file:///E:\UnityDocumentation\Manual\TargetMatching.html)一起使用。

## 运行时优化

### 可见性和更新

始终通过将 Animator 的 [Culling Mode](file:///E:\UnityDocumentation\Manual\class-Animator.html) 设置为 **Based on Renderers** 来优化动画，并禁用[蒙皮网格渲染器的](file:///E:\UnityDocumentation\Manual\class-SkinnedMeshRenderer.html) **Update When Offscreen** 属性。这样即可在角色不可见时让 Unity 不必更新动画。