



# 【 Unity基础教程 】 重点知识汇总

( 十八 )

Unity中Animator与Animation的区别

# 前言



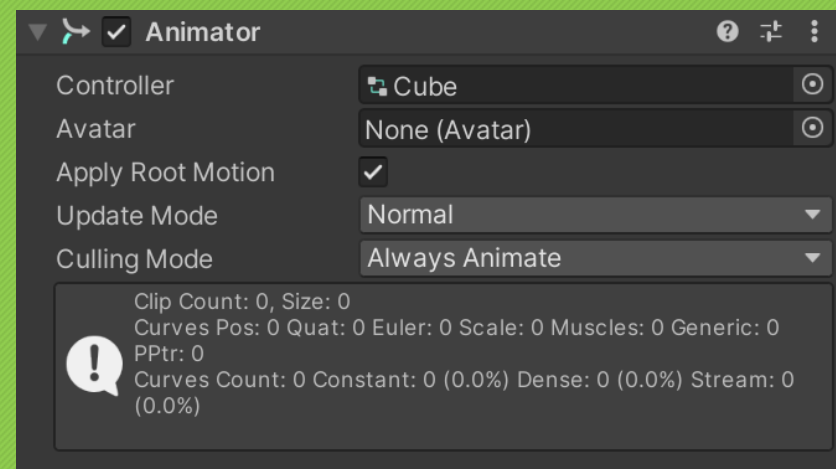
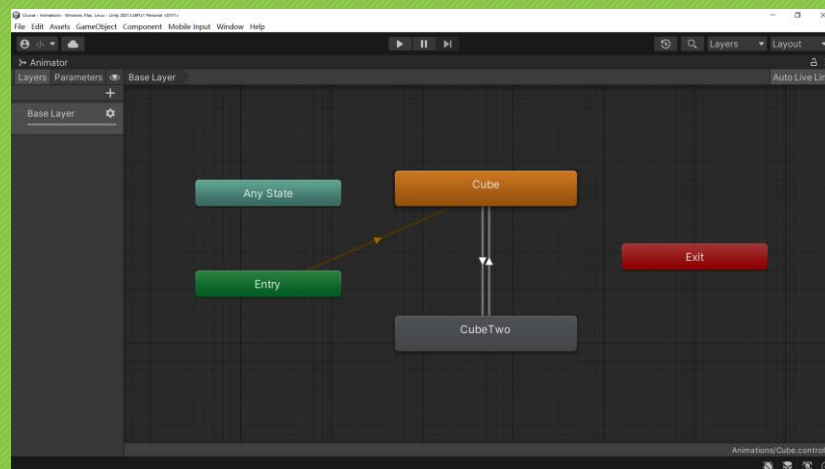
在Unity中，Animator和Animation都是用于控制对象的动画的**组件**，但它们各自有不同的用途和功能。下面将从**概念、功能、适用场景和使用方式**等方面详细讲解。

# Animator（动画控制器）



**概念：**Animator是Unity的新一代动画系统（**Mecanim**），用于控制复杂动画状态（更高级）。它依赖**Animator Controller**进行动画状态的管理和切换，支持**状态机**、**Blend Tree**、**动画过渡**、**IK**、**参数控制**等。适合用来处理复杂的动画系统，比如角色动画、UI动画等。

**示例：**假设我们有一个角色，它有走、跑和跳三种动画。我们可以创建一个Animator Controller，并为每种状态设置**动画片段**。在状态之间通过设置条件（如速度）来进行过渡。

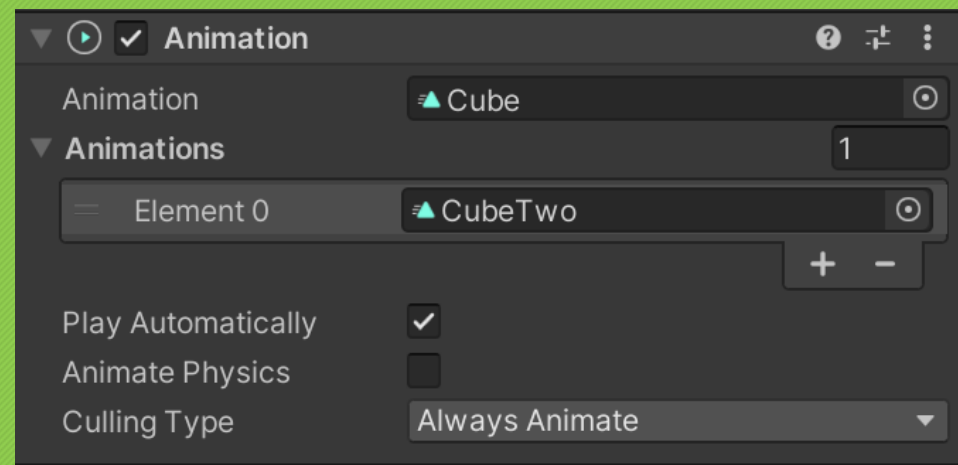
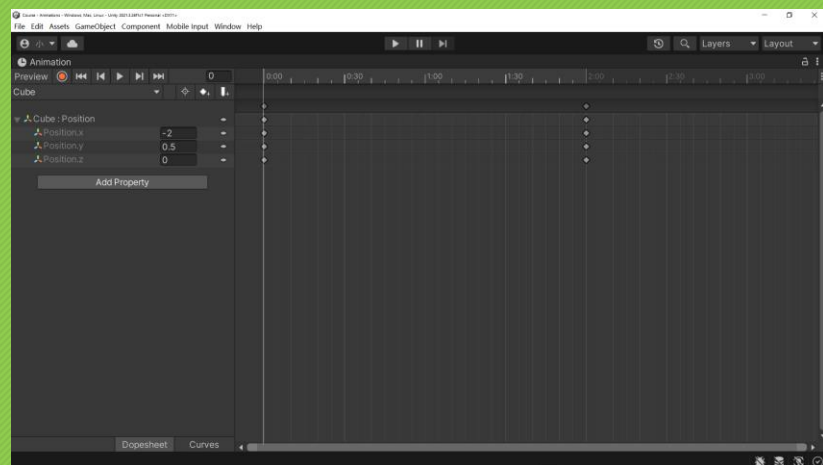


# Animation（旧版动画组件）



**概念：** Animation是Unity早期的动画系统，主要用于播放（直接管理）简单的**剪辑动画（Animation Clip）**，**不支持动画状态机，也不支持参数控制**。适用于简单的动画播放（如开门、按钮点击等），一般用于较简单的场景或静态物体。

**示例：** 假设我们有一个简单的物体（如一个方块），我们只想让它在场景中上下移动。在这种情况下，我们可以直接使用Animation组件来播放一个简单的动画片段，而不需要创建复杂的状态机。



# Animator与Animation区别（总结）



| 对比项   | Animator (Mecanim)   | Animation (旧动画系统)  |
|-------|--|--|
| 控制方式  | 通过 <b>Animator Controller</b> 控制动画状态                                       | 直接拖拽 <b>Animation Clip</b> 即可  |
| 适用场景  | 适用于复杂动画系统（如角色动画）   | 适用于简单动画（如开关门动画）  |
| 动画过渡  | 支持动画过渡、 <b>Blend Tree</b>  | 不支持动画过渡，动画切换时会立刻播放   |
| 脚本控制  | 通过 <b>Animator</b> 组件提供的 <code>SetTrigger</code> 、 <code>SetBool</code> 控制 | 通过 <b>Animation</b> 组件的 <code>Play()</code> 、 <code>Stop()</code> 控制 |
| IK 支持 | 支持 <b>IK</b> （反向运动学）   | 不支持 <b>IK</b>  |
| 性能开销  | 性能开销较高   | 性能开销较小   |

\*反向运动学（Inverse Kinematics，简称 IK）是一种用于计算物体或角色的运动学方法，特别是在动画和机器人学中，它的目标是通过给定末端执行器（如角色的手或脚）的位置或方向，计算出关节的角度或位置，以实现期望的姿势。

简而言之，反向运动学是**从结果推算出过程，即已知最终目标位置，计算出如何调整中间的各个关节位置。**



# 【 Unity基础教程 】 重点知识汇总

( 十八 )

Unity中Animator与Animation的区别