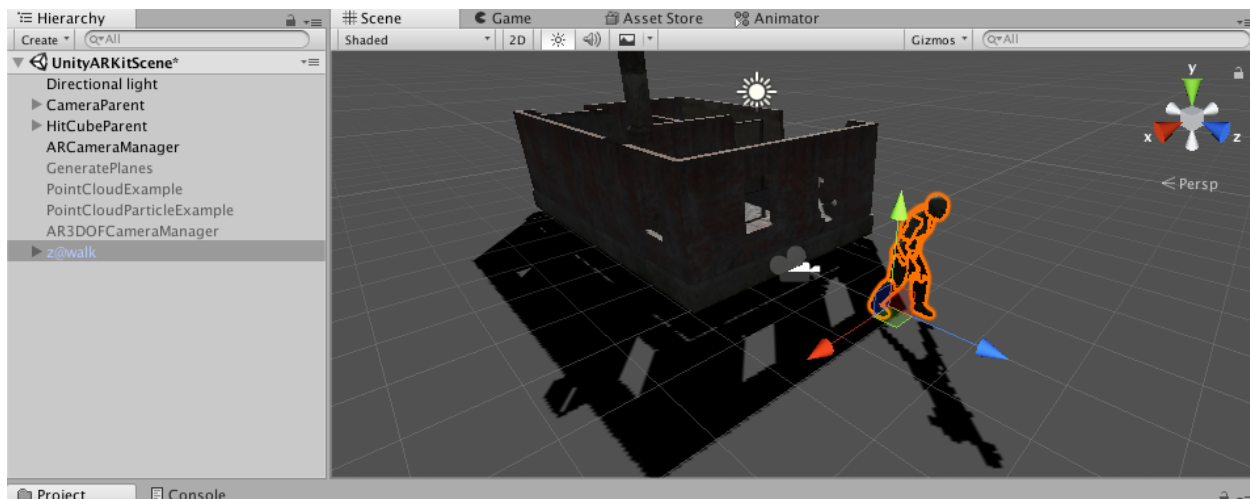


继续我们的学习。

接下来我们将把游戏中的敌人添加到游戏中，在Project视图找到Assets-Components-Zombie-Model中的z@walk，并将其拖到Hierarchy视图中。

接下来使用移动工具将敌人移动到合适的位置，如下图。

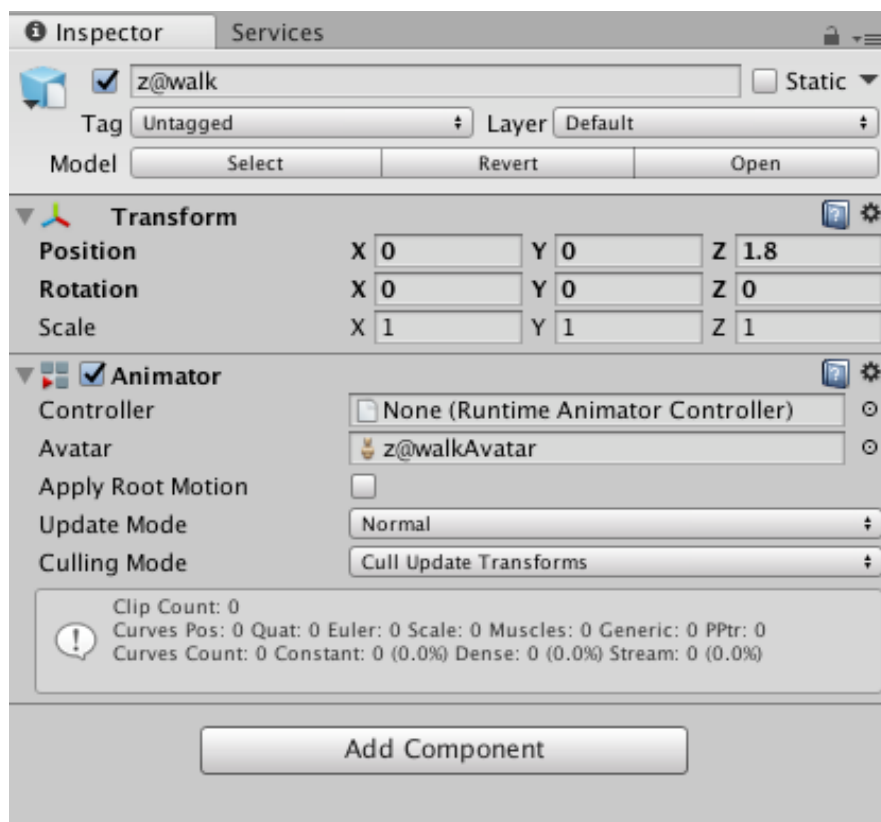


接下来我们将给这个敌人添加动画。

在Unity中有两种方式给游戏对象添加对象，分别是Legacy Animation和使用Animator的方式（又被称为Mecanim动画系统）。

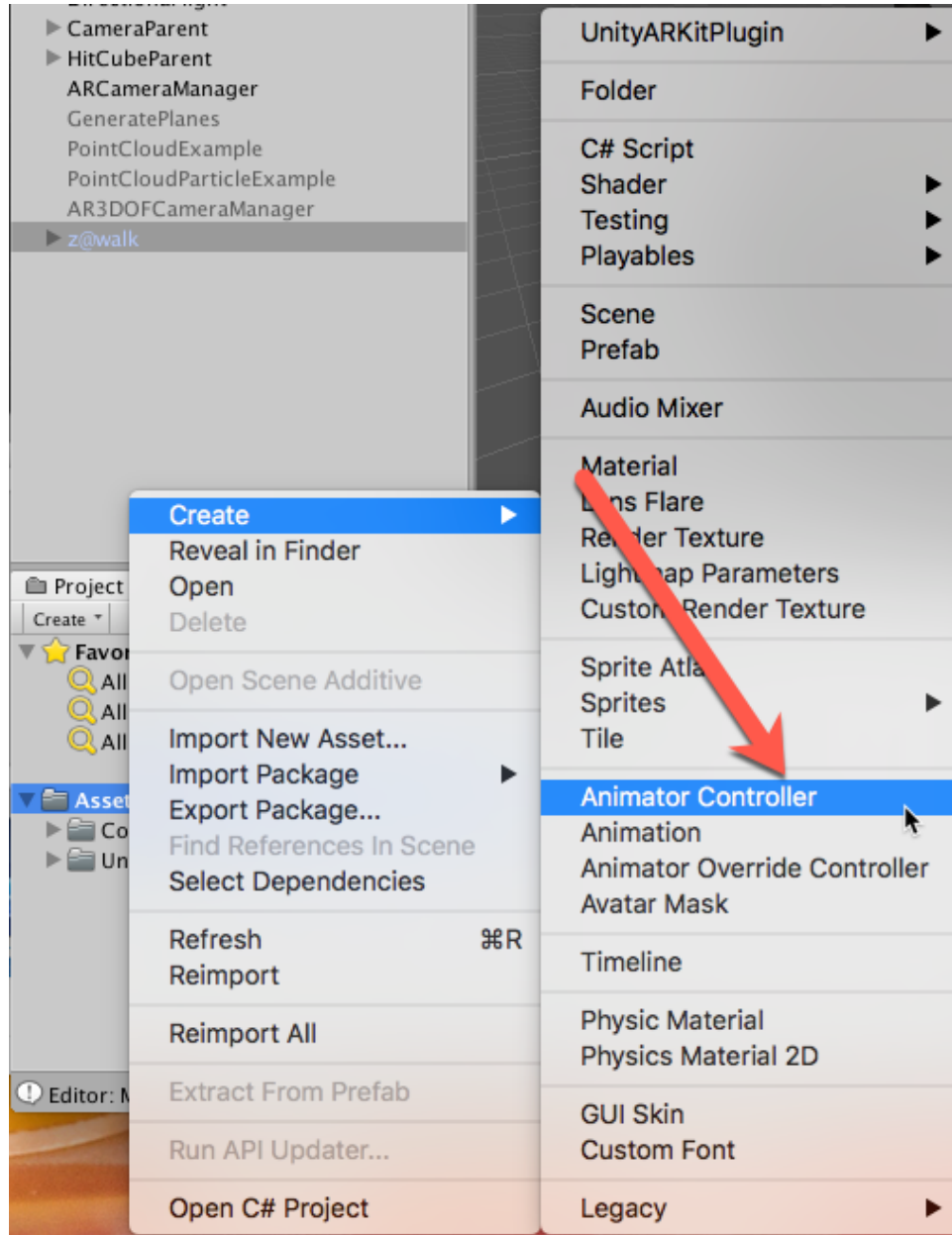
对于游戏角色来说，目前我们基本上都会使用Animator。

在Hierarchy视图中选中z@walk游戏对象，在Inspector中可以看到有一个Animator组件。

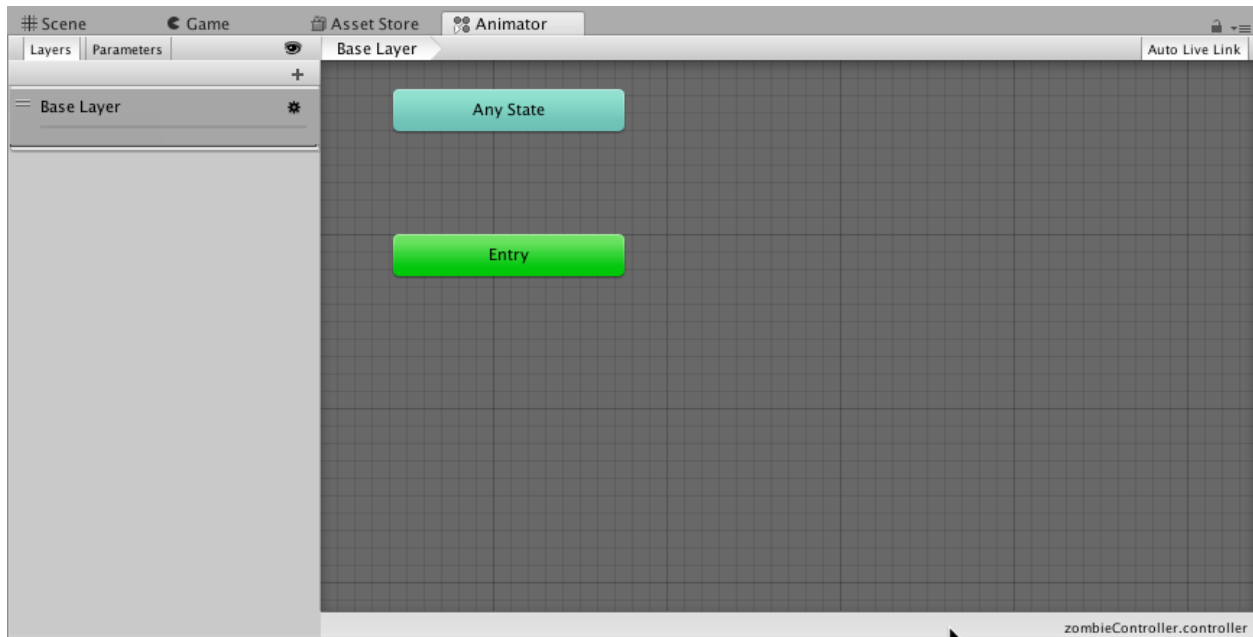


此时Animator组件中的Controller属性是None，也就是还没有创建合适的Animator。

因此我们要在Unity的Project视图中右键单击，选择Create-Animator Controller，



将其更名为zombieController，双击将其打开，在Unity界面的中间部分会打开一个Animator视图，如下图所示。



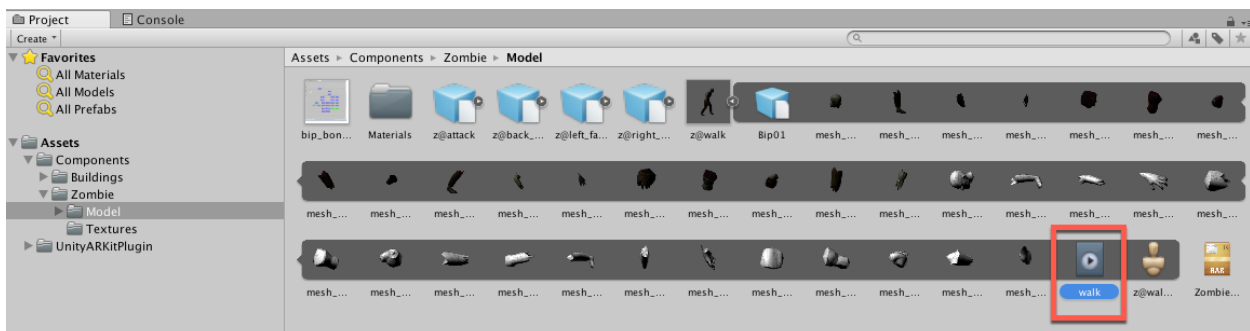
对于Animator的使用，有一个所谓“状态机”的概念，

在绝大多数情况下，我们都需要随着游戏环境的变化让游戏对象在不同的动画状态之间进行切换。举个简单的例子，在某个游戏中，当我们按下键盘上的空格键，或者是游戏控制器上的某个特定按键时，玩家所控制的游戏角色会从行走动画切换到跳跃动画。即便某个游戏对象只有一个对应的 Animation Clip，我们也需要让其归于一个Animator Controller的控制下。

Animator Controller使用State Machine（状态机）来管理游戏对象的不同动画状态及其之间的过渡。这个名词听起来有点可怕，但实际上很好理解。它可以看做某种类型的流程图，或是使用Unity内置的可视化编程语言所编写的小程序。我们可以在Animator视图中创建、浏览和修改Animator Controller的结构。

关于理论的部分暂不多讲，我们来通过具体的操作来熟悉如何设置Animator。

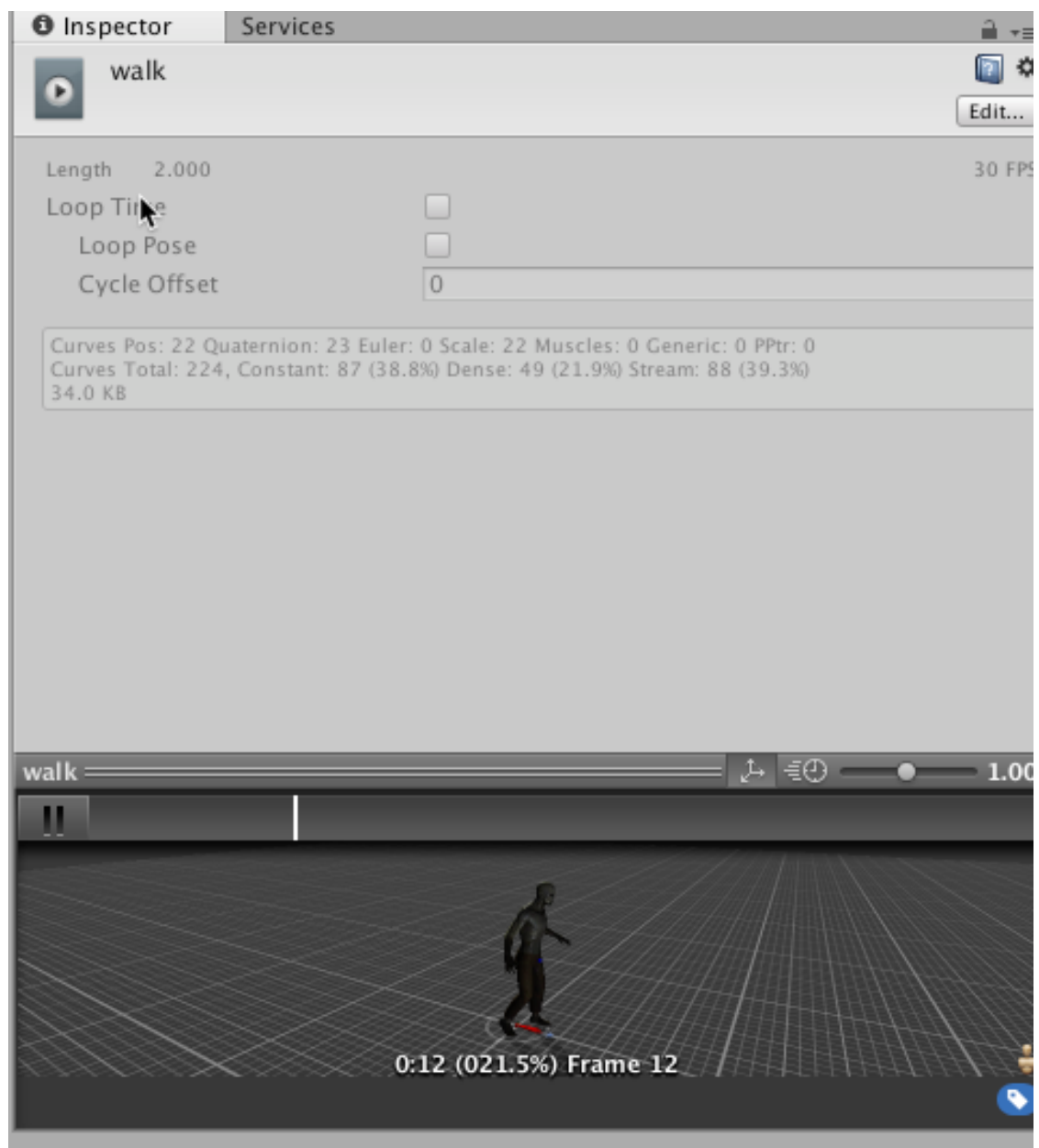
在Project视图中找到Assets-Components-Models-z@walk，将其展开，可以看到里面有一个名为walk的动画片段，也就是带一个播放按钮标志的文件，如下图所示。



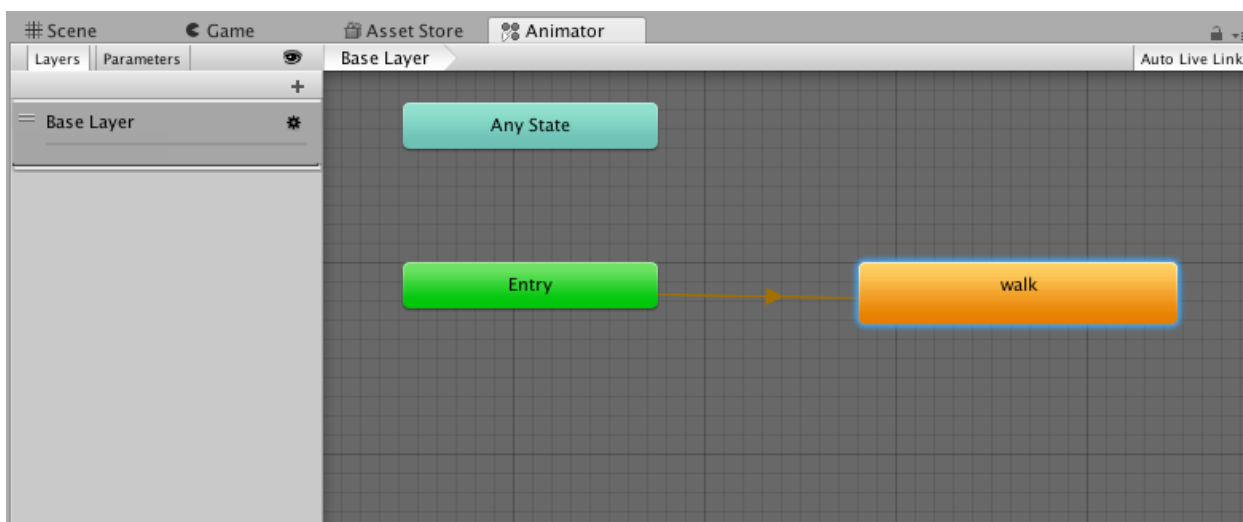
Mecanim动画系统中的一个核心概念是Animation Clips（动画片段），其中包含了丰富的动画信息，如特定的对象将如何更改其位置、旋转及其它属性。每个动画片段都可被视为一个简单的线性记录。

Unity支持使用第三方软件所创建的动画片段，如Max或Maya，或是使用动作捕捉设备及软件所获取的动画片段。

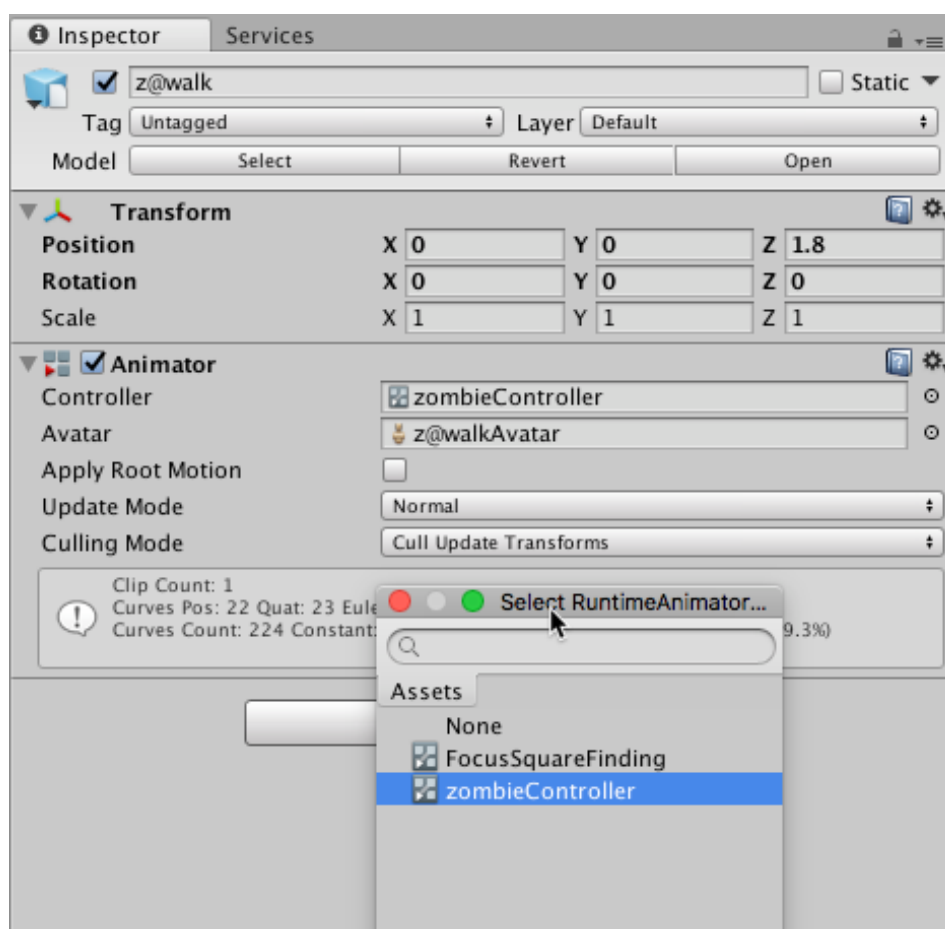
在Project视图中选中walk动画片段，在Unity右侧的Inspector视图下方可以看到该动画的预览，点击播放按钮就可以预览动画。



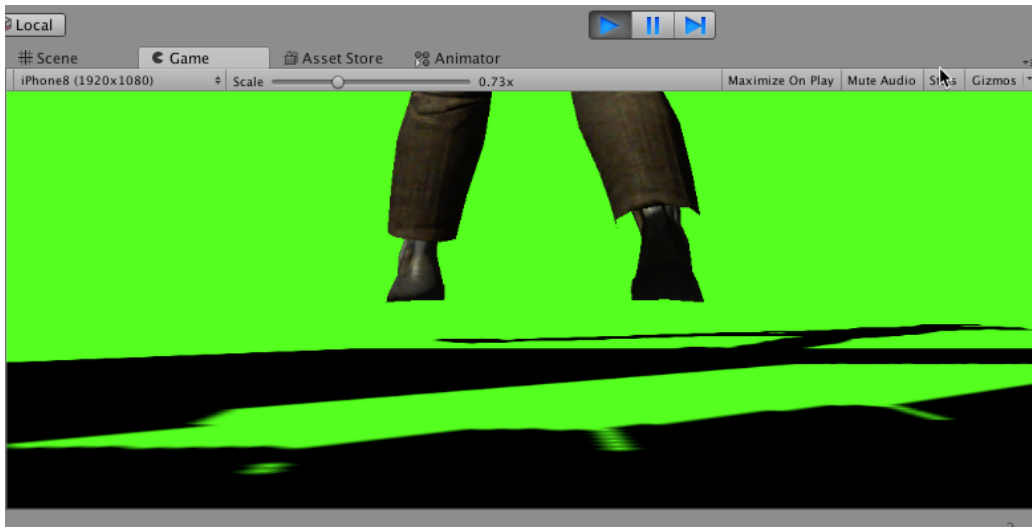
把walk这个动画片段拖动到Animator视图中，会看到在Entry和walk之间自动创建了一个关联。



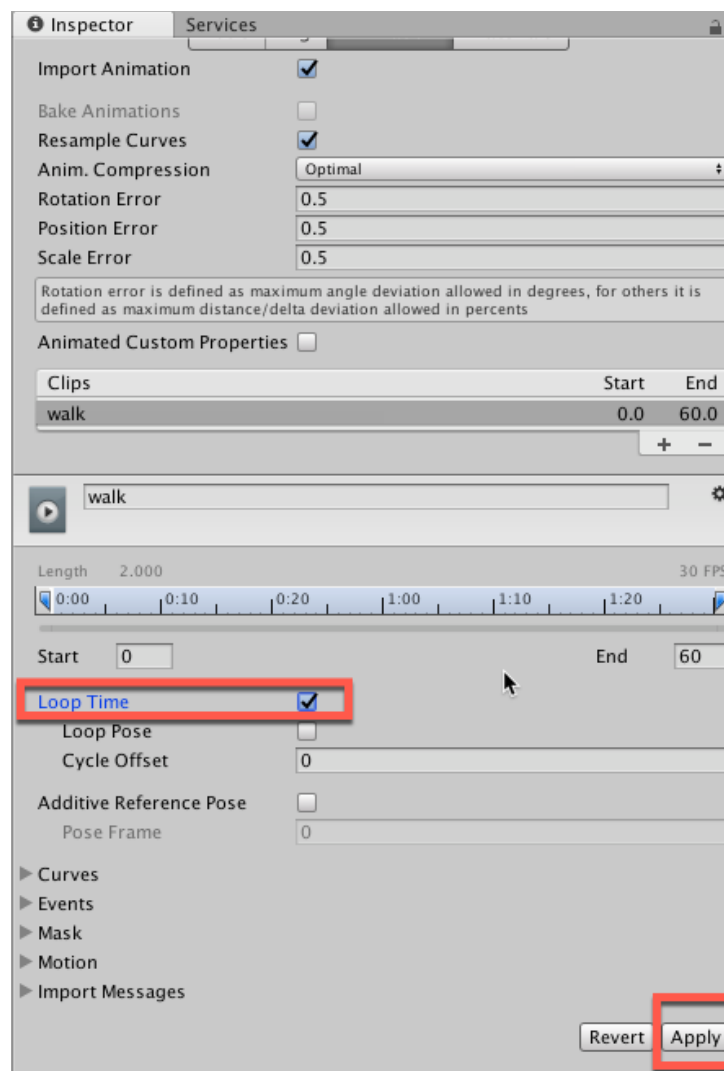
然后在Hierarchy视图中选择z@walk游戏对象，在Inspector视图的Animator组件中，在Controller属性处点击右侧的小圆圈，并选择刚刚创建的zombieController。



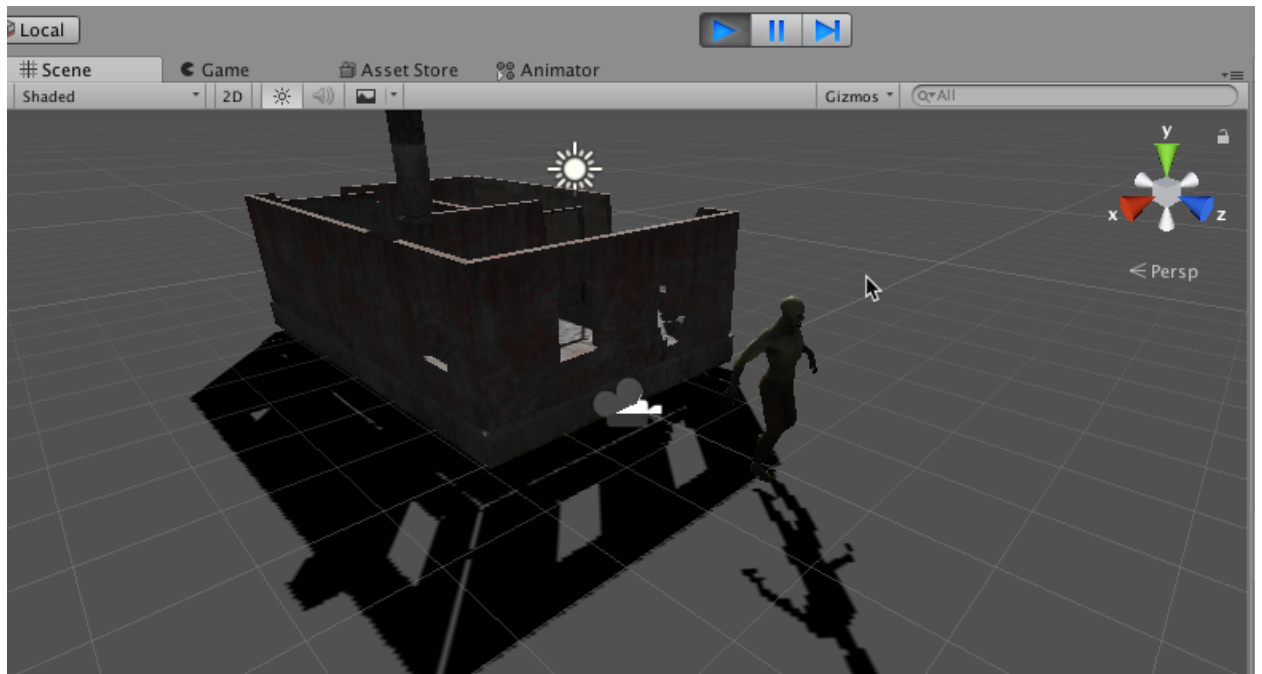
点击工具栏上的播放按钮，可以看到敌人的动画只播放了一次就停止了，这个问题需要修复。



在Project视图中找到刚才的walk动画片段，然后点击Inspector视图右上角的Edit按钮，勾选Loop Time选项，然后点击Apply按钮即可。



此时再次点击预览，会发现敌人角色的动画是循环播放的了。



注意一个小细节，刚才我们是在Game视图中查看的，现在则切换到了Scene视图中查看。