**Unity3D教程：动画系统教程（一）**

Unity的动画系统非常的灵活强大，动画系统支持动画融合、混合、叠加动画，行走循环的时间同步，动画层，控制动画的各个方面，以及支持基于物理的布娃娃系统和程序动画。这里做个笔记文章，记录整个从0开始使用Unity的动画系统。下载官网的CharacterAnimation例子里面带有3个动画模型，以及这三个模型的简单调用的例题。[Unity3D](http://www.unitymanual.com/" \t "_blank)教程手册

[Unity3D教程](http://www.unitymanual.com/)：动画系统教程（一）

这里例题中包含三个基本的数据模型，Goober是一个简单的小丑，Robot则是一个机器人战士，Soldier是一个人类战士。

Unity3D教程：动画系统教程（一）

在自带的动画模型里，除了模型Robot以外，还包含了很多@XXX结尾的动作，我们调用的时候，只需要按照@后面的动作名称调用即可。  
1）新建一个场景，创景地形，并且将Robot模型拖到场景中。

Unity3D教程：动画系统教程（一）

这个时候的场景中是没有任何代码的，直接运行场景，你就能看到这个模型在运行“idle”动作。

Unity3D教程：动画系统教程（一）

看来Unity不但自己加载了模型，而且识别出了动作，自动调用了待机动作。为了防止Unity自己的动作影响到我们的代码操作，在开始的时候我们就把默认动作关闭。创建一个JS脚本，加入如下代码：

function Start () { animation.Stop(); }<!--DVFMTSC-->

  在动画播放方式上，Unity有两种方法，一种是Play()，这种方式的动画没有任何混合，直接播放动作，这样的动作看起来比较生硬，而且在Unit用的官方文档里建议个别动画可以使用Play()方式播放，那似乎是不建议使用Play来播放动画；另外一种是Animation.CrossFade。这是一种再带混合的动画播放模式，他可以让动画逐渐变化，而不是生硬的切换。Unity3D教程手册  
  
    这就是动画融合的用武之地。在Unity同一个角色可以拥有任意数量的动画。所有动画被融合或添加在一起，来产生最终

// Makes a character contains a Run and Idle animation // fade between them when the player wants to move // 让一个角色包含Run和Idle动画，并且在玩家想移动的时候在他们之间淡入淡出。 function Update () { if (Mathf.Abs(Input.GetAxis("Vertical")) > 0.1) animation.CrossFade("Run"); else animation.CrossFade("Idle"); }

将上面这段代码拷贝到JS脚本中，我们按方向键就可以看到刚刚的Robot在“Run”和“Idle”两个动画柔和切换的动作变化。  
  
    如今的游戏，Animation Blending是确保角色平滑动画的一项重要功能。动画师创建的分段动画，如：行走循环、跑步循环、 空闲动画或射击动画。在游戏的任何时间点都有可能从空闲转换到走动，反之亦然。当然你不希望两个不同的动作之间突然跳转，想要动画平滑过渡。  
  
    通过animation还可以设定动画的运行方式，可以animation.WrapMode方式来修改是否循环播放。

animation.wrapMode = WrapMode.Loop; animation["jump"].wrapMode = WrapMode.Clamp;

动画层可以控制玩家的动画播放优先级，具体的用法似乎还挺丰富的，从文档上看，最起码的用途是可以在其他的动画还在播放的时候，优先播放而不影响底层级的动画。比如当层级都是1，在跑动的时候，点击射击按钮，则没有反应会一直奔跑。如果设定射击的动作层级为1，其他都为0的时候，跑动的时候点射击，角色就会执行射击动作。

function Start () { // Set all animations to loop animation.wrapMode = WrapMode.Loop; // except shooting animation["shoot"].wrapMode = WrapMode.Once; // Put idle and walk into lower layers (The default layer is always 0) // This will do two things // - Since shoot and idle/walk are in different layers they will not affect // each other's playback when calling CrossFade. // - Since shoot is in a higher layer, the animation will replace idle/walk // animations when faded in. animation["shoot"].layer = 1; // Stop animations that are already playing //(In case user forgot to disable play automatically) animation.Stop(); }

function Update () { if (Mathf.Abs(Input.GetAxis("Vertical")) > 0.1 || Mathf.Abs(Input.GetAxis("Horizontal")) > 0.1) animation.CrossFade("run"); else animation.CrossFade("idle"); // Shoot if (Input.GetButtonDown ("Fire1")) animation.CrossFade("shoot"); }