**Unity3D教程：动画合成教程（Animation Blending）**

Posted on 2013年07月02日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 93 次

现在的游戏,动画合成是一个必不可少的特征为确保角色有平滑的动画.动画师创建单个动画,如走路循环,跑步循环,空闲动画或射击动画.有些时候你的游戏需要能转换空闲动画到走路循环或反之.当然你不想要在运动中有突然的跳跃,你想要动画平滑转换.

就是动画合成到来的原因.在Unity你能在同一个角色身上有任意多个动画播放.所有的动画合成或添加到一起产生最终动画.

我们的第一步是使角色平稳合成空闲和走路动画.为了做这个我们使用简单的脚本,我们首先设置卷模式Wrap Mode的动画循环Loop.接着我们关闭自动播放使我们的脚本只有一个正在演奏的动画.

我们的第一个角色动画脚本十分简单;我们值需要一些方法知道我们的动画运行的速度,再减退走路和空闲动画.为了这个简单的测试我们使用pre-setup输入轴.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | function Update () |
| 02 |  |
| 03 | { **if** (Input.GetAxis(“Vertical”)> 0.2) |
| 04 |  |
| 05 | animation.CrossFade (“walk”); |
| 06 |  |
| 07 | **else** |
| 08 |  |
| 09 | animation.CrossFade (“idle”);} |
| 10 |  |
| 11 | Animation Layers |

**动画层次**

层次是一个令人惊讶的使用观念,允许你分组动画和区分权重.在Unity动画系统,你能混合你想要的任意多的动画片段.你能指定混合权重手动或简单的使用animation.CrossFade(),这是动画的自动权重.

混合权重总是正常的在应用之前

我们假定你有一个走路循环和一个跑步循环,每个都有权重为1(100%).当Unity产生最终动画他将恢复正常权重,意思是走路贡献为50%动画,跑步也贡献50%.

这很好,但是有时你想要区分2个同时演奏的动画那一个有更多的权重.当然你要确保权重之和为100%,但是他使用层次可以很容易的达到这个目的.

**层次例子**

一个例子,你可能有一个射击动画,一个空闲和走路循环.你将想要基于玩家的速度逐渐褪去走路和空闲的动画.但是当玩家开火你想要值展示开火动画.因此开火动画本质上有更高的权力.

简单的方法是保持走路和空闲动画在开火时.这样我们需要使开火的动画层次高于空闲和走路.意思是射击动画将有最高权重.走路和空闲动画只有在射击动画不使用100%混合权重使才能接收权重.所有当射击动画在CrossFading里,权重将从0开始一会到达100%.开始时走路和空闲层次将任然接收混合权重,知道射击动画完成逐渐增强,他们将不接收权重.我们需要严格执行!

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | function Start () |
| 02 |  |
| 03 | { *// Set all animations to loop设置所有动画为循环状态* |
| 04 |  |
| 05 | animation.wrapMode = WrapMode.Loop; |
| 06 |  |
| 07 | *// except shooting除了射击* |
| 08 |  |
| 09 | animation[“shoot”].wrapMode =WrapMode.Once; |
| 10 |  |
| 11 | *// Put idle and walk into lower layers (Thedefault layer is always 0)放置空闲和走路到底层次(缺省为0)* |
| 12 |  |
| 13 | *// This will do two things这里做2件事* |
| 14 |  |
| 15 | *// Since shoot and idle/walk are indifferent layers they will not affect射击和空闲/走路在不同层次没影响* |
| 16 |  |
| 17 | *// each other's playback when callingCrossFade.当呼叫CrossFade重放每个* |
| 18 |  |
| 19 | *// Since shoot is in a higher layer, theanimation will replace idle/walk射击在高层次,他取代空闲和走路* |
| 20 |  |
| 21 | *//Unity3D教程：www.unitymanual.com* |
| 22 |  |
| 23 | *// animations when faded in.动画将逐渐增强* |
| 24 |  |
| 25 | animation[“shoot”].layer = 1; |
| 26 |  |
| 27 | *// Stop animations that are already playing停止动画确实在演奏时* |
| 28 |  |
| 29 | *//(In case user forgot to disable playautomatically)别忘了关闭自动播放* |
| 30 |  |
| 31 | animation.Stop();} |
| 32 |  |
| 33 | function Update () { |
| 34 |  |
| 35 | *// Based on the key that is pressed,基于按下键* |
| 36 |  |
| 37 | *// play the walk animation or the idleanimation演奏走路或空闲动画* |
| 38 |  |
| 39 | **if**(Mathf.Abs(Input.GetAxis(“Vertical”))> 0.1) |
| 40 |  |
| 41 | animation.CrossFade(“walk”); |
| 42 |  |
| 43 | **else** |
| 44 |  |
| 45 | animation.CrossFade(“idle”); |
| 46 |  |
| 47 | *// Shoot射击* |
| 48 |  |
| 49 | **if** (Input.GetButtonDown(“Fire1”)) |
| 50 |  |
| 51 | animation.CrossFade(“shoot”);} |
| 52 |  |

缺省的animation.Play()和animation.CrossFade() 将停止或减弱动画在相同的层次.你想要避免这是严格的.在我们的射击,空闲,跑步例子,演奏空闲和跑步将不影响射击动画反之也可以(你可以改变行为通过设置一个参数给animation.CrossFade如果你想).

Additive Animations andAnimation Mixing

**添加动画和动画混合**

添加动画和动画混合允许你减少你创建的游戏的动画数量,且很重要的创建面部动

我们假设你想要创建一个角色倾向一边当跑步和转弯时. 你已经有了走路和跑步循环,现在你要制作单独的左倾向走,右倾向走,左倾向跑,右倾向跑动画.

但是这样意味着你有翻倍数量的动画工作!创建一个庞大的动画数量不是可取的.添加动画和混合能解决

**添加动画例子**

添加动画允许你覆盖动画效果到另一个之上演奏.当使用添加动画,Unity将计算动画片段的第一帧和现在帧的不同.接着他将应用这种不同到所有其他动画

现在你值有偏左和偏右动画.Unity将放置这个动画层次高于走路,空闲或跑步循环.

这是制造这些的代码:

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **private** **var** leanLeft : AnimationState; |
| 02 |  |
| 03 | **private** **var** leanRight : AnimationState; |
| 04 |  |
| 05 | function Start () |
| 06 |  |
| 07 | {leanLeft =animation[“leanLeft”]; |
| 08 |  |
| 09 | leanRight =animation[“leanRigh”]; |
| 10 |  |
| 11 | *// Put the leaning animation in a seperatelayer设置偏向动画在指定层* |
| 12 |  |
| 13 | *// So that other calls to CrossFade won'taffect it.所有呼叫CrossFade不会影响他* |
| 14 |  |
| 15 | leanLeft.layer = 10; |
| 16 |  |
| 17 | leanRight.layer = 10; |
| 18 |  |
| 19 | *// Set the lean animation to be additive设置偏向动画为添加物* |
| 20 |  |
| 21 | leanLeft.blendMode =AnimationBlendMode.Additive; |
| 22 |  |
| 23 | leanRight.blendMode =AnimationBlendMode.Additive; |
| 24 |  |
| 25 | *// Set the lean animation ClampForever设置偏向动画ClampForever* |
| 26 |  |
| 27 | *// With ClampForever animation's will notautomatically设置ClampForever动画不自动播放* |
| 28 |  |
| 29 | *//Unity3D教程：www.unitymanual.com* |
| 30 |  |
| 31 | *// stop when reaching the end of the clip停止当到片段最后* |
| 32 |  |
| 33 | leanLeft.wrapMode = WrapMode.ClampForever; |
| 34 |  |
| 35 | leanRight.wrapMode = WrapMode.ClampForever; |
| 36 |  |
| 37 | *// Enable the animation and fade it incompletely激活动画和逐渐增强到完全* |
| 38 |  |
| 39 | *// We don't use animation.Play here becausewe manually adjust the time我们不使用animation.Play因为要手动调整时间* |
| 40 |  |
| 41 | *// in the Update function.在Update函数里* |
| 42 |  |
| 43 | *// Instead we just enable the animation andset it to full weight替代我们激活的动画和设置他为完全权重* |
| 44 |  |
| 45 | leanRight.enabled = **true**; |
| 46 |  |
| 47 | leanLeft.enabled = **true**; |
| 48 |  |
| 49 | leanRight.weight = 1.0; |
| 50 |  |
| 51 | leanLeft.weight = 1.0; |
| 52 |  |
| 53 | *// For testing just play run animation andloop it尝试运行动画和循环他* |
| 54 |  |
| 55 | animation[“walk”].wrapMode =WrapMode.Loop; |
| 56 |  |
| 57 | animation.Play(“walk”);} |
| 58 |  |
| 59 | *// Every frame just set the normalized time每帧设置为正常时间* |
| 60 |  |
| 61 | *// based on how much lean we want to apply基于我们要应用的偏向有多少* |
| 62 |  |
| 63 | function Update (){ |
| 64 |  |
| 65 | **var** lean =Input.GetAxis(“Horizontal”); |
| 66 |  |
| 67 | *// normalizedTime is 0 at the first frameand 1 at the last frame in the clip 。normalizedTime在片段的第一帧为0和在最后帧为1* |
| 68 |  |
| 69 | leanLeft.normalizedTime = -lean; |
| 70 |  |
| 71 | leanRight.normalizedTime = lean;} |

建议:当使用添加动画,关键是你既要使用没有添加动画在每个改变,又要使用添加动画,不同的是动画将在最后帧的结果上.这必然不是你想要的.

**程序控制动画角色**

有时你想要拉动你的角色骨骼通过程序.例如你可能想要你的角色的头看向3D空间的某点.这用脚本很好解决.幸好,Unity做这个很简单.在Unity所有的骨骼是改变驱使表皮网格.因此你能脚本骨骼角色,像其他游戏物体一样

要知道的一个很重要的事情是动画系统是在Update()函数之后更新改变,和在LateUpdate()函数之前被呼叫.因此如果你想要使用LookAt()函数,你要让他在LateUpdate()里面,以确保你的动画确实高于一切.

玩偶被创建有相同的方式.你直接附上刚体,角色联合和碰撞仓给不同的骨骼.这将为身体动画你的表皮角色.