**iTween动画包教程**

Posted on 2013年04月07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 260 次

iTween是一个快速动画实现辅助开发的脚本，也即一个动画辅助脚本类，用它可以轻松实现各种动画、晃动、旋转、移动、褪色、上色、控制音频等等。如果你经常使用Flash AS，你就会知道Tween这个词,AS中也有很多Tween(过渡动画)的功能代码。

可能很多人不理解，既然有动画编辑器，为什么还需要用这个动画脚本呢？原因很简单,iTween最大的特点就是一步到位，即一个简单函数就能让物体实现一个完整的动画过程，非常方便，而不用通过动画编辑器来创建动画文件，再编辑曲线等。

在制作一些基本动画时，iTween更有时效的优势，同时也更节省资源。

iTween.moveTo(gameObject,{"x":1.7, "time":2});

iTween.moveTo函数后第一个参数是需要动画的物体，gameObject即代表自身，大括号跟随着一系列的参数表，x:代表x轴移动，time代表这个动画需要持续的时间，也即在2s的时间内x轴运动到1.7的位置。

private var go : GameObject;//需要动画的游戏物体  
private var cam : GameObject;//镜头  
function Awake(){  
//赋值  
go = gameObject;  
cam = camera.main.gameObject;  
}  
function Start(){  
iTween.rotateFrom(go,{"y":90, "time":1.5, "transition":"easeInExpo"});//旋转从90度到当前  
iTween.moveFrom(go,{"y":3.5, "time":1.5, "transition":"easeInExpo"});//从y值3.5移动到当前  
iTween.colorTo(go,{"r":3, "g":.5, "b":1.2, "time":.3, "delay":1.5});//颜色变化到新的值  
iTween.shake(cam,{"y":.3, "time":.8, "delay":1.5});//每1.5s震动一次镜头物体  
iTween.scaleTo(go,{"y":2, "time":2, "delay":2.3});//缩放到新比例  
iTween.rotateBy(go,{"x":.5, "delay":4.3});//每4.3s旋转0.5度  
iTween.moveTo(go,{"y":1.2, "delay":4.6});//移动到新位置  
iTween.moveTo(go,{"y":0, "delay":5.8, "transition":"easeInExpo"});//同前解释  
iTween.shake(cam,{"y":.3, "time":.8, "delay":6.8});//同前解释  
iTween.colorTo(go,{"r":.165, "g":.498, "b":.729, "time":.5, "delay":7.6});//同前解释  
iTween.scaleTo(go,{"y":1, "delay":7.6});//缩放到新比例  
}

**iTween动画设计包的使用**

Posted on 2013年04月07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 265 次

iTween动画设计包的使用。

相关文章：[**iTween动画包教程**](http://www.unitymanual.com/3715.html)

1.回调函数

回调函数，即当动画完成时那瞬间需要执行一次的一个函数，it中默认有一个onComplete函数，当动画完成时会自动执行，且你可以提供需要传递的一些参数。见如下代码：

private var tweenTarget : GameObject;  
var counter : GUIText;  
var count : int =0;

function Start(){  
tweenTarget=gameObject;  
roll("right");//开始执行  
counter.guiText.material.color = Color.black;  
}

private function roll(direction: String): void{  
switch(direction){  
//如果向右移  
case "right":  
iTween.rotateBy(tweenTarget,{"z":-.5});//旋转  
iTween.moveTo(tweenTarget,{"x":1.7, "onComplete":"roll", "onCompleteParams":"left"});//移动到 当完成动画时执行onComplete 并且传递一个参数left  
//也即相当于又这样调用了一次roll("left"); 可以推测到 当动画完成向右之后 又开始向左了  
count+=1;  
break;

//如果向左移 同上 当完成向左动画 则又开始向右 如此实现"乒乓"效果  
case "left":  
iTween.rotateBy(tweenTarget,{"z":1});  
iTween.moveTo(tweenTarget,{"x":-1.7, "onComplete":"roll", "onCompleteParams":"right"});  
count+=1;  
break;  
}

counter.text=count.ToString() + " Loops";  
}

2.Bezier贝塞尔曲线运动

除了普通的线性运动 如MoveTo这些函数 it也支持曲线运动 其中一种常用的曲线就是bezier

在it中如何使用曲线：

function Start()  
{  
iTween.moveToBezier(gameObject,{"time":3, "transition":"easeInOutQuint", "bezier":[Vector3(0,1.5,0), Vector3(0,0,1.5), Vector3(0,-1.5,3), Vector3(1.5,0,3), Vector3(1.5,0,0), Vector3(-1.5,0,0)]});  
}

可以看到，让物体跟随曲线运动。第一个参数指定当前物体运动，接下来是过渡方式：easeInOut 是一种起点和终点平滑过渡的方式。

共指定了6个向量点，需要记住参数是成双的即每个定点需要匹配一个控制点，才能定义好这条曲线。上面共有3个顶点，加3个这些定点的控制点。