**可视化路径插件iTween在Unity中的使用**

Posted on 2013年06月08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 90 次

iTween它由一个c#文件组成，支持Unity所有版本以及不管你用什么编程语言都可以使用它，如果你是用Js，你要将它放在Plugins文件夹里；如果是c#，放在项目的Assets里的任意目录即可。

相关文章：[iTween插件的介绍和用法](http://www.unitymanual.com/1683.html)

iTween的所有方法都有2种方式，一种简单的(simple），一种定制的复杂(customizable)的。 在复杂定制方式里可以设置iTween所有属性方法。iTween里的方法几乎都需要赋予一个GameObject，为了获取更大的性能，只有一些特定的 函数可以在Update方法中重复调用。

iTween可以做很多事情。我们先从简单的学起，先控制一个物体从一个位置移动到另一个地方，这里用到MoveTo()方法。让它两秒后移动到(2，0，0)位置上：

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

|  |  |
| --- | --- |
| 1 | iTween.MoveTo(gameObject， Vector3(2，0，0)， 3); |

上面是不是很简单?我们如果需要对这个动画进行更多的控制，我们需要用“customizable”模式，像下面这样写：

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

|  |  |
| --- | --- |
| 1 | iTween.MoveTo(gameObject， {“x”:2， “time”:3， “loopType”:“pingPong”， “delay”:1)); |

上面的意思就是三秒的时间将物体运动到x坐标为2的位置，运动的类型为“pingPong”，延时1秒执行这个运动。

在C#中。你可能要这样写才可以:

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

|  |  |
| --- | --- |
| 01 | Hashtable parameters = new Hashtable(); |
| 02 |  |
| 03 | parameters.**Add**(“x”，2); |
| 04 |  |
| 05 | parameters.**Add**(“time”，3); |
| 06 |  |
| 07 | parameters.**Add**(“looptype”，iTween。LoopType。pingPong); |
| 08 |  |
| 09 | parameters.**Add**(“delay”，1); |
| 10 |  |
| 11 | iTween.MoveTo(gameObject， parameters); |

在Js中只要一行代码到C#中变成了六行，有点违背了最小投入最大产出的中心思想，但是，不要怕，iTween有一个解决方案，让C#中也可以像JavaScript那样简洁。

iTween里的hash()方法就可以做到，看下面修改后的代码：

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

|  |  |
| --- | --- |
| 1 | iTween.MoveTo(gameObject， iTween。Hash(“x”，2，“time”，3，“loopType”，“pingPong”，“delay”，1)); |
| 2 |  |

以下是iTween插件各方法的大致用法，每个方法都有简单传参和Hashtable定制两种用法，Hashtable可用iTween。Hash()生成，每个方法所需参数及其功能请参考iTween内部提示。

常用方法简介：

AudioFrom:pitch和volum属性提供的是初始值

audioTo: pitch和volum属性提供的是终结值

audioUpdate:pitch和volum属性提供的是终结值 此方法用于Update()方法中

stab:播放AudioClip一次，不用手动加载AudioSource组件

CameraFadeAdd:创建一个对象可以模拟摄相机的淡入淡出。

CameraFadeSwap：改变摄相机的淡入淡出背景图(对象为CameraFadeAdd返回对象）

CameraFadeFrom:立即改变摄相淡入淡出的透明度然后随时间返回。amount:当执行淡入淡出时，其透明度的变化速度。（透明度越大，淡入淡出越快，个人认为100为满，如果速度较快，时间较长，渐变效果会在时间快要结束时出现。此方法配合CameraFadeAdd使用，只有在CameraFadeAdd前提下，才可以进行淡入淡出操作。此方法为从CameraFadeAdd返回的对象出淡出到原来的界面。

CameraFadeTo：随时间改变摄相机淡入淡出透明度，此方法为从本界面淡入到CameraFadeAdd返回的对象

ColorFrom:即刻改变对象的颜色值然后随着时间改变其回原来的颜色（总的来说，就是对GUIText和GUITexture的颜色的淡入淡出效果）。Color：此属性代表对象变化初始值。与audioFrom有异曲同工之效。

ColorTo：随着时间改变对象的颜色组。同上例一样。Color：此属性代表对象变化最终值，与audioTo有异曲同工之效

（注意，ColorFrom和ColorTo还有后面的ColorUpdate方法的NamedColorValue属性，有一些对象不具有NamedColorValue里的属性，运行时会有提示）

ColorUpdate：跟ColorTo类似，但花销的成本较少，此方法在Update方法中被调用

FadeFrom：即刻改变对象的的阿尔法值，然后随着时间将其阿尔法值变回原值。如果对象上有挂载a Light， GUIText or GUITexture这些组件，这些组件将成为被执行的对象。

注：阿尔法值可以粗略理解为对象的透明度，值越小，透明度越大。这里的 alpha或者 amount 是变化初值

FadeTo：同上，alpha或amount是变化终值。

FadeUpdate ：同FadeTo类似，在Update（）方法中调用，提供时刻改变属性值的环境。不局限于 EaseType

LookFrom：即刻旋转物体让其看提供的Vector3或都Transfrom，然后随时间旋转回原来的角度值

注：物体的脸部一般以本地坐标(即物体坐标)的Z轴，脸部朝向方法，即Z轴指向方法。

LookTo：随时间旋转物体让其脸部朝向所提供的Vector3或Transform位置。

LookUpdate：同LookTo类似，在Update()方法中调用。

MoveAdd：随时间改变游戏对象的位置（原理还有点蒙，感觉跟MoveBy有点像）amount:是改变物体位置的一个值，不是目标位置的坐标。

MoveBy：增加提供的坐标到游戏对象的位置

MoveFrom：立即改变游戏对象的位置为提供的，然后随时间改变游戏对象位置到初始位置

属性：movetopath：Boolean值 ，是否自动将物体置于Ptah的起始点，默认值为真

Path:目标文件可用路径编缉器获得

PunchPosition：对物体的位置添加一个摇晃的力，使其摇晃最终归于原来的位置 其晃动大小和方法由提供的amount(Vector3)决定（方法由Vector3的x，y，z共同决定，晃动大小，由各个方法的值的大小决定）

PunchRotation：对物体的旋转添加一个摇晃的力，使其旋转最终归于初始值。其旋转角度大小和方向由提供的Vector3决定，建议用单轴而不是整个Vector3，例如（0，1，0）是绕Y轴旋转，角度大小由Vector3Y轴值大小决定

PunchScale：对物体的大小比例添加一个摇晃的力，使其大小比例变化最终归于初始值。其大小比例变

化方向和大小由提供的Vector3决定。例如（0，1，0）是在Y轴方向对物体大小变化（即变化物体的高），大小由该方向的值大小决定。

PutOnPath ：根据提供的百分比将游戏物体置于所提供路径上（1为百分之百）。

PointOnPath：根据提供的百分比返回一条路径上的Vector3的位置

RectUpdate：返回一个RECT在提供的两个值之间，大小变化根据提供的速度

Vector3Update:返回一个Vector3在提供的两个值之间，大小变化根据提供的速度

Vector2Update:返回一个Vector3在提供的两个值之间，大小变化根据提供的速度

FloatUpdate:返回一个float在提供的两个值之间，大小变化根据提供的速度

RotateAdd:对游戏物体的旋转角度随着时间增加所提供的欧拉角（顺时针旋转。Vector3三个值解析：

X，Y，Z各代表围绕哪个轴转动。其转动角度就是X，Y，Z、的值的大小。amount:欧拉角大小）

RotateBy：把提供的值乘以360，然后随着时间旋转游戏物体的角度按计算得的值。例 如（0，1，0）就是绕Y轴旋转360度， 顺时针旋转。

RotateFrom：立即改变游戏物体角度的欧拉角，然后随着时间旋转回原来的角度，属性提供的欧拉角为变化初始值

RotateTo：旋转游戏物体角度到我们所提供的欧拉角角度。属性提供的欧拉角为变化终结值

RotateUpdate：跟RotateTo类似，该方法在Update中被调用，提供一个可改变属性值的环境。不用局限EaseType

ScaleAdd：随着时间根据提供的amount(Vector3)增加游戏物体的大小

ScaleBy:随着时间变形游戏物体，游戏物体最终变形大小由我们提供的amount(Vector3)值决定 算法：终变形大小=游戏物体初始的sacle \* 我们提供的 amount值

ScaleFrom：立即改变游戏物体的比例大小，然后随时间返回游戏物体原本的大小。amount:为物体变形的初始大小

ScaleTo：随着时间改变物体的比例大小到我们提供的Scale大小(scale值)

ScaleUpdate:跟ScaleTo类似，此方法用于Update中，提供可改变属性值的环境 ，不用局限于EaeType

ShakePosition：根据提供的amount衰减其值随机摇动游戏物体的位置，其晃动大小和方向由提供的amount(Vector3)决定（方向根据Vector3随机，晃动大小，由各个方向的值的大小决定）

ShakeRotation：根据提供的amount衰减其值随机摆动旋转游戏物体的角度 。Vector3三个值解析：X，Y，Z各代表围绕哪个轴旋转。其转动角度就是X，Y，Z、的值的大小

ShakeScale：根据提供的amount衰减其值随机摆动改变游戏物体的大小。其大小比例变化方向和大小由提供的Vector3决定。例如（0，1，0）是在Y轴方向对物体大小变化（即变化物体的高），大小由该方向的值大小决定

Pause：停止iTween 如果一个iTween以ignoreTimeScale定制且设为True停止工作。 参考不同参数Pause的说明。

Stop：停止iTweens，和Pause一样，不同的参数有不同作用。

Stop By Name:停止指定名字的iTween

ValueTo：返回一个插值在两件值之间的回调函数的值