**Unity3D教程：脚本初级知识（一）**

Posted on 2013年01月29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 407 次

**脚本概览**

这是一个关于Unity内部脚本如何工作的简单概览。

Unity内部的脚本，是通过附加自定义脚本对象到游戏物体组成的。在脚本对象内部不同志的函数被特定的事件调用。最常用的列在下面：

Update：这个函数在渲染一帧之前被调用，这里是大部分游戏行为代码被执行的地方，除了物理代码。

FixedUpdate：这个函数在每个物理时间步被调用一次，这是处理基于物理游戏的地方。

在任何函数之外的代码：

在任何函数之外的代码在物体被加载的时候运行，这个可以用来初始化脚本状态。

注意：文档的这个部份假设你是用Javascript，参考用C#编写获取如何使用C#和Boo编写脚本的信息。

你也能定义事件句柄，它们的名称都以On开始，（例如OnCollisionEnter），为了查看完整的预定义事件的列表，参考MonoBehaviour 文档。

**常用操作**

大多数游戏物体的操作是通过游戏物体的Transform或Rigidbody来做的，在行为脚本内部它们可以分别通过transform和rigidbody访问，因此如果你想绕着Y轴每帧旋转5度，你可以如下写：

function Update(){  
transform.Rotate(0,5,0);  
}

如果你想向前移动一个物体，你应该如下写：

function Update(){  
transform.Translate(0,0,2);  
}

**跟踪时间**

Time类包含了一个非常重要的类变量，称为deltaTime，这个变量包含从上一次调用Update或FixedUpdate（根据你是在Update函数还是在FixedUpdate函数中）到现在的时间量。

所以对于上面的例子，修改它使这个物体以一个恒定的速度旋转而不依赖于帧率：

function Update(){  
transform.Rotate(0,5\*Time.deltaTime,0);  
}

**移动物体：**

function Update(){  
transform. Translate (0, ,0,2\*Time.deltaTime);  
}

如果你加或是减一个每帧改变的值，你应该将它与Time.deltaTime相乘。当你乘以Time.deltaTime时，你实际的表达：我想以10米/秒移动这个物体不是10米/帧。这不仅仅是因为你的游戏将独立于帧而运行，同时也是因为运动的单位容易理解。( 米/秒)另一个例子，如果你想随着时间增加光照的范围。下面的表达式，以2单位/秒改变半径。

function Update (){  
light.range += 2.0 \* Time.deltaTime;  
}

当通过力处理刚体的时候，你通常不必用Time.deltaTime，因为引擎已经为你考虑到了这一点。访问其他组件

组件被附加到游戏物体，附加Renderer到游戏物体使它在场景中渲染，附加一个Camera使它变为相机物体，所有的脚本都是组件，因为它们能被附加到游戏物体。

最常用的组件可以作为简单成员变量访问：

Component 可如下访问

Transform transform

Rigidbody rigidbody

Renderer renderer

Camera camera (only on camera objects)

Light light (only on light objects)

Animation animation

Collider collider

…等等。

对于完整的预定义成员变量的列表。查看Component，Behaviour和MonnoBehaviour类文档。如果游戏物体没有你想取的相同类型的组件，上面的变量将被设置为null。任何附加到一个游戏物体的组件或脚本都可以通过GetComponent访问。

transform.Translate（0,3,0）; //等同于  
GetComponent(Transform).Translate(0, 1, 0);

注意transfom和Transform之间大小写的区别，前者是变量（小写），后者是类或脚本名称（大写）。大小写不同使你能够从类和脚本名中区分变量。应用我们所学，你可以使用GetComponent找到任何附加在同一游戏物体上的脚本和组件，请注意要使用下面的例子能够工作，你需要有一个名为OtherScript的脚本，其中包含一个DoSomething函数。OtherScript脚本必须与下面的脚本附加到相同的物体上。

function Update(){  
otherScript = GetComponent(OtherScript); //这个在同一游戏物体桑找到名为OtherScript的脚本  
otherScript.DoSomething(); //并调用它上加的DoSomething  
}