**Unity3D等待函数介绍**

Posted on 2013年03月19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 291 次

[**Unity3D**](http://www.unitymanual.com)等待函数介绍。Coroutines & Yield是Unity3D编程中重要的概念，它可以实现将一段程序延迟执行或者将其各个部分分布在一个时间段内连续执行，但是在Javascript与C#中实现Coroutines & Yield，在语法上却有一些区别：

yield不可单独使用需要与return配合使用，例如：

yield return 0; //等0帧

yield return 1; //等1帧

yield return WaitForSeconds(3.0); //等待3秒

所有使用yield的函数必须将返回值类型设置为IEnumerator类型，例如：

IEnumerator DoSomeThingInDelay() {...}

最后，也是在”Using C#”这个章节中没有讲到的关键一点是，所有IEnumerator类型函数必须使用”StartCoroutine”这个函数触发，不能单独使用，例如：

StartCoroutine(DoSomeThingInDelay());

最后附上学习Coroutines & Yield时所做的小例子，脚本的作用是不断随机改变材质的颜色。

脚本如下：

using UnityEngine;  
using System.Collections;  
public class RandomColor : MonoBehaviour {

public float delayInSecond = 1;  
public Material targetMaterial;

// Use this for initialization  
void Start () {  
StartCoroutine(AutoChangeColor());  
}  
// Update is called once per frame  
void Update () {  
}

IEnumerator AutoChangeColor()  
{  
yield return 0; //确保Time.deltaTime为0  
Color colorNew = GenerateRandomColor();  
Color colorNow = targetMaterial.GetColor(&quot;\_Color&quot;);  
float timeEclapsed = 0;  
for (timeEclapsed = 0; timeEclapsed &lt; delayInSecond; timeEclapsed += Time.deltaTime)  
{  
float progress = timeEclapsed / delayInSecond;  
Color colorTween = new Color(  
(colorNew.r - colorNow.r) \* progress + colorNow.r,  
(colorNew.g - colorNow.g) \* progress + colorNow.g,  
(colorNew.b - colorNow.b) \* progress + colorNow.b  
);  
targetMaterial.SetColor(&quot;\_Color&quot;, colorTween);  
yield return 1;  
}

StartCoroutine(AutoChangeColor());  
}

Color GenerateRandomColor(){  
Color color = new Color();  
color.r = Random.value;  
color.g = Random.value;  
color.b = Random.value;

return color;  
}  
}