**调用其他GameObject的脚本函数**

Posted on 2013年05月23日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 24 次

调用其他实例的脚本是一种很常用的需求，在 Unity3D 中实现起来也有多种方法，假设在 ScriptB 有一个 DoSomething 函数，如下所示：

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

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **public** **class** ScriptB : MonoBehaviour |
| 04 |  |
| 05 | { |
| 06 |  |
| 07 | **public** **void** DoSomething() |
| 08 |  |
| 09 | { |
| 10 |  |
| 11 | Debug.Log(“Hi there”); |
| 12 |  |
| 13 | } |
| 14 |  |
| 15 | } |
| 16 |  |

在 ScriptA 中如果要访问某个 GameObject 的 ScriptB 脚本中的 DoSomething 函数：

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

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **public** **class** ScriptA : MonoBehaviour |
| 04 |  |
| 05 | { |
| 06 |  |
| 07 | **public** ScriptB other; |
| 08 |  |
| 09 | **void** Update() |
| 10 |  |
| 11 | { |
| 12 |  |
| 13 | other.DoSomething(); |
| 14 |  |
| 15 | } |
| 16 |  |
| 17 | } |
| 18 |  |

注意 ScriptA 有一个 ScriptB 类型的公共变量，这是一个绝妙的小技巧，如此一来你便可以通过编辑器拖拽任何含有 ScriptB 这一脚本的 GameObject至该变量上（因为是 public 的，所以暴露于编辑器界面中），Unity 会自动识别你需要赋值的不是那个 GameObject，而 是它所包含的 ScriptB 脚本。

除此之外，如果你不喜欢这种在编辑器中直接赋值的办法，你可以通过GameObject.Find(“somename”) 来获取GameObject：

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

|  |  |
| --- | --- |
| 1 | GameObject go = GameObject.Find(“somegameobjectname”); |
| 2 |  |
| 3 | ScriptB other = (ScriptB) go.GetComponent(typeof(ScriptB)); |
| 4 |  |
| 5 | other.DoSomething(); |
| 6 |  |

不过需要注意的是 GameObject.Find()的执行效率非常低，所以如果可能的话还是尽可能使用“直接赋值法”吧。