**Unity3D教程：脚本初级知识（二）**

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 346 次

**访问其它游戏物体**

大多数高级的代码不仅需要操作一个物体，Unity脚本接口有各种方法来找到并访问其他游戏物体和组件。在下面，我们假定有个一名为OtherScript,js的脚本附加到场景的游戏物体上。

var foo = 5;  
function DoSomething ( param : String) {  
print(param + " with foo: " + foo);  
}

1.通过检视面板赋值引用你可以通过检视面板赋值变量到任何物体

var target : Transform;  
function Update ()  
{  
target.Translate(0, 1, 0);//变换拖动到target的物体  
}

你也可以在检视面板中公开到其他物体的引用，下面你可以拖动一个包含的游戏物体到检视面板中的target槽。设置在检视面板中赋值的target变量上的foo，调用DoSomething。

var target : OtherScript;  
function Update ()  
{  
target.foo = 2;//设置target物体的foo变量  
target.DoSomething("Hello");// 调用target上的DoSomething  
}

2.通过物体层次定位对于一个已经存在的物体，可以通过游戏物体的Transform组件来找到子和父物体；

transform.Find("Hand").Translate(0, 1, 0);//找到脚本所附加的游戏物体的子“Hand”

一旦在层次视图中找到这个变换，你可以使用GetComponent来获取其他脚本。

transform.Find("Hand").Translate(0, 1, 0); //找到名为“Hand”的子 然后应用一个力到附加在hand上的刚体  
transform.Find("Hand").GetComponent(OtherScript).DoSomething("Hello"); //在附加到它上面的OtherScript中，设置foo为2；  
transform.Find("Hand").rigidbody.AddForce(0, 10, 0);// //变换的所有子向上移动10个单位

你可以循环所有的子。

for (var child : Transform in transform)  
{  
child.Translate(0, 1, 0);  
}

参考Transform类文档获取更多信息。3.根据名称或标签定位.

你可以使用GameObject.FindWithTag和GameObject.FindGameObjectsWithTag搜索具有特定标签的游戏物体，使用GameObject.Find根据名称查找物体。

function Start ()  
{  
var go = GameObject.Find("SomeGuy");// 按照名称  
go.transform.Translate(0, 1, 0);var player = GameObject.FindWithTag("Player");// 按照标签  
player.transform.Translate(0, 1, 0);  
}

你可以在结果上使用GetComponent，在找到的游戏物体上得到任何脚本或组件。

function Start ()  
{  
var go = GameObject.Find("SomeGuy");// 按名称  
go.GetComponent(OtherScript).DoSomething();  
var player = GameObject.FindWithTag("Player");// 按标签  
player.GetComponent(OtherScript).DoSomething();  
}

一些特殊的物体有快捷方式，如主相机使用Camera.main。4.作为参数传递

一些事件消息在事件包含详细信息。例如，触发器事件传递碰撞物体的Collider组件到处理函数。

OnTriggerStay给我们一个到碰撞器的引用。从这个碰撞器我们可以获取附加到其上的刚体。

function OnTriggerStay( other : Collider ) {  
if (other.rigidbody) {// 如果另一个碰撞器也有一个刚体  
other.rigidbody.AddForce(0, 2, 0);// 应用一个力到它上面  
}  
}

或者我们可以通过碰撞器获取附加在同一个物体上的任何组件。

function OnTriggerStay( other : Collider ) {  
if (other.GetComponent(OtherScript)) {// 如果另一个碰撞器附加了OtherScript  
other.GetComponent(OtherScript).DoSomething();// 调用它上面的DoSomething  
// 大多数时候碰撞器不会附加脚本  
// 所以我们需要首先检查以避免null引用异常  
}  
}

注意通过上述例子中的other变量，你可以访问碰撞物体中的任何组件。5.一种类型的所有脚本

使用Object.FindObjectsOfType找到所有具有相同类或脚本名称的物体，或者使用Object.FindObjectOfType.找到这个类型的第一个物体。

function Start ()  
{  
var other : OtherScript = FindObjectOfType(OtherScript); // 找到场景中附加了OtherScript的任意一个游戏物体  
other.DoSomething();  
}

向量Unity使用Vector3类同一表示全体3D向量，3D向量的不同组件可以通过想x，y和z成员变量访问。

var aPosition : Vector3;  
aPosition.x = 1;  
aPosition.y = 1;  
aPosition.z = 1;

你也能够使用Vector3构造函数来同时初始化所有组件。

var aPosition = Vector3(1, 1, 1);

Vector3也定义了一些常用的变量值。

var direction = Vector3.up; // 与 Vector3(0, 1, 0);相同

单个向量上的操作可以使用下面的方式访问：

someVector.Normalize();

使用多个向量的操作可以使用Vector3类的数；

theDistance = Vector3.Distance(oneVector, otherVector);

（注意你必须在函数名之前写Vector3来告诉JavaScript在哪里找到这个函数，这适用于所有类函数）你也可以使用普通数学操作来操纵向量。

combined = vector1 + vector2;

查看Vector3类文档获取完整操纵和可用属性的列表。