<http://blog.csdn.net/u011484013/article/details/51487936>

**概述**

    Unity提供的消息推送机制可以非常方便我们的脚本开发，它实现的是一种伪监听者模式，利用的是反射机制。

**常用函数**

    关于消息推送，常用的函数有三个：”SendMessage“、”SendMessageUpwards“、”BroadcastMessage“。这些函数都是GameObject或者Component的成员函数，只要得到一个对象，然后调用它的这三个函数就可以进行一个消息的推送。

**1.SendMessage**

    原型：public void SendMessage(string methodName, object value = null, SendMessageOptions options = SendMessageOptions.RequireReceiver);

    作用：调用一个对象的methodName函数，这个函数可以是公有的也可以是私有的，后面跟一个可选参数（此参数作为传入参数），最后面跟一个可选的设置参数（它有两个选项，后面再讲）。

**2.SendMessageUpwards**

    原型：public void SendMessageUpwards(string methodName, object value = null, SendMessageOptions options = SendMessageOptions.RequireReceiver);

    作用：它的作用和SendMessage类似，只不过它不仅会向当前对象推送一个消息，也会向这个对象的父对象推送这个消息（记住，是会遍历所有父对象）。

**3.BroadcastMessage**

    原型：public void BroadcastMessage(string methodName, object parameter = null, SendMessageOptions options = SendMessageOptions.RequireReceiver);

作用：这个函数的作用和SendMessageUpwards的作用正好相反，它不是推送消息给父对象，而是推送消息给所有的子对象，当然，也是会遍历所有的子对象。

void Awake()

{

gameObject.SendMessage("OnClick", true, SendMessageOptions.DontRequireReceiver);

}

void OnClick(bool flag)

{

Debug.Log("flag = " + flag);

}



但是如果传的参数为null就会出错，因为OnClick的参数为bool

void Awake()

{

gameObject.SendMessage("OnClick", null, SendMessageOptions.DontRequireReceiver);

}

