开发环境

Window7

[Unity3D](http://www.unitymanual.com/)3.4

MB525defy     Android 2.2.1

       Stray满级85了！！赶上月卡Over之前达成目标，这样羽化的约定也算完成了，现在开始好好学习，以后可能会很忙~ ~ 这周最开心的事莫过于老哥来买了3DS，裸眼3D效果不错，但看久也就没什么新意了，上面的火焰纹章着实让人很兴奋，还有小小太阳和北欧女神这种大作，任天堂虽然画面次了点，游戏性倒是真正满点，里面充满了无数创意和灵感，能有机会在游戏机平台上开发游戏就好了~~哈哈，现在羽化一个10平方米不到的房间里面放了5个游戏机，没时间玩啊。。。怨念中- - 话说最近遇到很多不错的人，各取所需大家一起进步也不错~ ~

       好了开始今天的博文，主要说的是Unity保存成Xml，这原本是我找的一篇资料，但羽化发现无法运行在手机平台，所以就改动了下，代码本身不是很难，所以往下看- -

本次学习：

1.   Unity数据存储XML

2.   简单研究下Unity中JS和C#的互调

**1.Unity数据存储XML**

     先把代码送上XML.js

<font color="#333333"><font face="Arial">import System;import System.Xml;import System.Xml.Serialization;import System.IO;import System.Text; //约束条件class TestData{ var Test1 : String; var Test2 : String; var Test3 : float; var Test4 : int;} //数组约束class UserData{ public var \_iUser : TestData = new TestData(); //function UserData() { }} private var X = "NONE"; private var c1 : String;private var c2 : String;private var c3 : float;private var c4 : int;private var \_FileLocation : String;private var \_FileName : String = "TestData.xml";private var \_data; var myData : UserData[];private var tempData : UserData = new UserData(); var i : int = 0; var ShowData : int = 0; function Awake(){ \_FileLocation = Application.persistentDataPath;} function Start(){ FirstSave();} //实现一个 TextWriter，使其以一种特定的编码向流中写入字符。 var streamWriter : StreamWriter; function FirstSave(){//初始化XML tempData.\_iUser.Test1 = "?"; tempData.\_iUser.Test2 = "?"; tempData.\_iUser.Test3 = 0; tempData.\_iUser.Test4 = 0; //FileInfo 方法在您创建或打开文件时返回其他 I/O 类型 var fileInfo : FileInfo = new FileInfo(\_FileLocation+"/"+ \_FileName); if(!fileInfo.Exists) { streamWriter = fileInfo.CreateText(); //XML化 \_data = SerializeObject(tempData); for(i=0;i<10;i++) { streamWriter.WriteLine(\_data); } X = "FSave"; streamWriter.Close(); } else { X = "Saved"; }} function Save(sc1 : String, sc2 : String, sc3 : float, sc4 : int){//保存数据到指定的XMl里 tempData.\_iUser.Test1 = sc1; tempData.\_iUser.Test2 = sc2; tempData.\_iUser.Test3 = sc3; tempData.\_iUser.Test4 = sc4; //FileInfo 方法在您创建或打开文件时返回其他 I/O 类型 var fileInfo : FileInfo = new FileInfo(\_FileLocation+"/"+ \_FileName); var streamWriter : StreamWriter; //fileInfo.Delete(); streamWriter = fileInfo.CreateText(); \_data = SerializeObject(tempData); for(i=0;i<10;i++) { streamWriter.WriteLine(\_data); } X = "Saved"; streamWriter.Close();} function Load(){//读取保存在XML里的数据 var streamReader : StreamReader = File.OpenText(\_FileLocation+"/"+ \_FileName); for(i=0;i<10;i++) { \_data = streamReader.ReadLine(); myData = DeserializeObject(\_data); } streamReader.Close();} var guiSkin : GUISkin; function OnGUI() { GUI.skin = guiSkin; if(GUI.Button(Rect(10, 160, 150, 50),"save")) { Save("Test1","测试2",1.23,50);//要显示中文需设定中文字体 } if(GUI.Button(Rect(160, 160, 150, 50),"load")) { Load(); ShowData = 1; } if(ShowData == 1) { GUI.Label(Rect(10,210,50,25),myData[0].\_iUser.Test1); GUI.Label(Rect(60,210,50,25),myData[0].\_iUser.Test2); GUI.Label(Rect(110,210,50,25),myData[0].\_iUser.Test3 + ""); GUI.Label(Rect(160,210,50,25),myData[0].\_iUser.Test4 + ""); GUI.Label(Rect(10,210+25\*1,50,25),myData[1].\_iUser.Test1); GUI.Label(Rect(60,210+25\*2,50,25),myData[2].\_iUser.Test2); GUI.Label(Rect(110,210+25\*3,50,25),myData[3].\_iUser.Test3 + ""); GUI.Label(Rect(160,210+25\*4,50,25),myData[4].\_iUser.Test4 + ""); } GUI.Label(Rect(170,170+53\*0,150,50),X + "");} function UTF8ByteArrayToString(characters : byte[] ){ var encoding : UTF8Encoding = new UTF8Encoding(); var constructedString : String = encoding.GetString(characters); return (constructedString);} //byte[] StringToUTF8ByteArray(string pXmlString)function StringToUTF8ByteArray(pXmlString : String){ var encoding : UTF8Encoding = new UTF8Encoding(); var byteArray : byte[] = encoding.GetBytes(pXmlString); return byteArray;} // Here we serialize our UserData object of myData//string SerializeObject(object pObject)function SerializeObject(pObject : Object){ var XmlizedString : String = null; //创建其支持存储区为内存的流。 var memoryStream : MemoryStream = new MemoryStream(); //将对象序列化到 XML 文档中和从 XML 文档中反序列化对象。 var xs : XmlSerializer = new XmlSerializer(typeof(UserData)); //表示提供快速、非缓存、只进方法的编写器，该方法生成包含 XML 数据的流或文件 var xmlTextWriter : XmlTextWriter = new XmlTextWriter(memoryStream, Encoding.UTF8); xs.Serialize(xmlTextWriter, pObject); memoryStream = xmlTextWriter.BaseStream; // (MemoryStream) XmlizedString = UTF8ByteArrayToString(memoryStream.ToArray()); return XmlizedString;} // Here we deserialize it back into its original form//object DeserializeObject(string pXmlizedString)function DeserializeObject(pXmlizedString : String){ var xs : XmlSerializer = new XmlSerializer(typeof(UserData)); var memoryStream : MemoryStream = new MemoryStream(StringToUTF8ByteArray(pXmlizedString)); var xmlTextWriter : XmlTextWriter = new XmlTextWriter(memoryStream, Encoding.UTF8); return xs.Deserialize(memoryStream);}</font></font>

这只能说是个引子，深入研究可以考虑下保存成数据库形式，羽化在上面都有小点备注，由于这个转成XML的方法是.Net平台的，所以羽化也看不懂，可以在微软开发网站里面去自己查看。详细例子在最后有下载地址。

**2.简单研究下Unity中JS和C#的互调**

羽化最近在做无缝地图的工作任务时想把地图块保存在一个二维数组中，发现js写二维数组貌似有点不给力，所以看了下网上的教程，一般调用貌似都会失败，所以这里贴一个网上的代码，首先羽化要说感谢原作者的分享，其次羽化发现在手机平台上貌似不能用这种方法搞定js和C#的互调,只能用这方法是C#调用js的方法，但群里面有人指出可以改变文件摆放位置实现手机上JS与C#互调，有兴趣的同学可以研究下~ ~ 这里在提供一个万能的方法就是GameObject.SendMessage ,可以实现不同脚本方法对调-0-效率如何还有待考察。。。

       到今天羽化开发中遇到这类问题得到一个结论：PC上无论怎么调用都不会出现问题，而Android上只能单向调用，如果你想CS调用JS，那么JS脚本一定是要在CS脚本之前运行，实现方法就是改变脚本位置，尽量避免使用SendMessage，虽然很方便，但代价太高。。。

test1.js

function OnGUI(){ if(GUI.Button(Rect(25,25,100,30),"JS Call CS" )){var c = gameObject.GetComponent("test2");c.PrintTest();}}function testPrint(){print("CS Call JS");}

# test2.cs

using UnityEngine;using System.Collections;public class test2: MonoBehaviour {void OnGUI(){if(GUI.Button(new Rect(25,70,100,30), "CS Call JS")){test1 c = (test1)gameObject.GetComponent("test1");c.testPrint();}}void PrintTest(){print("JS Call CS");}}

//原作者说明

這里必須要注意的是JS文件必須是在 "StandardAssets"、 "Pro StandardAssets“和 "Plugins"這三個目錄中的任何一個里，而CS文件不能與JS文件在一個目錄中。原因是，這三個目錄里的腳本被最先編譯，"Editor"目錄里的稍后編譯，其他的腳本最后編譯。目前Unity3D的2.5的版本似乎不支持C# 3.0，所以無法用var的關鍵字，這樣就只支持強類型，所以如果在一個目錄下則CS文件無法讀取JS里的方法，也就無法編譯通過了。而JS調用CS方法則無此限制。

看来原作者是个台湾人，其实羽化发现台湾论坛里面Unity高手很多 -0-

群里经常有人问开始的Unity图标和软件名称在哪修改，其实在Player Settings的Splash Image和上面的Product Name里面修改。

这里送上之前整合写的个例子，一些小测试希望对大家有用。