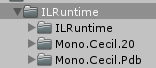
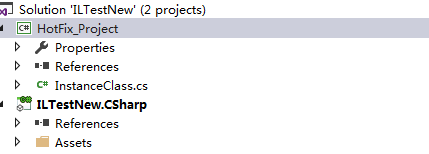
1. 下载群共享的Unity工程
2. 建立Unity空项目
3. 

将群共享的Unity工程下的上面几个文件拷到自己建的Unity空项目Asset目录下，smcs StreamingAssets这两个文件也要拷到Asset目录下

* <https://github.com/Ourpalm/ILRuntime> 从这里下载 最新的zip 文件，拷贝最新的Mono.Cecil.20、 Mono.Cecil.Pdb、 ILRuntime 这三个文件到自己建的Unity工程目录进行覆盖，同时删除ILRuntime文件下的Properties 文件就不会报错了

1. 可以向群主那样建立一个C#脚本 进行演示调用就是用群主那个ILRuntimeController.cs文件
2. 修改dll文件的项目工程，先将IL工程引入，也可以自己新建，我这里方便省事直接用群主的那个HotFix\_Project 工程 引用后如图：



编辑InstanceClass文件，如下：

public InstanceClass()

{

UnityEngine.Debug.Log("My test hello world !!! InstanceClass::InstanceClass()");

this.id = 2;

}

public InstanceClass(int id)

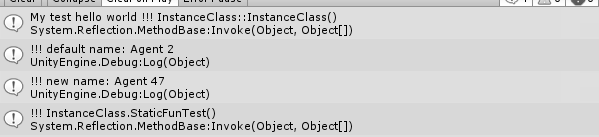
{

UnityEngine.Debug.Log("My test hello world !!! InstanceClass::InstanceClass() id = " + id);

this.id = id;

}

1. 在VS中右键HotFix\_Project工程在菜单中选择Build选项，这样就生成了新的dll文件和pdb文件，现在可以去收获果实了，看看是不是能够从dll文件里写代码
2. 编译生成的dll 文件目录 是在 项目名/bin/Debug/ 有两个一个是dll 一个是pdb
3. 将上面两个文件拷贝到工程中的StreamingAssets文件夹下覆盖，运行Unity 就成功了如下图：



上面就是一个简单的入门教程，引入热更新IL 要结合自己游戏的框架，QQ群：512079820 群主热心，有什么问题直接在群里@群主就可以了

注释：

生成的Dll：解析库

生成的pdb:符号解析库

代码解释1.：先要加载上面的两个文件 才能执行里面的代码

\_app = new AppDomain();

string dllPath = Application.streamingAssetsPath + "/HotFix\_Project.dll";

string pdbPath = Application.streamingAssetsPath + "/HotFix\_Project.pdb";

Stream dllStream = null;

Stream pdbStream = null;

try

{

// load only in editor only, otherwise is your concern :)

dllStream = new FileStream(dllPath, FileMode.Open);

pdbStream = new FileStream(pdbPath, FileMode.Open);

\_app.LoadAssembly(dllStream, pdbStream, new Mono.Cecil.Pdb.PdbReaderProvider());

// register for delegate

\_app.DelegateManager.RegisterMethodDelegate<System.Object>();

\_app.DelegateManager.RegisterMethodDelegate<ILRuntime.Runtime.Intepreter.ILTypeInstance>();

\_app.DebugService.StartDebugService(56000);

代码解释2.调用构造函数

先要获取这个类的类型：IType instanceClass = \_app.GetType("HotFix\_Project.InstanceClass");

ILTypeInstance obj = ((ILType)instanceClass).Instantiate(true); 获取这个类的一个对象

代码解释3.调用类里的一个方法

先获取这个类类型，实例化这个类的一个对象（上面的构造函数看代码解释2），然后

获取这个方法如下调用：

IMethod getNameFun = instanceClass.GetMethod("GetName", 0);

调用这个方法：

string defaultName = \_app.Invoke(getNameFun, obj) as string;

代码解释4.调用类里的一个静态方法

IType instanceClass = \_app.GetType("HotFix\_Project.InstanceClass");

IMethod method = instanceClass.GetMethod("StaticFunTest", 0);

\_app.Invoke(method, null);