1. 安装luaforwindows，安装目录下ite编辑工具

xlua原理：<https://www.cnblogs.com/iwiniwin/archive/2021/09/24/15323970.>html

1. lua语法
2. 删除汉字要删除两次
3. 保存文件时后缀为lua.txt
4. table的键开始为1而不是0
5. luainterface.dll在c#中调用lua代码
6. lua.dofile()lua文件编码格式为ansi
7. 用从c#写好lua的解析，就可以直接直接更新lua脚本来更新逻辑
8. luanet.dll用来在lua中调用c#方法，需要放到输出目录bin
9. lua中访问c#方法使用：来访问，方法带有out参数时不用传参数，ref需要传参，作为返回值返回，当方法返回值为空时，lua也会返回nil，所以要定义个变量来接受空返回值
10. Xlua环境配置，设置自动识别hotfix，luacallcsharp，csharpcalllua
11. Xlua热更：添加宏命令HOTFIX\_ENABLE，添加hotfix特性；generate code生成代码，会在gen目录下生成DelegateGensBridge代码中生成相关函数的代理；然后执行Hotfix inject in Editor进行il代码的注入，调用mono.cecil，会对生成的IL代码进行修改，相当于修改了打上特性标签的类的方法实现；所以如果修改了代码，重新编译后更改的il代码会恢复，所以要再次生成代码
12. Lua和c#的交互原理，lua本身是由c语言编写，可以通过c/c++为桥梁，数据交互通过栈的方式
13. 添加CSharpCallLua特性并生成代码，一般用在委托，在DelegateGensBridge生成对应的方法，将lua的table映射到c#的interface是引用类型映射
14. 将lua的table映射到c#中实现的luaTable类应用类型映射
15. 直接将lua的table映射到class、struct、dictionary、list是值类型拷贝，只拷贝相应的字段
16. Lua部分的映射由单独的类统一管理初始化，并保存，降低映射的开销
17. LuaCallCSharp特性，用来lua调用c#代码，常见加在类上，会生成代码wrap包，local Test = CS.MyExamples.Test;可以用来在lua中持有需要的类，方便后续使用，注意是类，需要对象的话local test = Test();即可
18. 处理委托和事件的方式不同，委托使用+-，event使用方法调用的方式添加和移除，委托可以直接执行testobj.TestDelegate('hello')，事件需要添加调用的方法testobj:CallEvent()
19. Lua文件加载常用loader的方式，xlua提供如下的接口，在使用require时会执行注册的loader代码，将加载到的lua代码返回

public delegate byte[] CustomLoader(ref string filepath);

public void LuaEnv.AddLoader(CustomLoader loader)

1. 使用luabehaviour来模拟monocomponent的生命周期函数
2. Xlua中使用textasset来管理lua脚本资源，require的方式直接加载lua脚本，需要放在resources文件夹下，也可以自定义loader来加载，常用