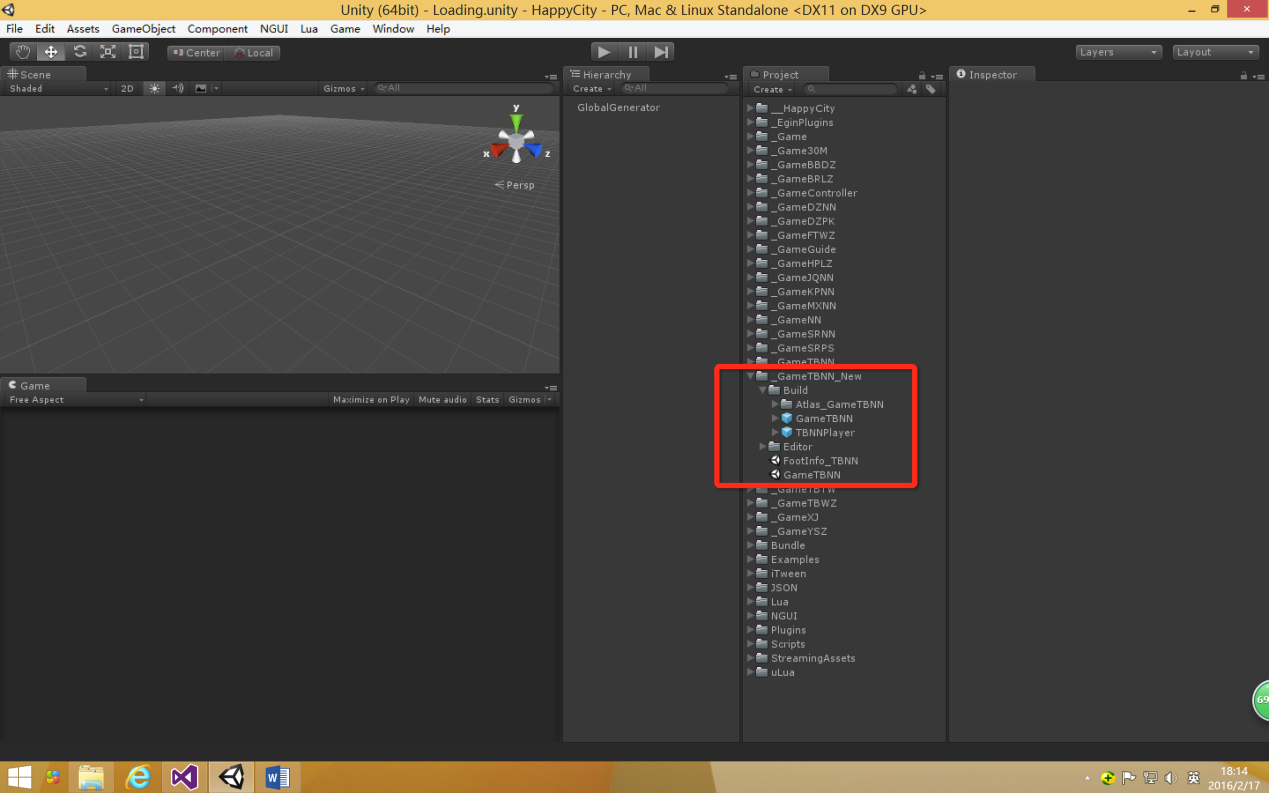
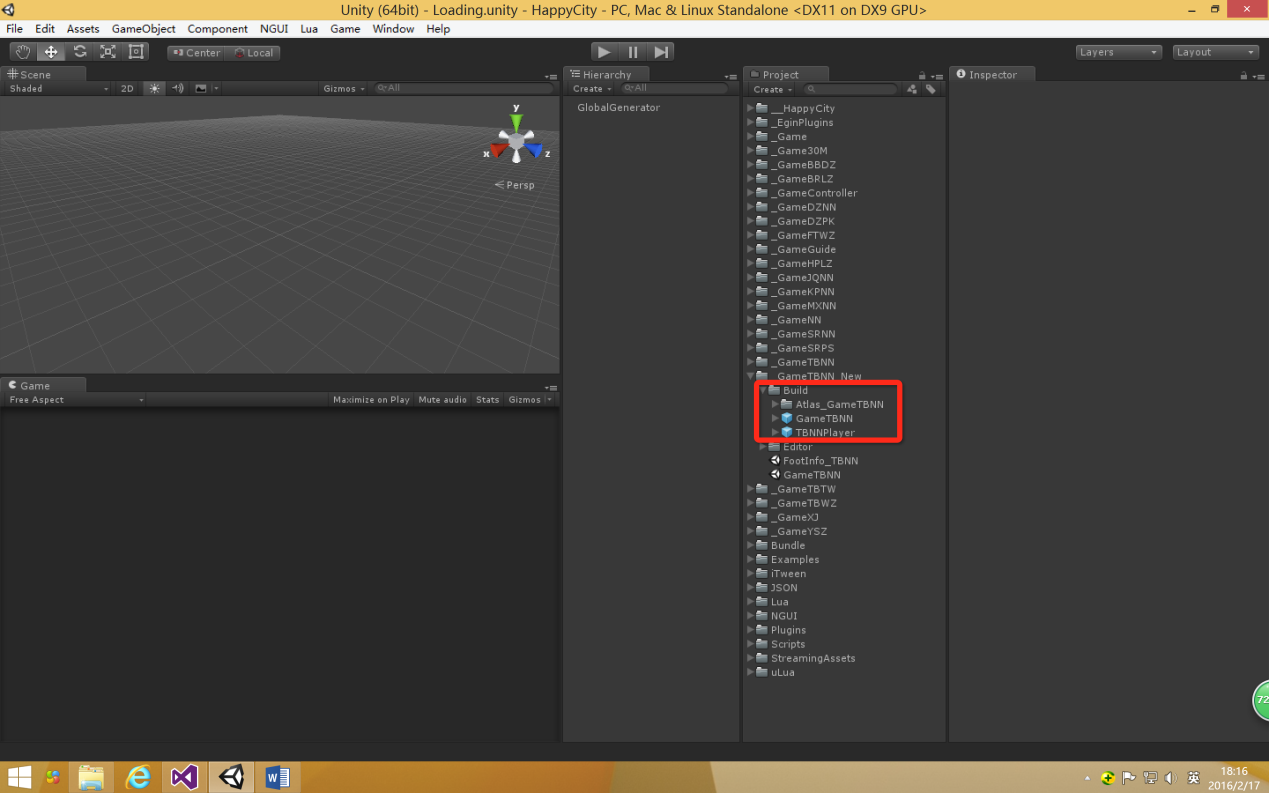
# 添加新的游戏模块

声明: 文档以\_GameTBNN\_New 项目为例

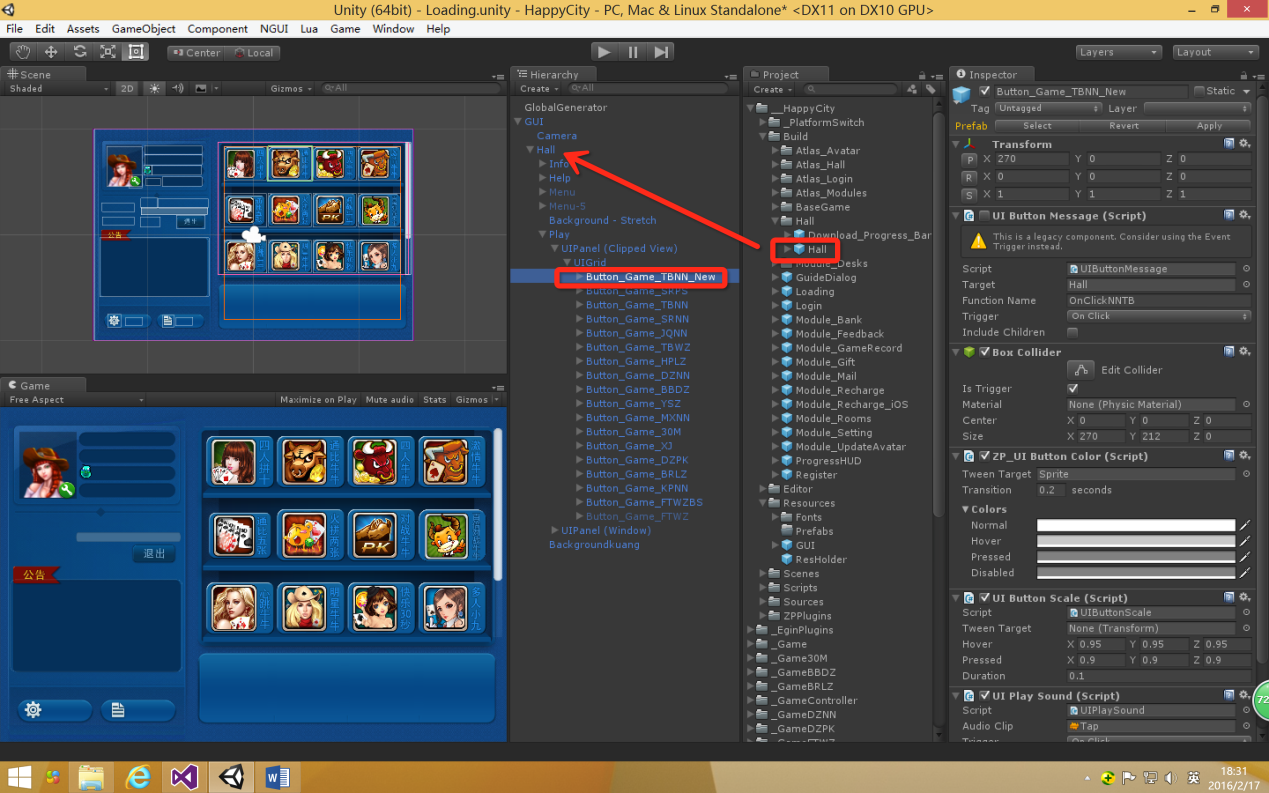
1.新建(或导入)项目(或模块)到Unity



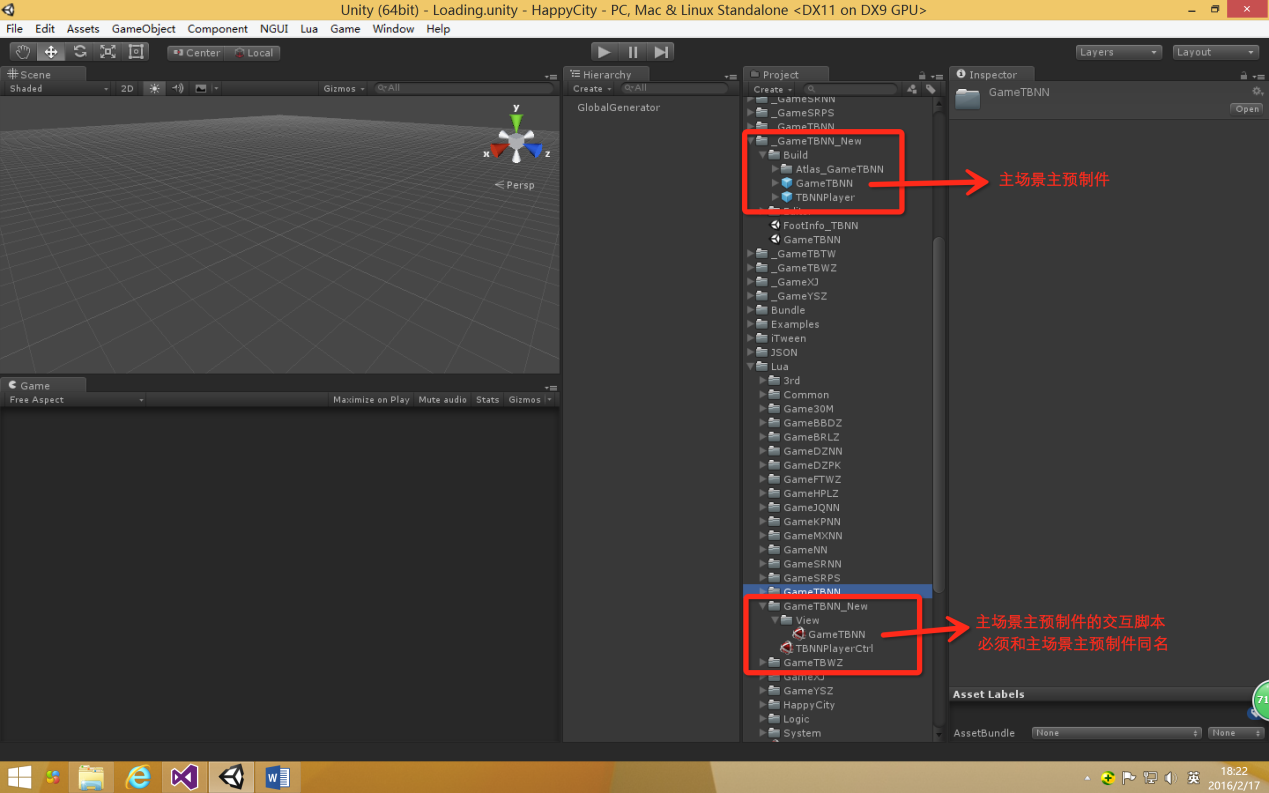
2.在新游戏模块目录下创建 Build文件夹,并把所有需要打包的资源放到该目录下



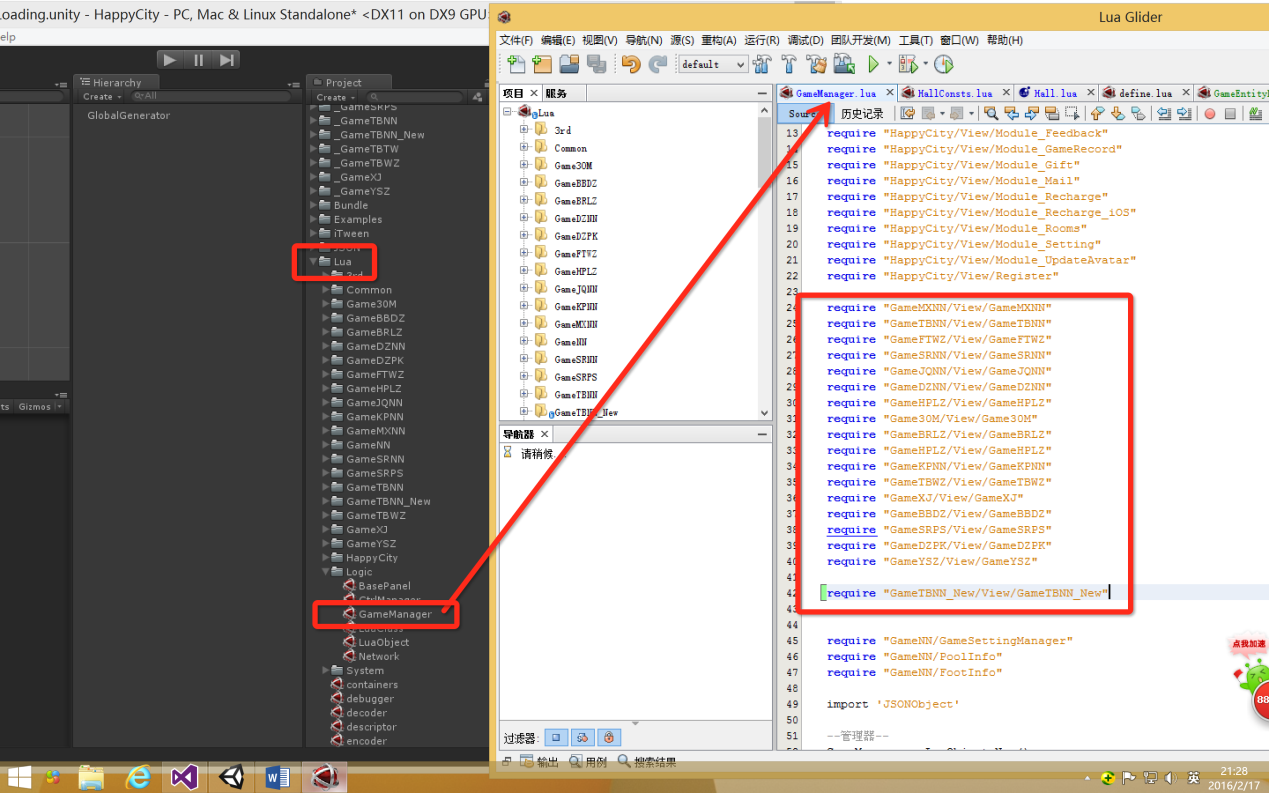
3.在大厅中添加游戏icon



4.在lua中创建该游戏模块的目录结构,同时保证主场景的主预制件的交互脚本和该预制件同名. 注意目录和文件名不能以下划线开头

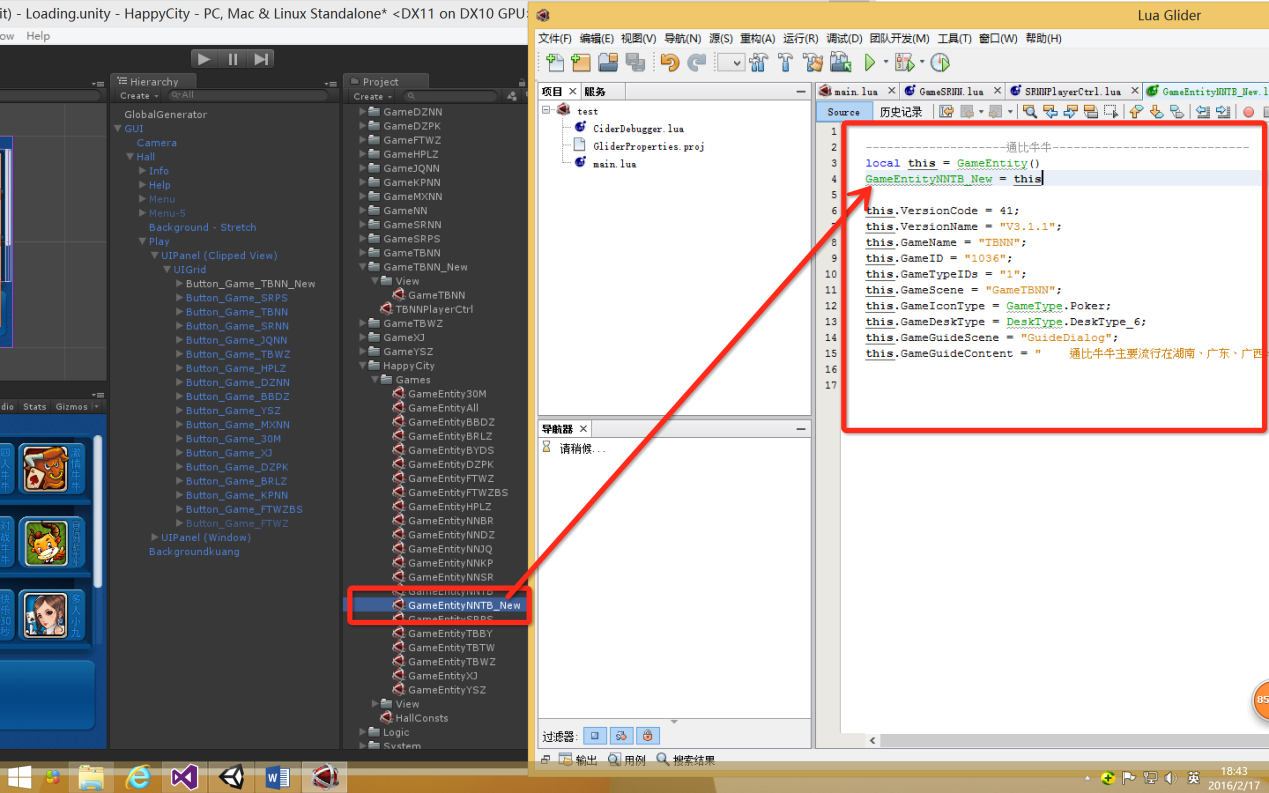


5.在Lua/logic/gamenmanager目录下添加对主预制件交互脚本的引用

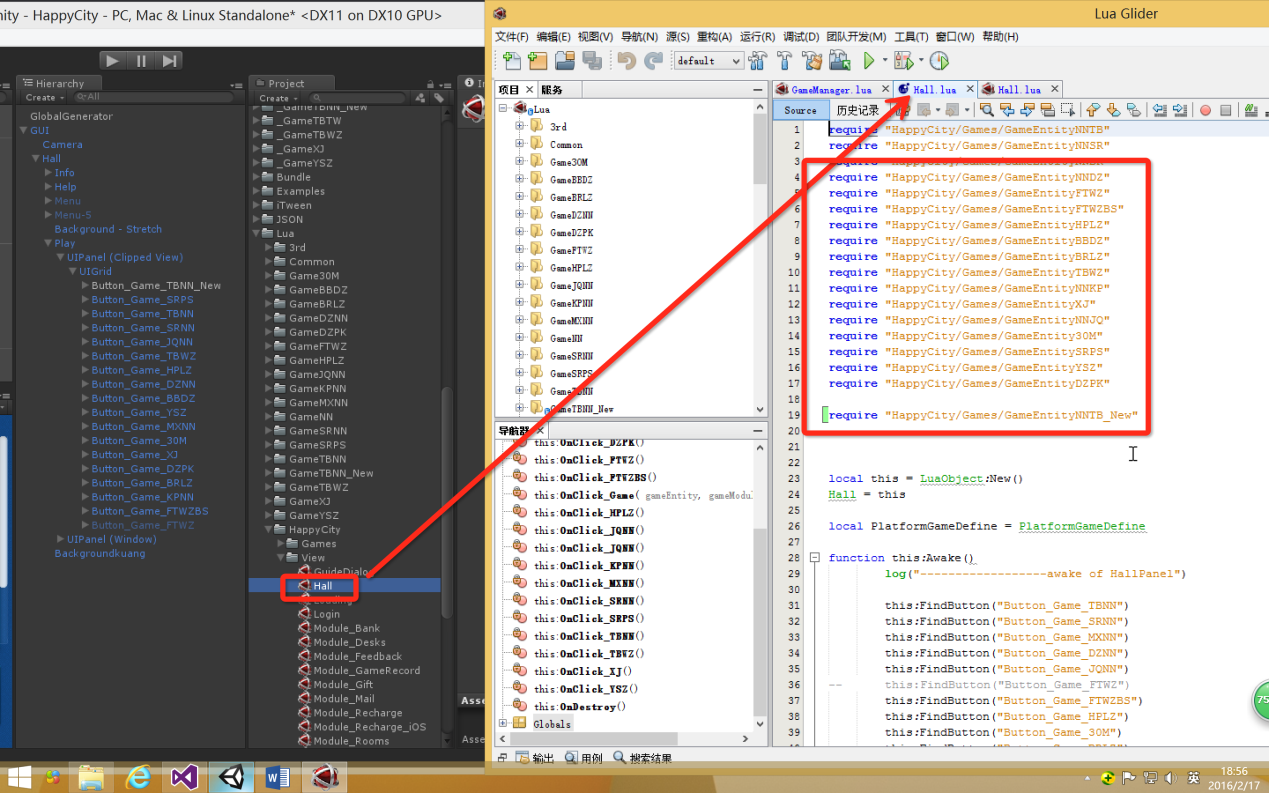


6.在lua的HappyCity中添加该游戏模块的GameEntityxxx.lua文件,并完成该文件的编码

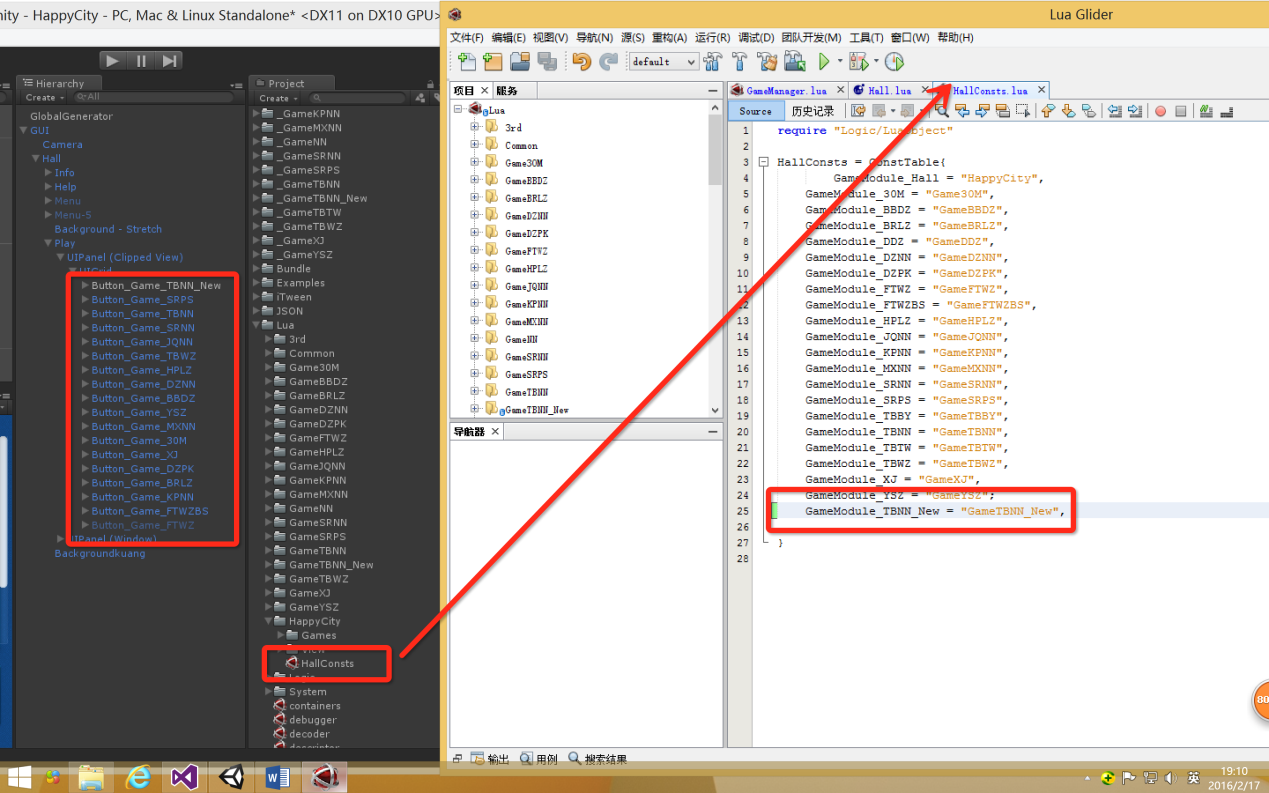
注意: GameScene的值要跟主场景的主预制件的名称保持一致



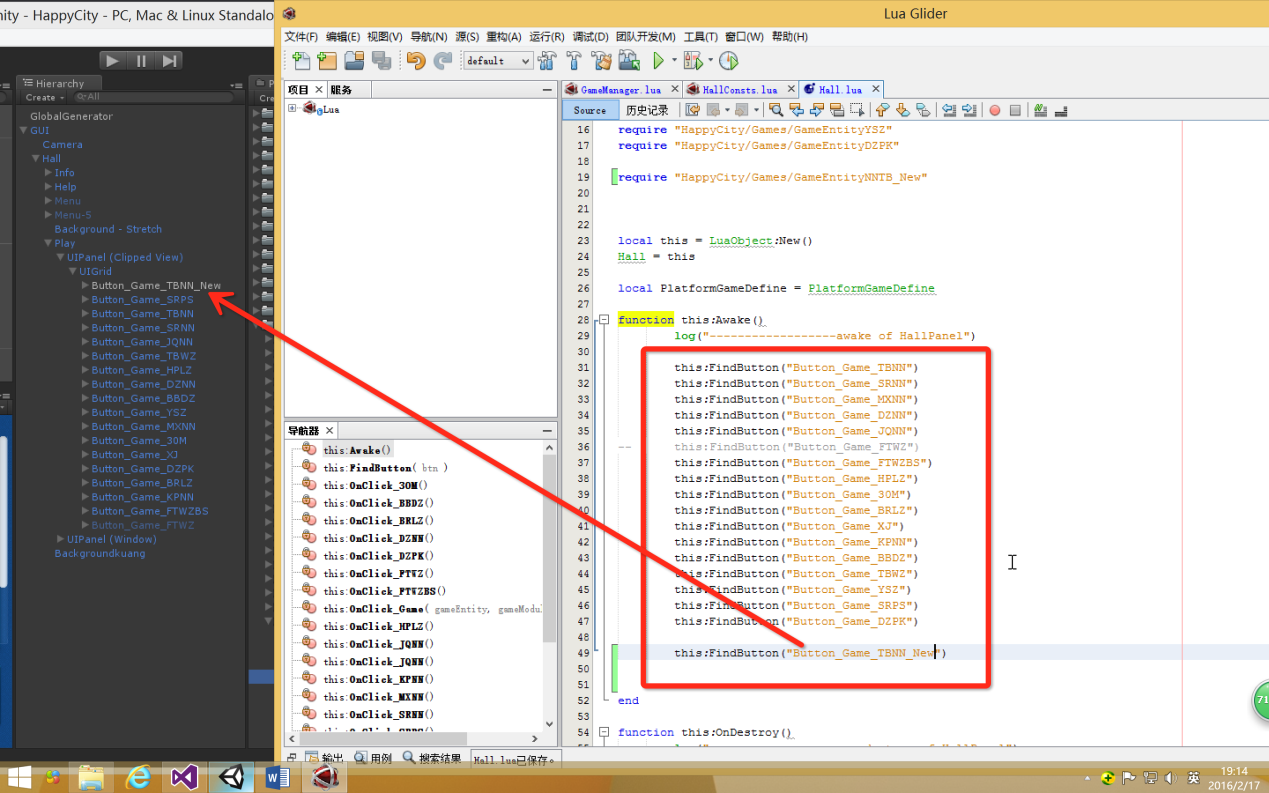
7.在Lua/HappyCity/View/Hall 中添加对 GameEntityxxx 的引用



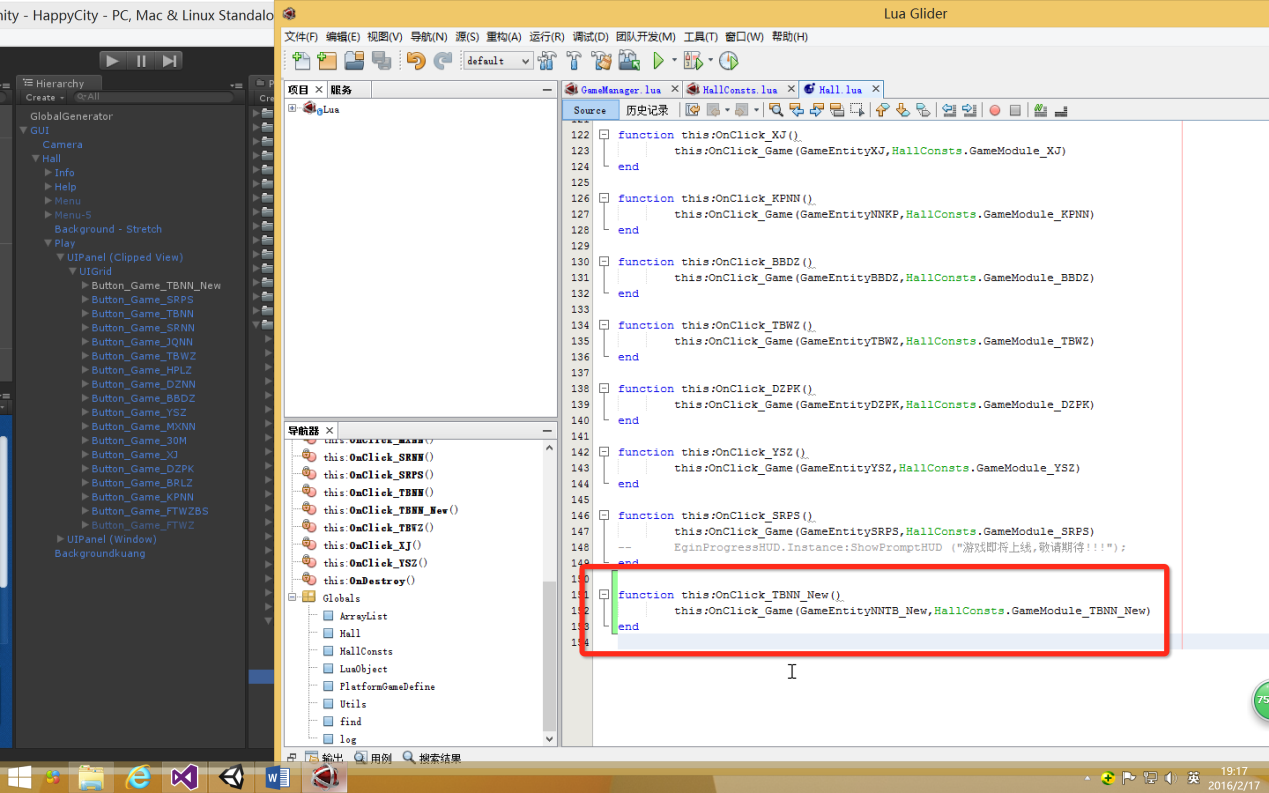
8.在Lua/HappyCity/HallConstant 中添加该模块,注意名称必须与模块名称一致(如果该模块只是一个已有模块的变种,那么保证该值与按钮的名称对应. 参考FTWZ和 FTWZBS)



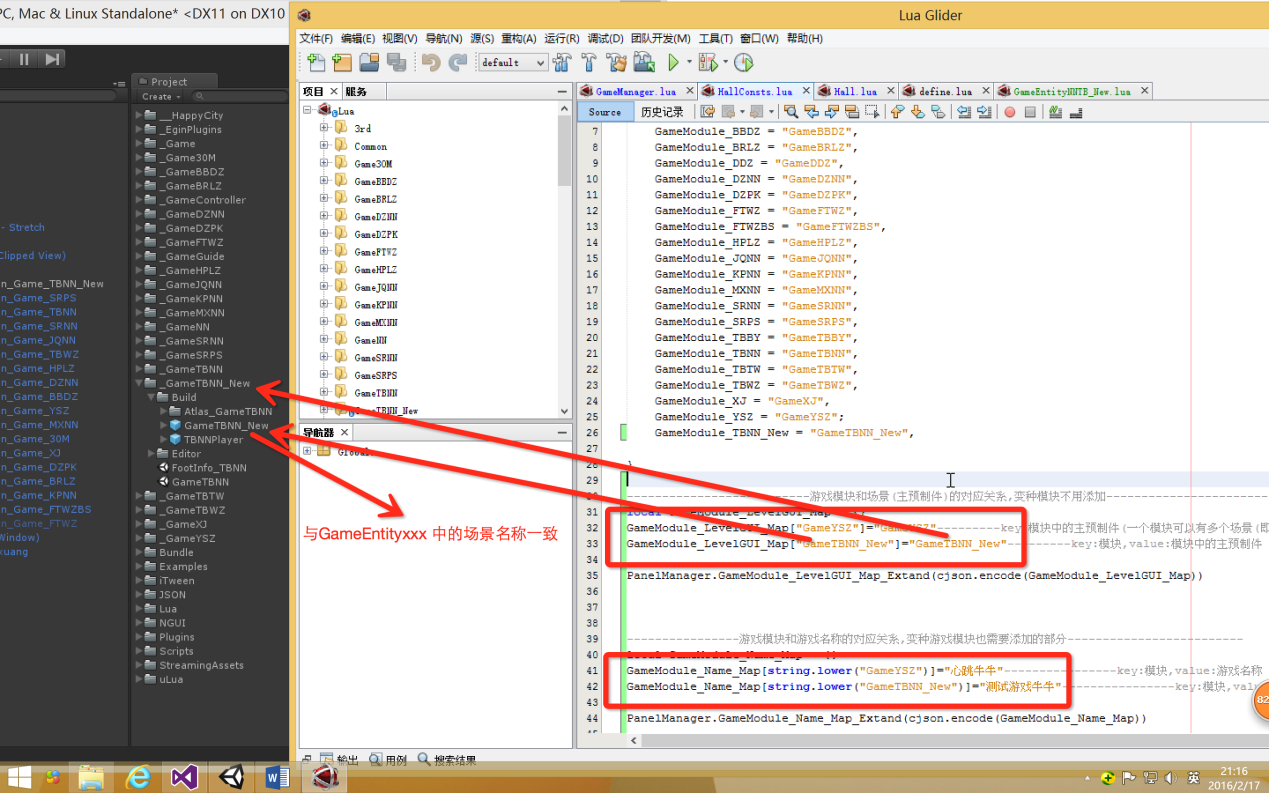
9.在Lua/HappyCity/View/Hall中查找icon按钮,并添加点击方法



添加点击方法. 注意: OnClick\_Game方法的第三个参数为变种模块的名称,没有可以不填. (具体参考FTWZBS)



10.添加游戏模块与场景的对应关系(上面的方框),同时添加游戏模块与名称的对应关系(下面的方框)



11.重新生成luawrap,重新打包资源文件,运行测试

添加新模块 完

添加新模块[升级修改内容]

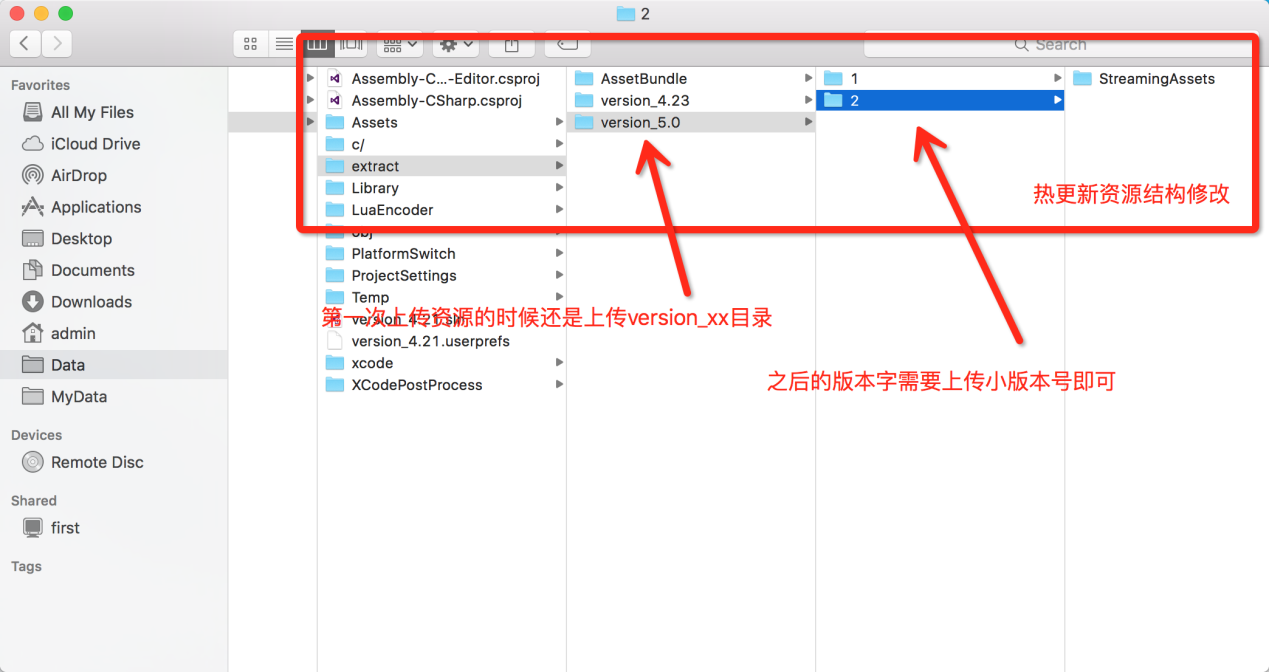
1. 大厅中添加按钮引用的资源模块(注意不要交叉引用)



1. 大厅中按钮点击相关代码修改



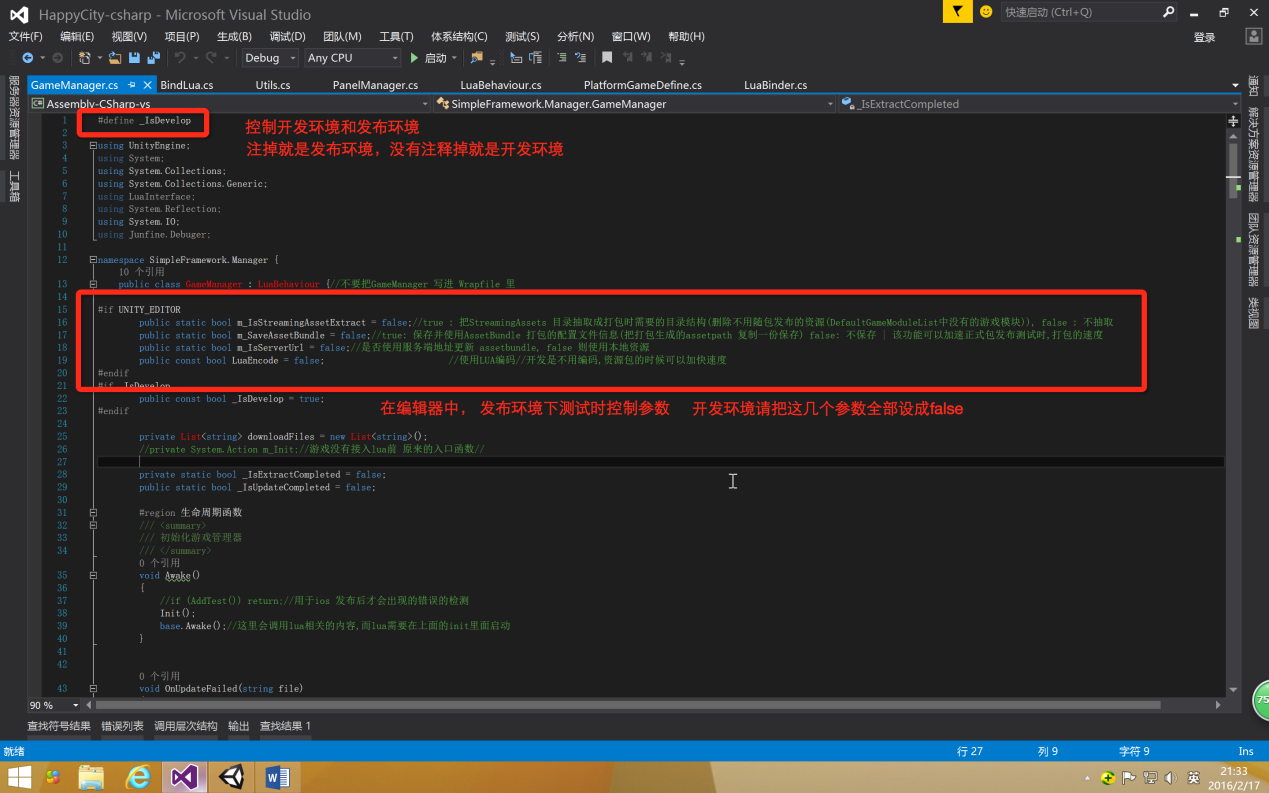
1. 热更新资源结构修改



添加新模块[升级修改内容]完

# 开发环境和发布环境的配置与测试

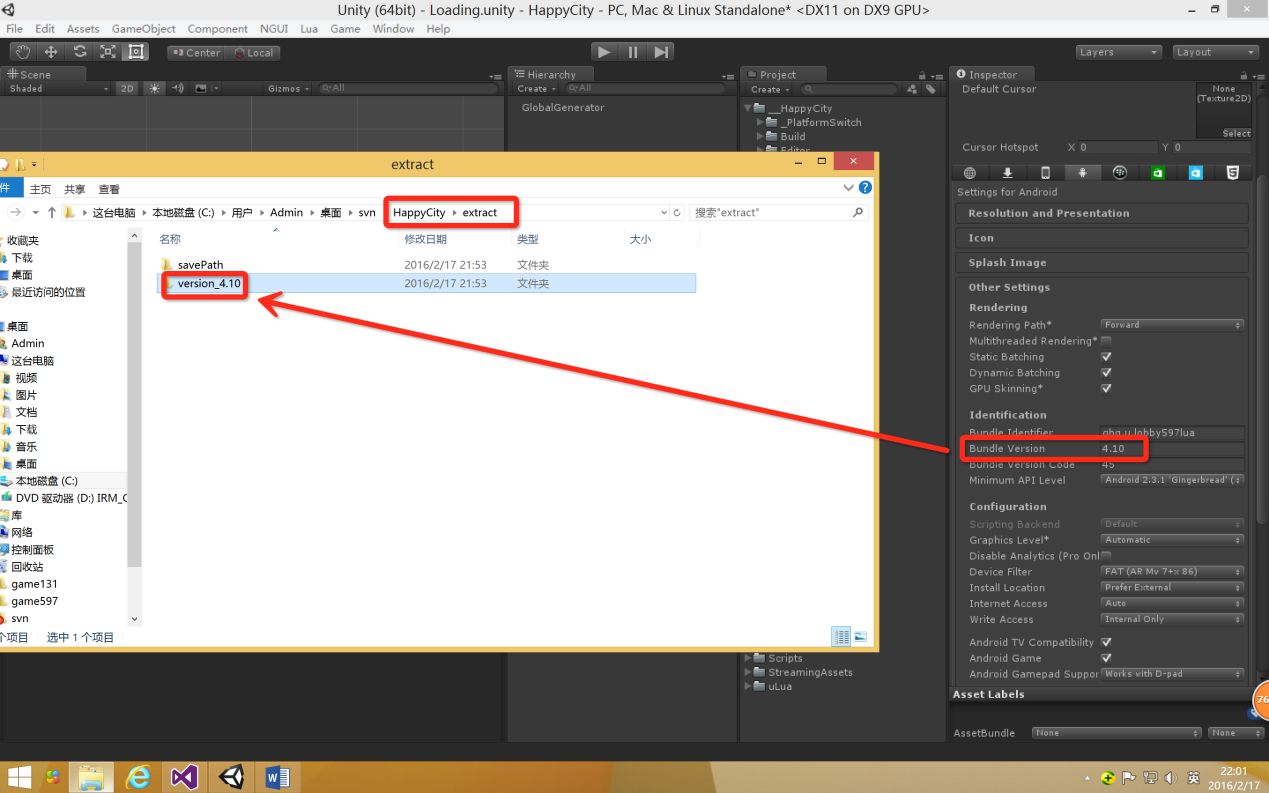
Gamemanager.cs 对整体环境的控制. 注意: 不要提交GameManager.cs 和 Packager.cs 文件



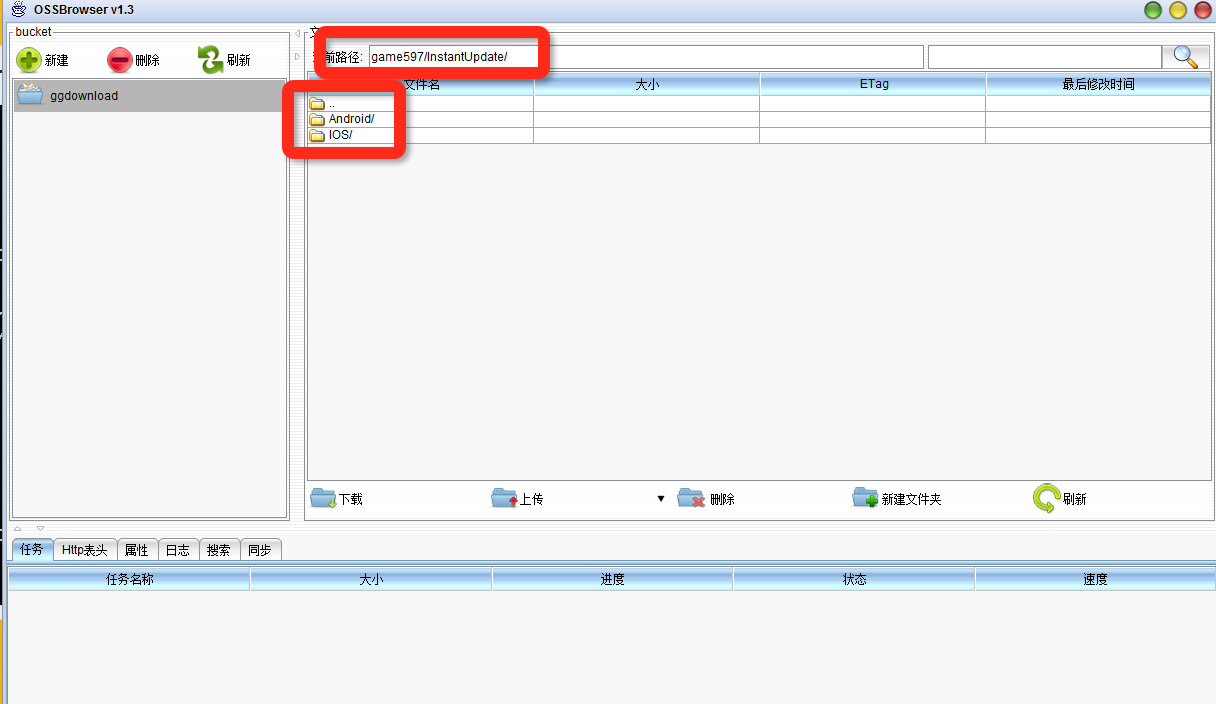
1.控制StreamingAssets里面的随包资源模块: 在Packager.cs 的DefaultGameModuleList字段中添加的模块,就会打包成随包资源. 修改该字段后需要重新打资源包.

2.把资源放到服务器上: 环境设置成发布环境,同时控制参数中m\_IsServer 设成false,其他都设成true, 重新打资源包(此时,资源更新仅从本地更新), 测试整个游戏没有问题后把资源放到服务器上.

资源所在位置. UnityProject/extract下面的对应版本号(bundle version)的文件夹



注意: 1.该版本号在mac打包会出错,这种情况下需要把该目录的名称改过来后再放到服务器上2.服务器上的资源要与平台相匹配(ios和 android)



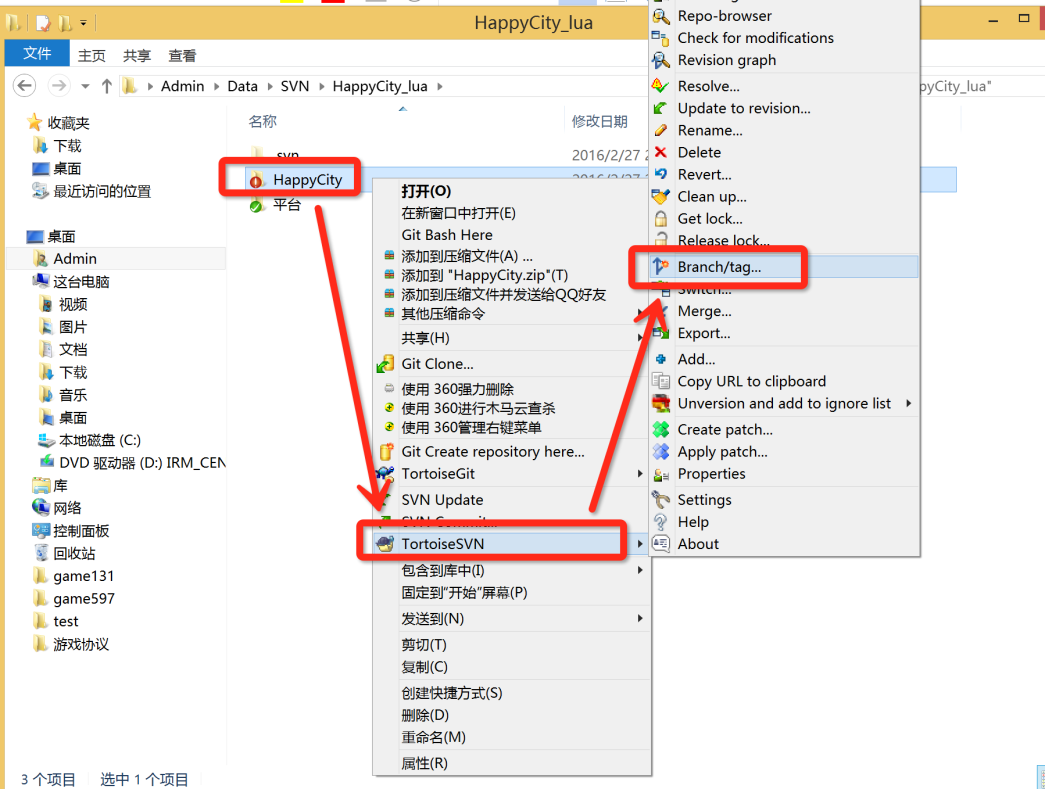
3.发布正式包: 正式发布前要把环境设置成发布环境,并把控制参数全部设成true(此时资源更新会从服务端更新),测试没有问题后才可以打正式包(.ipa/.apk)

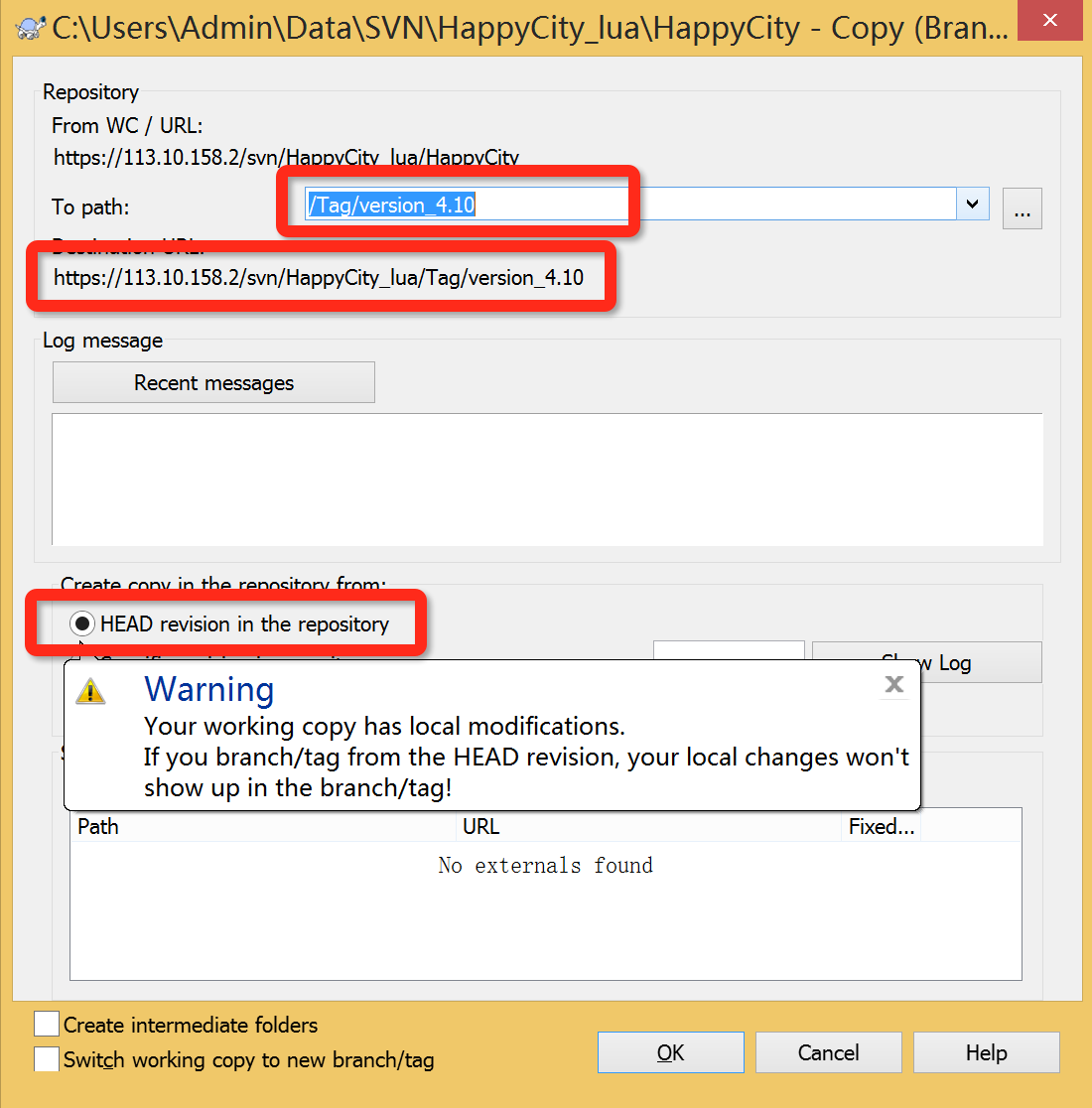
注意: 如果出现已经下载的模块无法再测试下载的情况下,把本地下载的资源的存储位置上的资源清空后重启游戏测试即可. 本地存储位置: Windos: c/simpleframework. Mac: UnityProject/c/

注意: IOS打包请在Mac下进行, Android打包请在Windows下进行(包括资源包和安装包)

打一个新版本的时候(与平台(131,597,510k)无关),需要先对svn进行打Tag,然后用更新该Tag到本地进行打包, 否则不同的svn路径下的打资源后的md5值会不同.

以version\_4.10为例: (打Tag的目的是记录好c#对应的版本,以便修改时与c#版本对应)





# 各个游戏平台bundleId 说明

bundleId：

game597

企业版／android  qbq.u.lobby597lua

IOS qbq.u.lobby

game131

企业版／android game407.u.lobby131lua

IOS game407.u.lobby

# 配置文件说明:

项目中平台(597,131,510k)的配置文件:

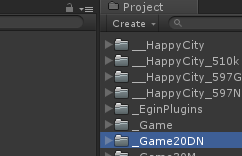
Android包 config999.xml

ios包 config.xml

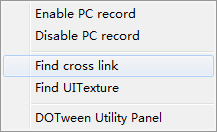
企业包 config888.xml

# 查游戏的交叉引用

1.在unity中project选需要查的目标，



2.执行菜单栏 tools/find cross link



3. 交叉引用会在log中打出



# 开发环境指定目录打包

**功能**: 选择某大厅或某游戏的打包工具,,目前在windows开发环境测试正常,因为编辑环境下没用压缩,所以速度比原来快了不少,修改某个prefab的话,一分钟内就能打出assetbundle看效果

**步骤**

1.svn更新

2.修改某个prefab或material

3.选择某游戏或大厅的文件夹

4.执行tool/create asign assetbundle

5.unity运行看效果,

(**注意**:只能用于开发环境,新加预制件要指定好assetbundlename,第一次选可能会慢些,后面单独修改某个prefab的话,速度就快很多了)