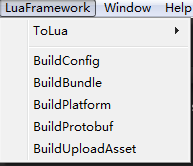


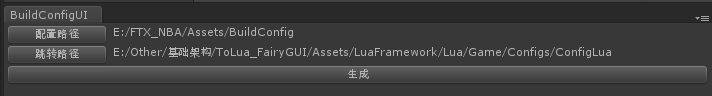
GenerateAll：wrap全部文件，没有wrap的文件，无法在Lua中使用

ClearWrapFiles：清除全部wrap的文件，出现warp文件有改动的情况可以使用



BuildConfig：

代码所在路径：Editor/BuildConfig



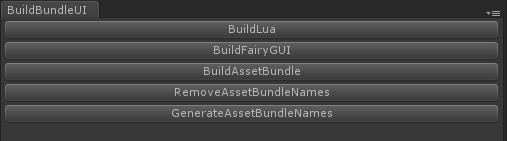
配置路径：把需要转成Lua配置的word文档全部放到该文件夹里面

跳转路径：跳转到Lua的配置文件夹上面，方便策划查找到文件上传配置

生成：把配置路径下的word文档转换成lua文件

BuildBundle：

代码所在路径：Editor/ BuildBundle



BuildLua：把lua代码打包成AB包，发包之后加载AB包解析成lua能识别的文件

BuildFairyGUI：把Fairy资源打包成AB包

BuildAssetBundle：把prefab资源打包成AB包(先GenerateAssetBundleNames生成标签，再点击BuildAssetBundle)

RemoveAssetBundleNames：prefab是通过设置标签的形式来打包的，提供了移除标签的功能

GenerateAssetBundleNames：生成prefab的标签，生成标签打包AB包比较快



备注：后面四个功能都需要在BuildBundleManager类里面添加数据才可以使用

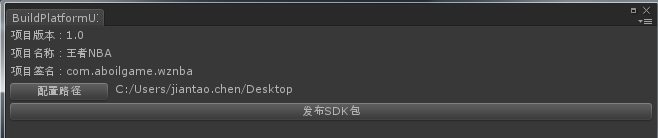
第一个参数：GameAssets资源都需要放在这个文件夹，全路径为：GameAssets/TextureProp

第二个参数：生成AB包之后的前缀，比如生成的文件名叫login，那么全名为：prop\_login

第三个参数：需要打包资源的后缀，如果是MP3的话，那么第三个参数就要写”mp3”

BuildPlatform:

代码所在路径：Editor/BuildPlatform



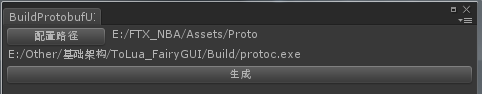
项目版本、项目名称和项目签名都是在代码里面修改的，因为这个基本不会怎么改动

配置路径：生成的APK的包所在的路径

发布：这里会把所有需要生成的资源和需要配置的资源全部都生成一次，然后打包成APK包

BuildProtobuf：

代码所在路径：Editor/ BuildProtobuf



配置路径：proto文件所在的文件夹

生成：把proto文件转换成pb文件

BuildUploadAsset

代码所在路径：Editor/ BuildUploadAsset



保存全部文件：记录StreamingGameAssets下面的全部文件并保存，用于比对有变动的文件，所有有变动的文件都会在列表里面显示，文件路径：Editor/ ModifyFile/ Config

重新生成资源：所有资源全部重新生成一次，用于比对有哪些文件有变动

Android测试服：把有变动的文件提交到外网测试服，方便测试在手机上测试更新是否正常

Android正式服：把有变动的文件提交到外网正式服，对所有玩家起作用，切记：当提交到正式服之后，那么记得点击保存全部文件