**XUPorter**

<https://onevcat.com/2012/12/xuporter/>

github：<https://github.com/onevcat/XUPorter>

A dependency porter from Unity to Xcode

<http://www.xuanyusong.com/archives/2720>

<https://blog.csdn.net/wolf96/article/details/79103304>

unity4用这套，unity5有自己的api

为什么想要自动添加？

由于unity是全平台的游戏开发环境，在开发中针对特定平台的特定功能时，很难避免根据对象平台的不同而引入不同的依赖。包括源码、需要的库和框架等。在使用各种插件后这种情况愈发严重：比如像加入内购功能，stroreKit.framework必不可少，而且也需要相应的处理代码。按照一般的Unity插件开发流程，在完成.cs的接口声明和Unity侧的调用实现后，最重要的当然是在ios native侧完成实现。而在以前，包括依赖库和所有源码文件，都只有在unity生成xcode工程之后，再手动添加。如果工程小依赖少的话花不了多少时间，但是如果项目很大，很可能折腾十来分钟，严重影响了工作效率，必须加以解决。

XUPorter可以读取Xcoe工程文件并进行解析，之后再Unity工程的Assets目录下寻找所以后的.projmods文件，并根据文件内容向工程中添加文件或库。

使用方法

将Guthub项目中的所有文件copy到unity工程文件夹下的/Assets/Editor目录中，XUPorter使用一个改良版的MiniJSON来进行。如果你的项目中已经在使用这个MiniJSON了的话，可以直接将XUPorter文件夹下的MiniJSON文件夹删掉；如果不一样的话，你可以选择其中一个重构一下或者加上命名空间来解决类名冲突。接下来，Mods文件夹下是示例文件以及需要导入XCode的文件。在看完以后，你需要把Modes文件夹下的所有.projmods文件以及Modes/iOS文件夹下的内容删除或者替换为你所需要的内容。

.projmods文件是一个JSON格式的配置patch文件，定义了要如何设置Xcode工程。举个基本的例子，比如KKKKeychain.projmods：



各参数定义如下：

Group：所有由该projmods添加的文件和文件夹所属的xcode中的group名称

Libs：在xcode build phases中需要添加的动态链接库的名称，比如libz.dylib

Frameworks：在xcode build phases中需要添加的框架的名称，比如Security.framework，这里是系统的framework，

并不是第三方的framework

Headerpaths：在xcode编译设置中的Header Search Paths路径

Files：加入工程的文件名

Folders：加入工程的文件夹，其中所有的文件和文件夹都将被加入工程中

Linker flags：添加到工程linker flag中的链接配置，比如-ObjC

Excludes：忽略的文件的正则表达式，匹配的文件将不会被加入到工程中

Frameworks分成两种，一种是系统自带的framework还有一种是第三方做出来的framework，**Framework节点里面放的是系统自带的frameworks，files节点里放的是第三方做出来的framework，**尤其是第三方的framework如果位置放的不对，就不会被xcode所引用，切记切记！！

Folders：可以把某个文件夹下的所有文件拷贝在xcode工程里面，一般sdk都会附带一些oc代码文件，最好通过folders把oc的代码拷贝在工程里面。或者也可以把oc的代码放在plugins下面，同样打包的时候也会拷贝进xcode工程



**Unity打完IOS或者Android包以后会自动回调一个静态方法**

**[postProcessBuild(100)]**

**Public static void OnPostProcessBuild(BuildTarget target, string pathToBuiltProject)**

自动添加framework的原理其实就是等包打完之后，在这个方法里面进行文件的操作，把需要的framework plist oc代码拷贝进去，或者修改它们。

[PostProcessBuild]

public static void OnPostProcessBuild(BuildTarget target, string pathToBuiltProject)

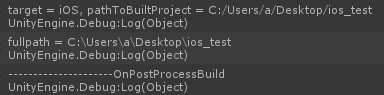
{

Debug.Log("target = " + target + ", pathToBuiltProject = " + pathToBuiltProject);

Debug.Log("fullpath = " + Path.GetFullPath(pathToBuiltProject));

Debug.Log("---------------------OnPostProcessBuild")

}



更多的例子可以参看Mods文件夹中的其他projmods文件。所有定义路径都是基于当前projmods文件位置的相对路径。最后，在完成projmods后，unity会在编译完成后，调用XCodePostProcess的OnPostProcessBuild来对编译得到的XCode进行修改。