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

废话之前

[XUPorter项目Github链接](https://github.com/onevcat/XUPorter)

**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：比如想加入内购功能，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太多时间，但是如果项目很大，很可能折腾一次就要十来分钟，严重影响了工作效率，必须加以解决。

**怎么办**

Unity开发团队也意识到了这个问题，在Unity编译的最后加入了一个脚本调用的命令，会自动搜索Editor文件夹下的PostprocessBuildPlayer，并进行调用，在该文件中可以自己加入脚本来向Xcode中添加库和文件。关于PostprocessBuildPlayer的详细信息，可以参看[官方文档](http://docs.unity3d.com/Documentation/Manual/BuildPlayerPipeline.html)，关于向Xcode中添加文件或库，gonzoua的[xcs](https://github.com/gonzoua/xcs)也许是不错的选择。但是似乎xcs只能针对Xcode3来添加，在Xcode4中，主工程文件的结构发生了改变，导致xcs失效，而这个项目也迟迟没有更新（也许有时间我会考虑接手继续这个项目，但肯定不是现在…）。因此不得不打其他主意。

在Unity3.5中，加入了一个很棒的标签——[[PostProcessBuild]](http://docs.unity3d.com/Documentation/ScriptReference/PostProcessBuildAttribute.html)，被该标签标注的函数将自动在build player后被调用，这为大家提供了一个不需要用脚本和命令行就能添加或修改编译得到的工程的绝好的入口。darktable用python实现了一个Xcode4工程文件读写的接口[Mod PBXProj](https://bitbucket.org/darktable/mod-pbxproj/overview)，但是对于Unity来说，更需要的是C#的实现。Cariola完成了[一部分实现](https://github.com/dcariola/XCodeEditor-for-Unity)，但是存在一些错误和不太好用的地方，代码也很乱。我在其基础上进行了一些改进和整理。但是因为变动的还是比较大，很难merge回去，所以决定自己开一个项目来继续推进这个项目。

**XUPorter**

我把它叫做XUPorter，a dependency porter from Unity to Xcode。XUPorter可以读取Xcode工程文件并进行解析（再次感谢darktable的工作），之后在Unity工程的Assets目录下寻找所有的.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Github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](https://github.com/prime31/UIToolkit/blob/master/Assets/Plugins/MiniJSON.cs)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Mods文件夹下的所有.projmods文件以及Mods/iOS文件夹下的内容删除或者替换为你所需要的内容。

在[这里](http://d.pr/f/HAzc)提供了.unitypackege格式文件的下载，你也可以选择下载打包好的文件并导入你的工程，之后的步骤和上面一样。

.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KKKeychain.projmods：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | {  "group": "KKKeychain",  "libs": [],  "frameworks": ["Security.framework"],  "headerpaths": [],  "files": [],  "folders": ["iOS/KKKeychain/"],  "linker\_flags": [],  "excludes": ["^.\*.meta$", "^.\*.mdown$", "^.\*.pdf$"]  } |

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
* headerpaths：Xcode中编译设置中的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工程中

更多的例子可以参看Mods文件夹中的其他projmods文件。所有的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工程进行修改。

之后进一步要做的是为MiniJSON添加一个namespace，这样可以避免不必要的冲突。另外如果您有什么好的想法，也欢迎fork这个项目并给我pull request。项目的github链接请[猛击这里](https://github.com/onevcat/XUPorter)。