梳理一下xcode的目录及配置

左侧：

1. 项目目录
2. 没用过
3. 没用过
4. 查找
5. 警告和报错
6. 没用过
7. 调试的时候的各种指标
8. 断点

中间：

左侧分project和targets，目前只知道target是类似可以分不同的版本，默认是第一个正式版本，设置也都在这里设置，project目前写这个的时候还没测，可能是模拟器吧，有空看一下

1. General
   1. 名字包名版本啥的
   2. 目标平台 ios版本
   3. 横竖屏
   4. 库
2. Signing&Capabilities
   1. 这里可以选择自动管理签名，自动管理之后先选择team，然后会根据包名自动选择签名
   2. 如果不选择的话，就需要手动选择签名
3. Info，一些配置，比如我接的sdk就需要往里录一些facebook id啥的，目前不知道具体怎么用
4. Build settings
   1. 所有配置基本都在这里
   2. Build Options里有个Enable Bitcode，这个基本上都需要设为NO
   3. Linking里的Other Linker Flags里我接的sdk里就需要加-ObjC
5. Build Phases
   1. 库文件，目前有个不太理解的地方是，有的时候这里显示引入库了，但实际并没有引入，我需要把这里的库删了再重新导入才生效，暂时不知道为什么

以上都是这两天接苹果的一些粗略的了解，了解的并不深

以下就是和unity相关的了，这里东西比较多，还有一个额外的导入类，放在同级目录里了，因为接很多SDK需要配置一堆东西，如果每次导出XCODE后再去修改，肯定是不行的，所以需要一套自动配置的代码，达到导出XCODE后可以直接打包，先缕一下目录关系

1. unity会把Plugins里的ios相关文件都导出，就算不是ios命名的文件夹也一样
2. 导出后在xcode里会大概分为两个目录，framework会放在Frameworks目录里，其他的.mm .h .a都会放在Libraries里
3. 一般sdk都会有一些单独初始化代码和一些其他代码，所以这里需要用一个类继承UnityAppController，见CustomAppController所以这
4. 差点忘了还得写unity和oc互相调用
   1. Unity-oc：
      1. Unity端

[DllImport("\_\_Internal")]

private static extern void UnityToOC(int type, string s);

* + 1. Oc端：

//声明

**extern** "C"{

**void** UnityToOC(**int** type, **char**\* c);

}

//实现

**void** UnityToOC(**int** type, **char**\* c){balabala}

* 1. Oc-unity unity的gameobject名，回调函数名，参数，只能有一个string
     1. UnitySendMessage("ShoumengObject", "PayResult", [str UTF8String]);
  2. 这里就得说到json了，oc里的json是需要和类互转的，貌似不能直接根据string取，见文件

1. 自动配置xcode工程，见文件