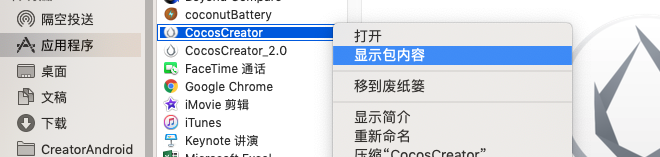
cocoscreator 嵌入原生IOS

这个教程是根据cocos2d-js嵌入iOS研究而来 可以先看看原生cocos2d-js如何嵌入ios的

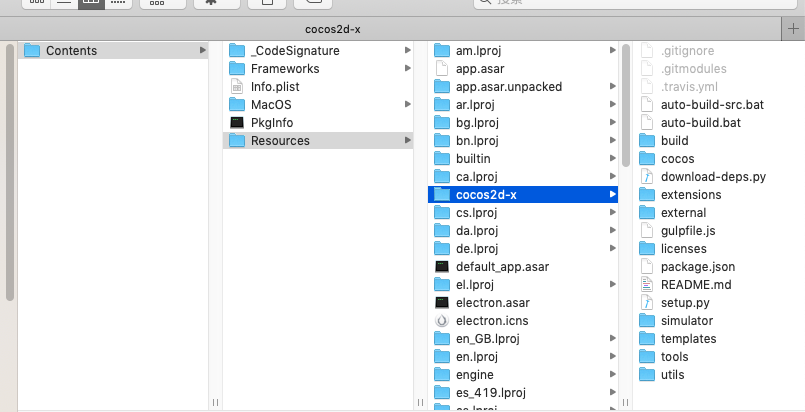
<https://blog.csdn.net/houjia159/article/details/81014787>

下面是cocoscreator嵌入ios (由于creator原生cocos2dx对jsbinding和c++进行了整合 就不用生成 libjscocos2d IOS.a 了)

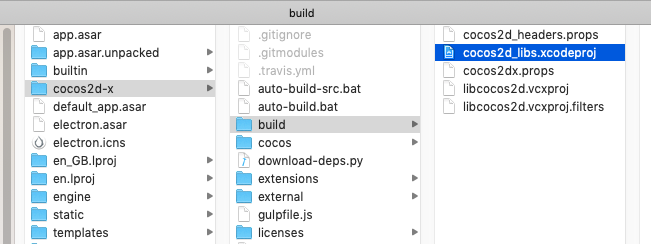
1、进入应用程序选择 CocosCreator（2.2.0版本） 显示包内容



2、按照下图找到coco2dx源代码



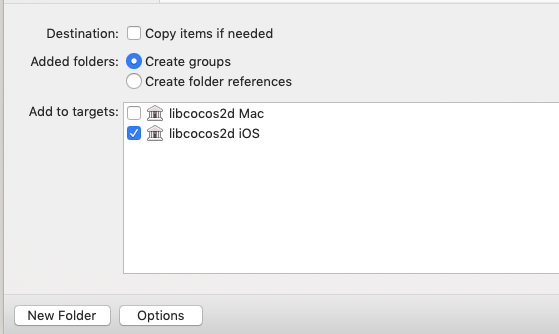
3、按照下图打开 cocos2d\_libs.xcodeproj 工程这个是c++和js-binding的原生工程



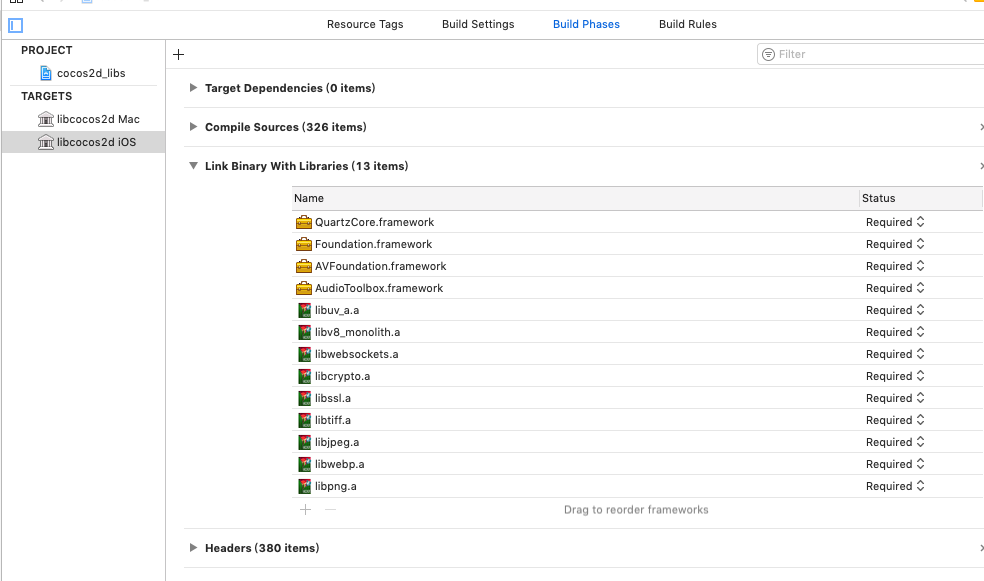
4、需要将cocos项目proj.ios\_mac 里面的Classes代码目录复制到build目录下，和cocos2d\_libs.xcodeproj同级。

5、将Classes目录加入到工程里面  （选择 Create groups 和libcocos2d iOS）

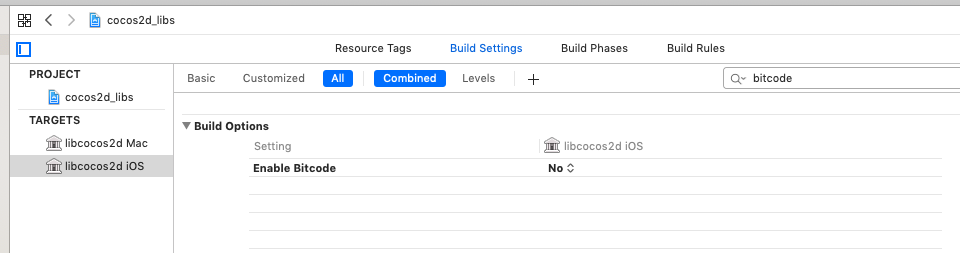
这个Classes目录原来只有两个文件 Appdelegate.h 和cpp文件，下面是写好的代码目录直接复制就行.



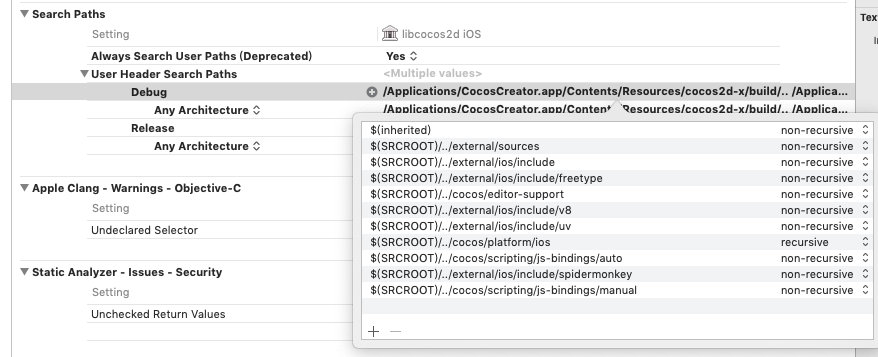
6、选择ios工程的Build Phases将一下库加入 (一个也不要少 不然报错)



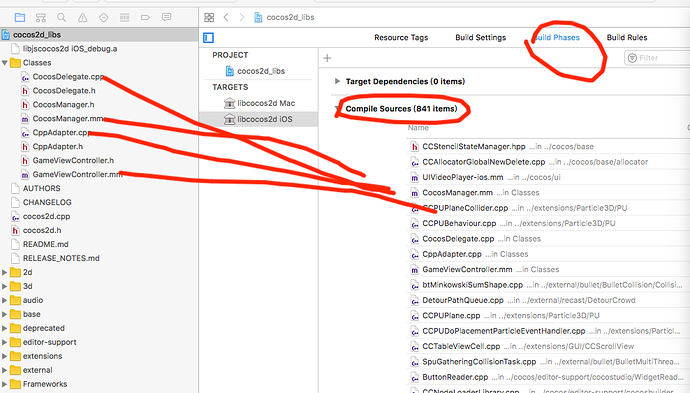
7、 如果Bitcode 开启则需要关闭



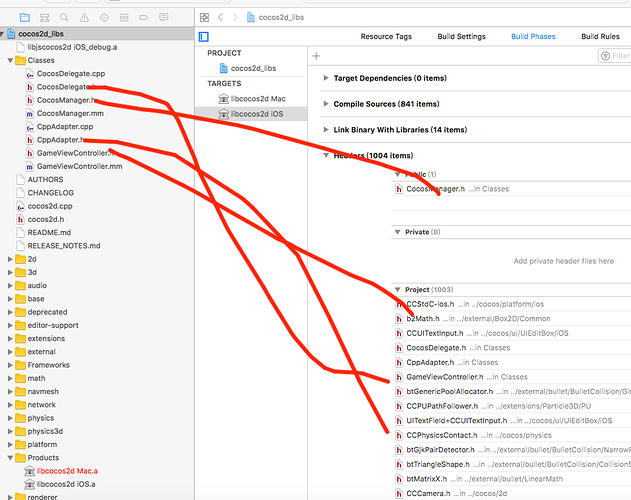
8、选择     Build Settings  设置头文件查找路径  配置头文件路径 （同样一个也不要少）



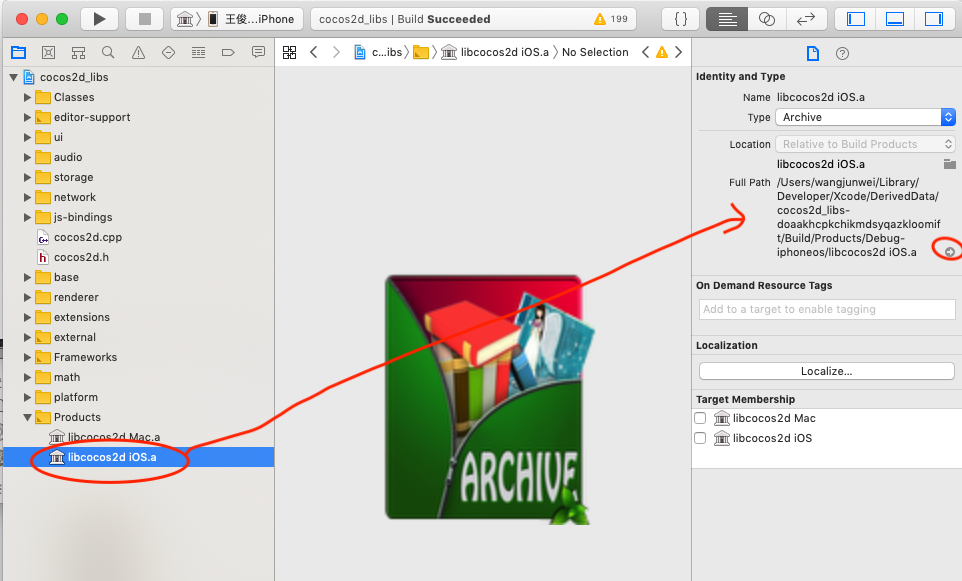
9、如图把Classes下的可执行文件即.mm .cpp文件拖到Compile Sources里，参与编译。

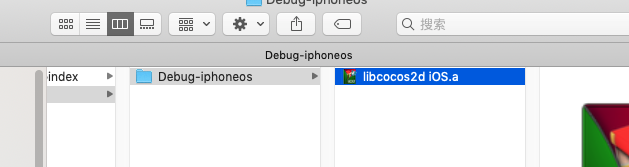


10、这里是重点了！！！   把CocosManager.h拖到Headers下面的Public下，其他的头文件都拖到Project下，因为之后CocosManager.h要暴露出来作为接口给iOS用的。



11、然后Command+B编译 生成一个     libcocos2d iOS.a  ，这个文件比较大200多M这个是所有c++引擎和js-binding以及你Classes代码目录里代码的集合库。点击最下方Products的libcocos2d iOS.a 在右边找到路径 点击 箭头 就跳转到实际路径了.

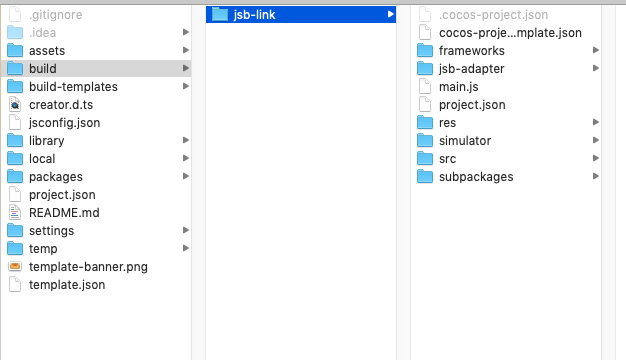


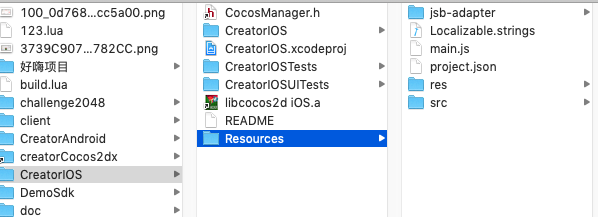


12、这样cocos2dx的静态库文件就生成了。里面只是引擎+jsbinding+Classes代码目录的合集 没有js代码和res资源文件。

13、接下来仙剑一个ios空工程。然后把刚刚生成的静态库 libcocos2d iOS.a 和CocosManager.h拷贝到ios工程里面。这个CocosManger.h是一个暴露给 原生的一个api文件。里面的api可以直接被原生调用。

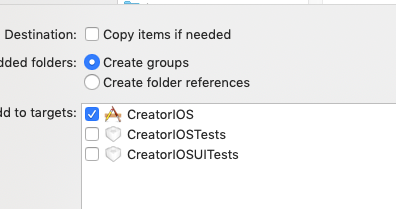
14、新建一个Resources目录 ，把cocoscretor目录下的build/jsb-link 目录里的 res   src  jsb-adapter main.js project.json放进去



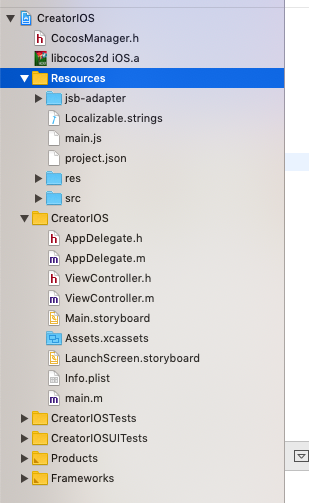


15、这样需要的资源就全了 （libcocos2d iOS.a     CocosManager.h   Resources）,打开工程文件

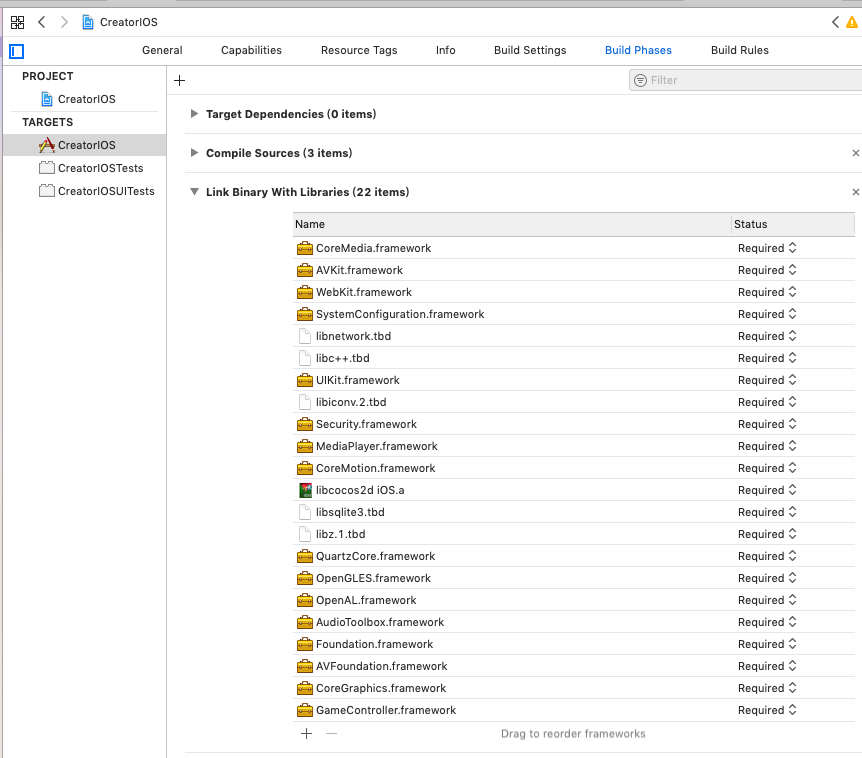
将这三个导入项目中（Resources 选择 Create groups，里面的文件选择 Create folder references ，不然会报错 ）



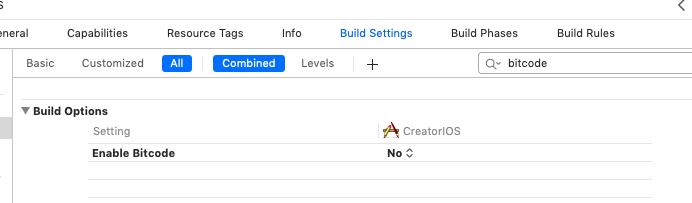
16、导入后的目录结构是这样的 （Resources是黄色  里面的目录都是蓝色）



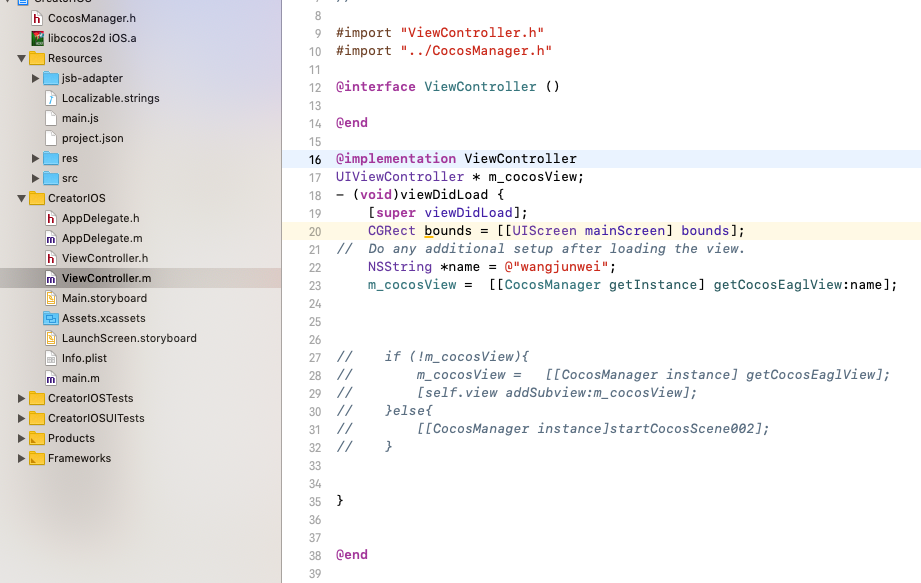
17、把下面的库全部加上 （一个也不要少)   同时关闭 bitcode



————————————————————关闭bitcode---------------------------------------



18、最后在ViewController.m引入 CocosManager.h文件 初始化cocos界面就可以了。



19、以上只是最简单的嵌入方式，后面可以加上 jsb调用 app原生，jsb是cocos2dx引擎写好的api。可以直接调用oc代码。

        if (cc.sys.os === cc.sys.OS\_IOS || cc.sys.os === cc.sys.OS\_OSX) {

            jsb.reflection.callStaticMethod('AppController', 'showAlertDialog:withMessage:', "Title", "Native Call Test");

        }else if(cc.sys.os === cc.sys.OS\_ANDROID){

           let className = "com/example/creatorandroid/GameActivity";

           let methodName = "showAlertDialog";

           let methodSignature = "(Ljava/lang/String;Ljava/lang/String;)V";

           jsb.reflection.callStaticMethod(className, methodName, methodSignature, "Title", "Native Call Test");

        }

20、app原生oc调用js，可以直接调用CocosManager.h里的方法。然后在CocosManager.cpp里调用js代码来实现。