LayaPlayer-iosSDK使用文档

目录

[1、适用人群 2](#_Toc460276667)

[2、创建项目 2](#_Toc460276668)

[3、修改最低版本 2](#_Toc460276669)

[4、设置头文件搜索路径 3](#_Toc460276670)

[5、修改lib搜索路径 4](#_Toc460276671)

[6、添加依赖库 4](#_Toc460276672)

[7、添加Framework 5](#_Toc460276673)

[8、修改BitCode 5](#_Toc460276674)

[9、ViewController代码 6](#_Toc460276675)

[10、修改AppDelegate代码 10](#_Toc460276676)

[11、导入资源 10](#_Toc460276677)

[12、编译并运行项目 11](#_Toc460276678)

[13、启动地址的修改 12](#_Toc460276679)

[14、项目配置，config.ini 12](#_Toc460276680)

[15、平台对接 12](#_Toc460276681)

[16、SDK目录描述 12](#_Toc460276682)

[17、IPV6 12](#_Toc460276683)

[18、编译项目 13](#_Toc460276684)

[19、Logo界面 14](#_Toc460276685)

[19.1进度条控制 14](#_Toc460276686)

[19.2 替换掉开发者的自己的Logo 15](#_Toc460276687)

[19.3 想要做特别酷炫的进度条怎么办？ 15](#_Toc460276688)

[19.4 后续白名单功能 15](#_Toc460276689)

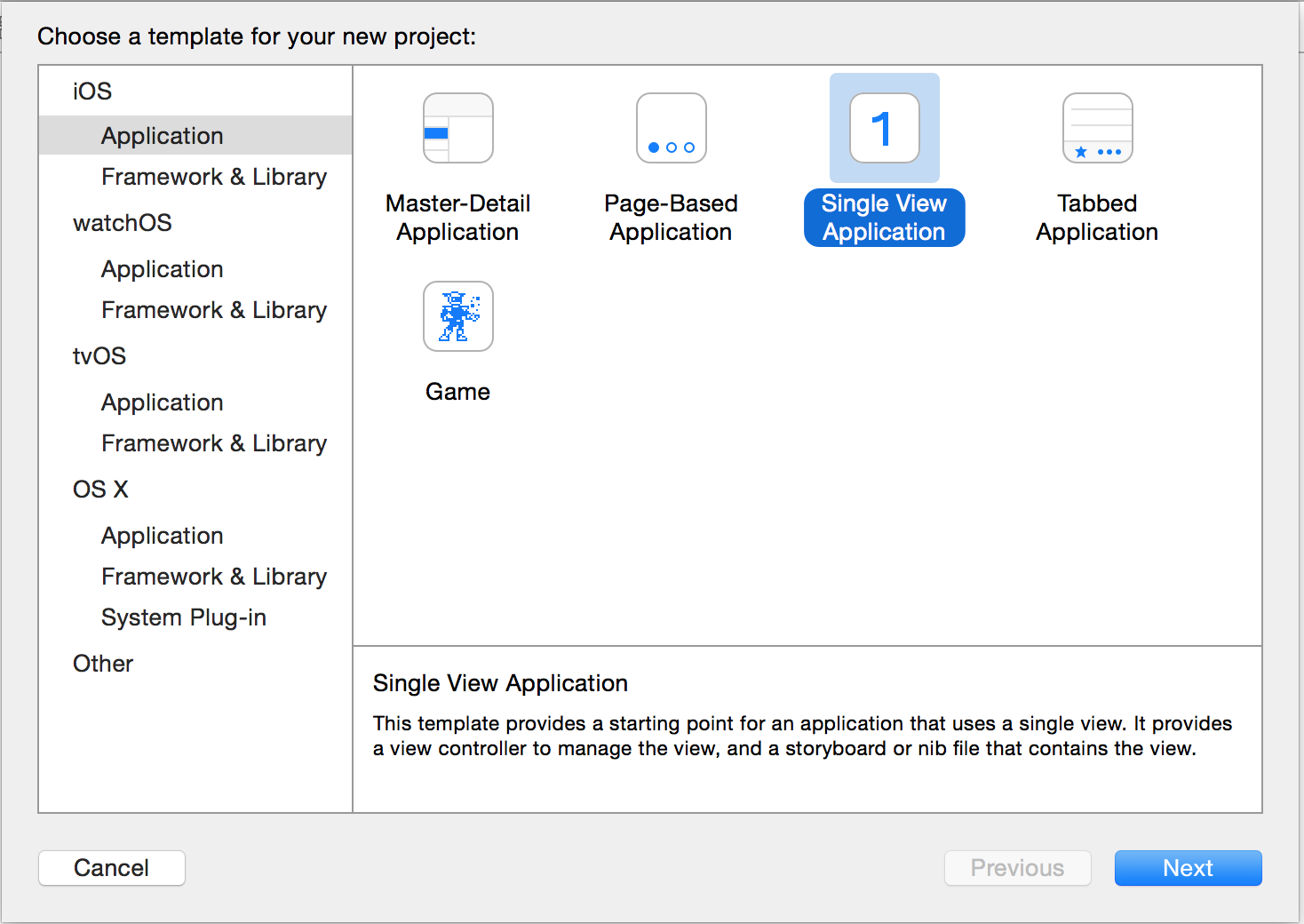
[20、注意事项 16](#_Toc460276690)

# 1、适用人群

该文档适用会ios开发的程序员，因为还是需要一些ios开发的基本知识。

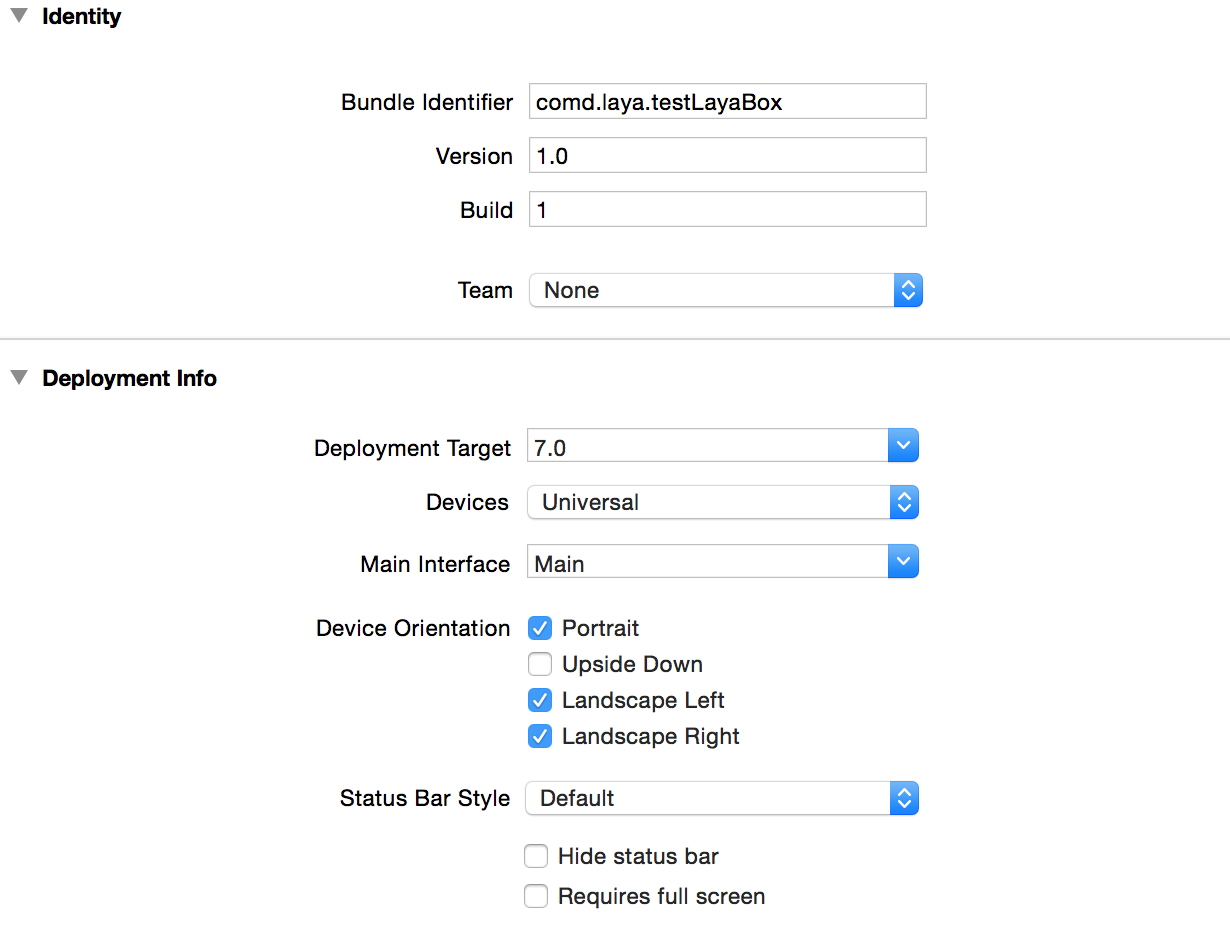
# 2、创建项目

创建一个ios的Application，选择Single View Application。



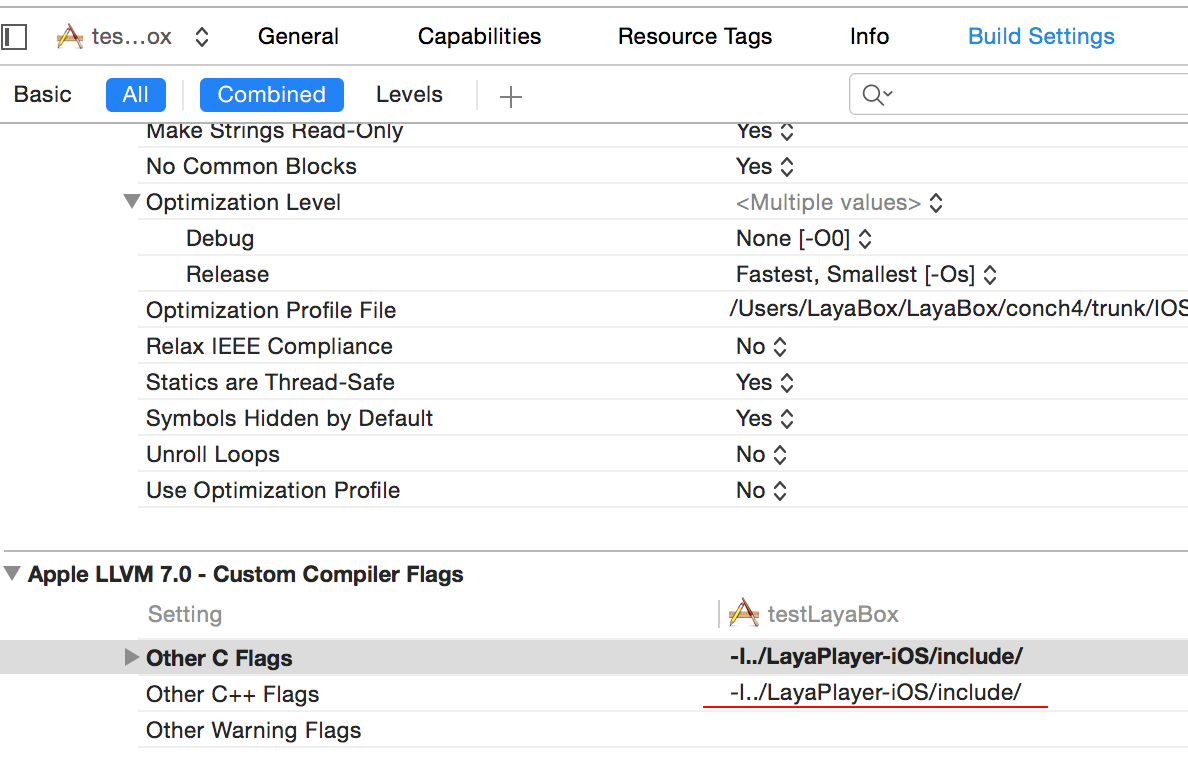
# 3、修改最低版本

LayaPlayer最低支持ios7.0版本，然后修改一下最低版本为，7.0



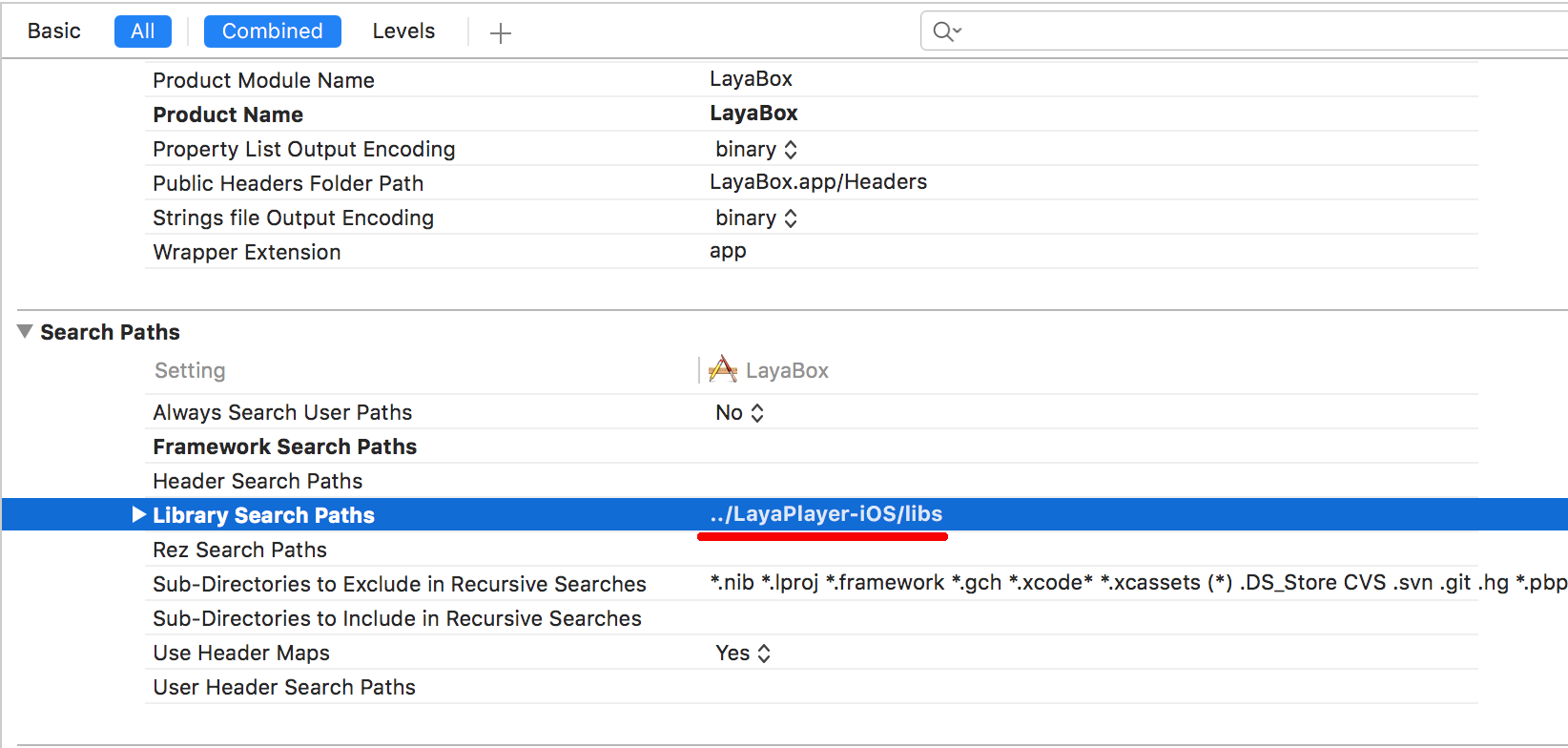
# 4、设置头文件搜索路径

选择Build Setting选项在Custom CompilerFlags选项中，增加头文件搜索路径，“-I../LayaPlayer-iOS/include/ ”，如下图所示：



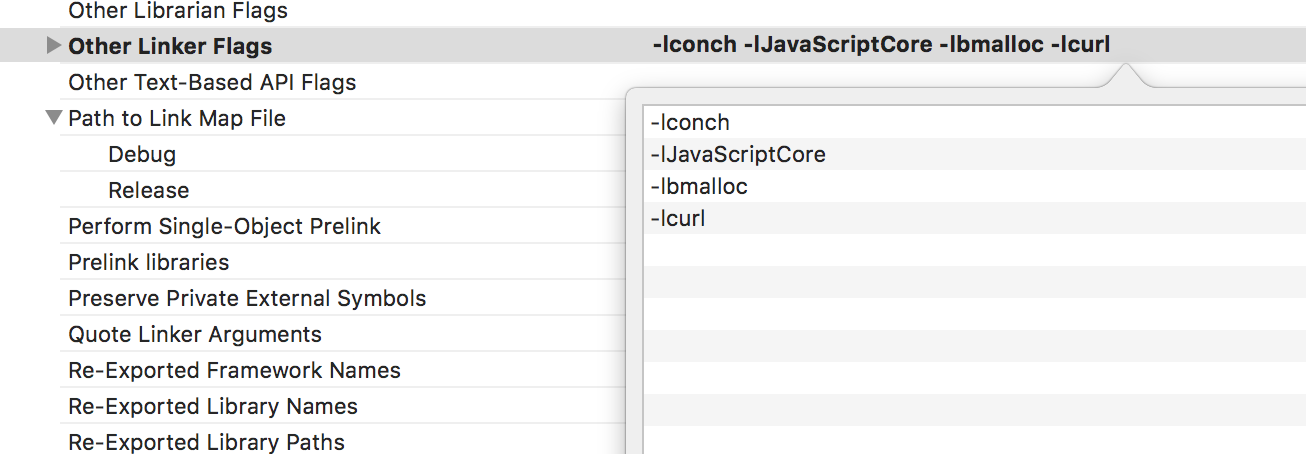
# 5、修改lib搜索路径

选择Build Setting-->Search Paths-->Library Search Paths目录下，添加库文件的搜索路径"../LayaPlayer-iOS/libs"，如下图所示：



# 6、添加依赖库

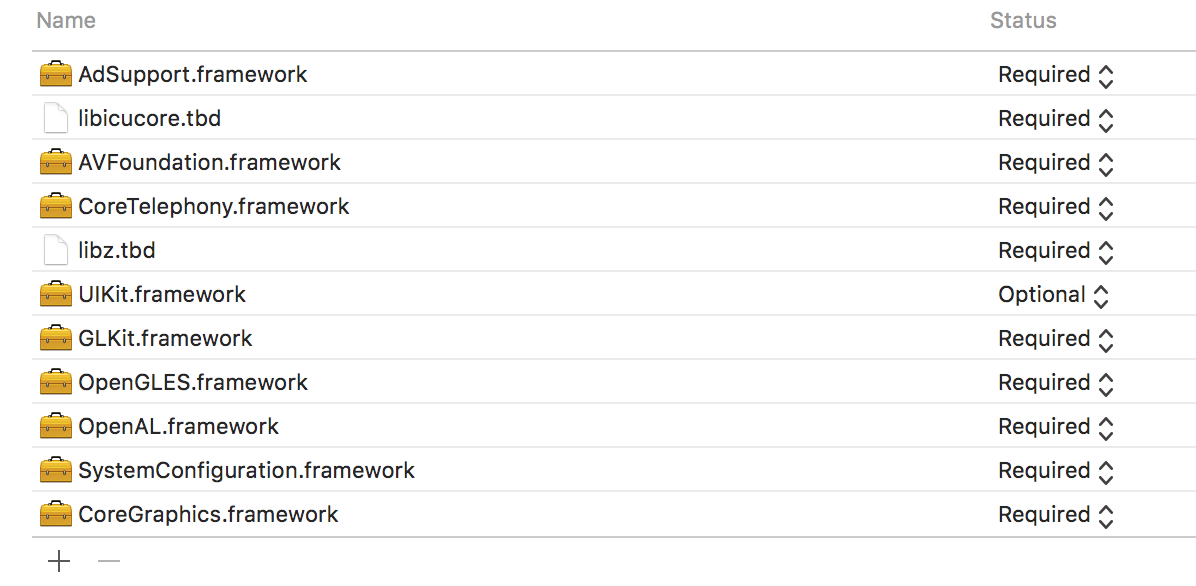
选择Build Setting-->Linking-->other Linker Flags目录下，填入依赖的库，"-lconch"、  
"-lJavaScriptCore"、"-lbmalloc"，如下图所示:



注意：如果想支持https，可以link "-lcurls"。

# 7、添加Framework

LayaPlayer需要一些系统默认的Framework，需要在 General-->Link Frameworks and Libaries中添加，如下图所示:



# 8、修改BitCode

现在编译一下项目发现无法编译过，提示：

ld: '../LayaPlayer-iOS/libconch.a(path.o)' does not contain bitcode. You must rebuild it with bitcode enabled (Xcode setting ENABLE\_BITCODE), obtain an updated library from the vendor, or disable bitcode for this target. for architecture armv7  
clang: error: linker command failed with exit code 1 (use -v to see invocation)

现在需要修改一下项目配置，Build Settings-->Build Options-->Enable Bitcode 改成no

# 9、ViewController代码

正式添加代码，修改ViewController，让其继承 GLKViewController，代码如下所示：

#import <UIKit/UIKit.h>

#import <GLKit/GLKit.h>

#import <conchRuntime.h>

@interface ViewController : GLKViewController

{

@public

GLKView\* m\_pGLKView;

EAGLContext\* m\_pGLContext;

conchRuntime\* m\_pConchRuntime;

}

+(ViewController\*)GetIOSViewController;

-(id)init;

@end

修改ViewController为mm格式，并且代码如下：

#import "ViewController.h"  
@implementation ViewController  
static ViewController\* g\_pIOSMainViewController = nil;  
+(ViewController\*)GetIOSViewController  
{  
    return g\_pIOSMainViewController;  
}  
-(id)init  
{  
    self = [super init];  
    if( self != nil )  
    {  
        g\_pIOSMainViewController = self;  
        return self;  
    }  
    return Nil;  
}  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    //保持屏幕常亮，可以通过脚本设置  
    [[UIApplication sharedApplication] setIdleTimerDisabled:YES];  
    self->m\_pGLContext = [[EAGLContext alloc] initWithAPI:kEAGLRenderingAPIOpenGLES2];  
    if (!self->m\_pGLContext)  
    {  
        NSLog(@"Failed to create ES context");  
    }  
    m\_pGLKView = (GLKView \*)self.view;  
    m\_pGLKView.context = self->m\_pGLContext;  
    m\_pGLKView.drawableDepthFormat = GLKViewDrawableDepthFormat24;  
    [EAGLContext setCurrentContext:self->m\_pGLContext];  
    self.preferredFramesPerSecond = 10000;  
      
    //conchRuntime 初始化ConchRuntime引擎  
    m\_pConchRuntime = [[conchRuntime alloc]initWithView:m\_pGLKView EAGLContext:m\_pGLContext logoPath:@"logo.jpg"];  
}  
- (void)dealloc  
{  
    [self tearDownGL];  
    if ( [EAGLContext currentContext] == self->m\_pGLContext )  
    {  
        [EAGLContext setCurrentContext:nil];  
    }  
}  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
      
    //conchRuntime 内存警告的时候的处理  
    [m\_pConchRuntime didReceiveMemoryWarning];  
}  
- (void)tearDownGL  
{  
    [EAGLContext setCurrentContext:self->m\_pGLContext];  
}  
- (void)glkView:(GLKView \*)view drawInRect:(CGRect)rect  
{  
    //conchRuntime renderFrame  
    [m\_pConchRuntime renderFrame];  
}  
- (void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event  
{  
    //conchRuntime touch  
    [m\_pConchRuntime touchesBegan:touches withEvent:event];  
}  
- (void)touchesMoved:(NSSet \*)touches withEvent:(UIEvent \*)event  
{  
    //conchRuntime touch  
    [m\_pConchRuntime touchesMoved:touches withEvent:event];  
}  
- (void)touchesEnded:(NSSet \*)touches withEvent:(UIEvent \*)event  
{  
    //conchRuntime touch  
    [m\_pConchRuntime touchesEnded:touches withEvent:event];  
}  
- (void)touchesCancelled:(NSSet \*)touches withEvent:(UIEvent \*)event  
{  
    //conchRuntime touch  
    [m\_pConchRuntime touchesCancelled:touches withEvent:event];  
}  
-(NSUInteger)supportedInterfaceOrientations  
{  
    return [conchConfig GetInstance]->m\_nOrientationType;  
}  
- (BOOL)shouldAutorotate  
{  
    return YES;//支持转屏  
}  
@end

conchRuntime的重点代码解释：

(1)、initWithView 初始化conchRuntime需要传入GLKView和EAGLContext

(2)、renderFrame 主要的渲染函数

(3)、touchesBegan、touchesMoved、touchesEnded、touchesCancelled中分别调用conchRuntime的函数

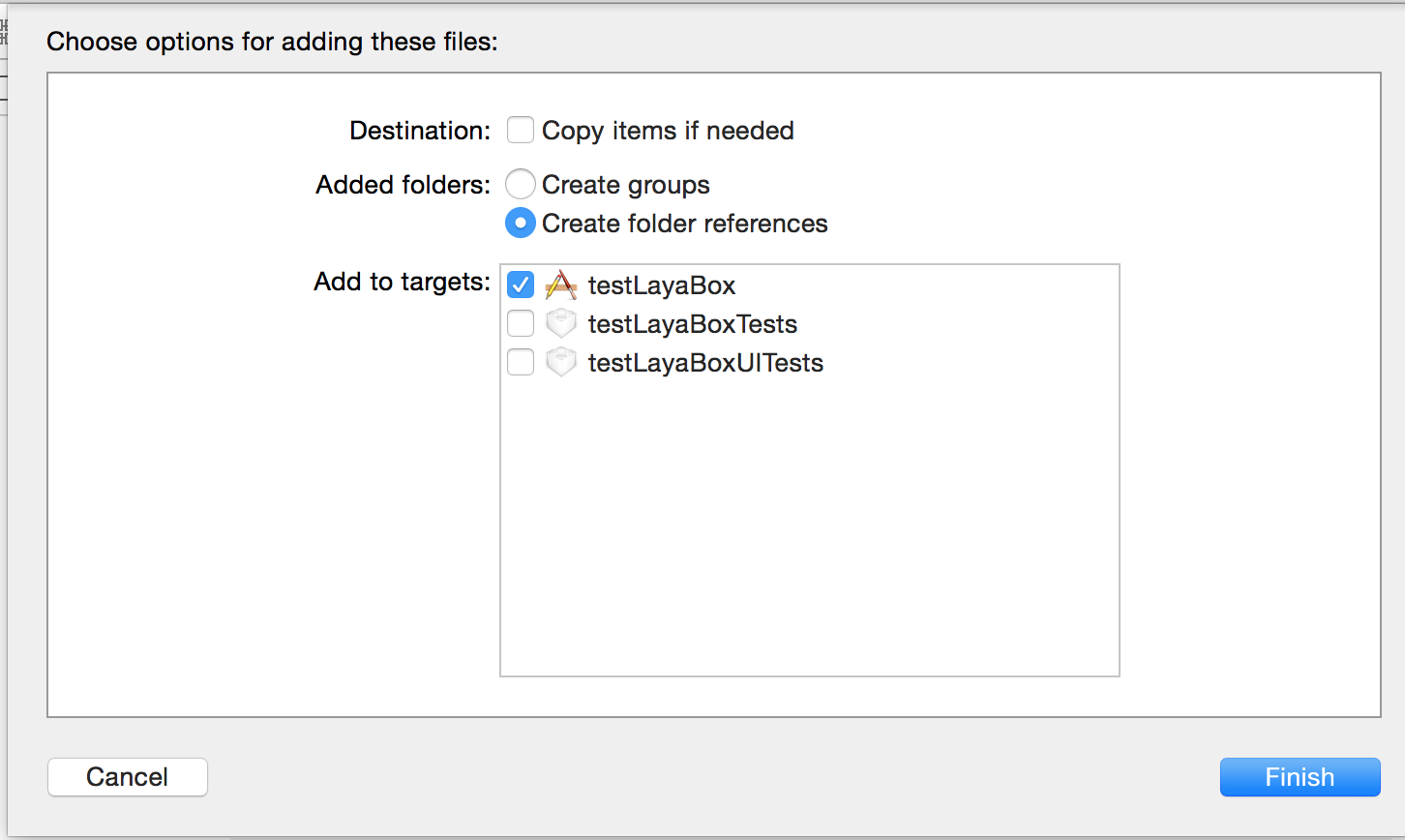
# 10、修改AppDelegate代码

修改AppDelegate为mm格式在AppDelegate的didFinishLaunchingWithOptions类中new ViewController，如下代码所示：

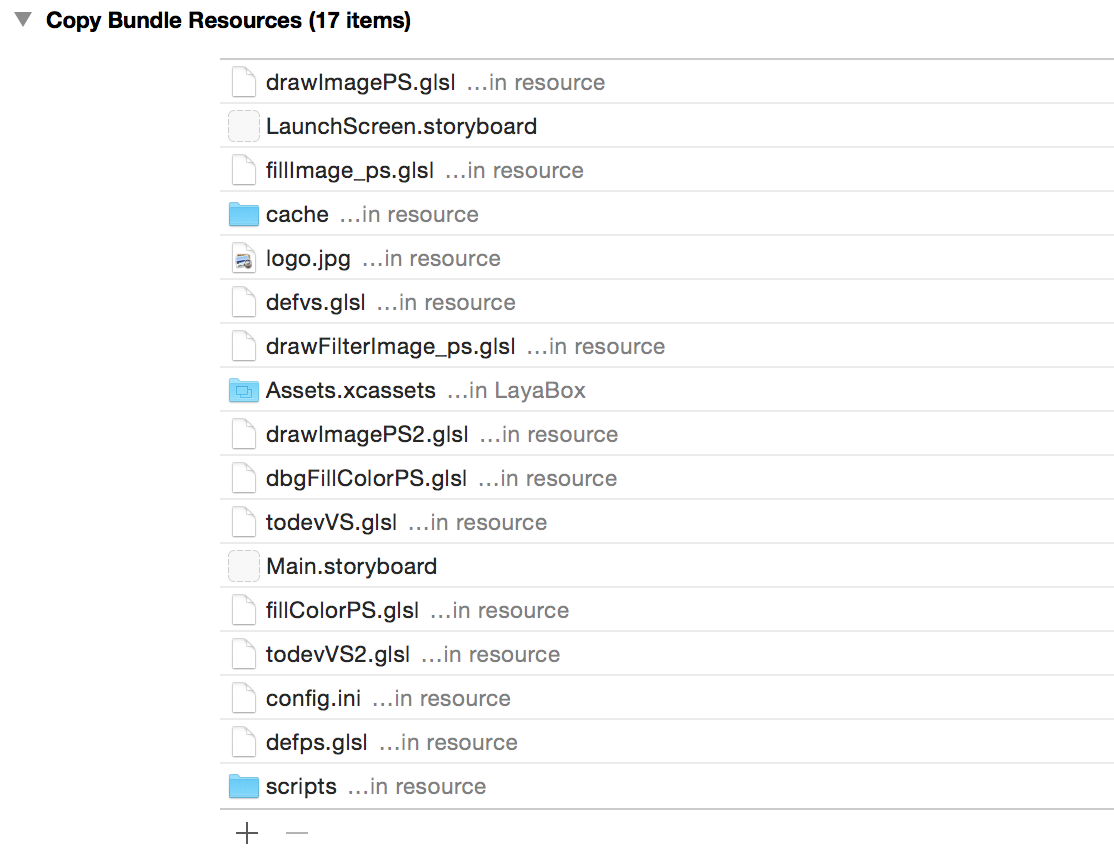
self.window=[[UIWindow alloc]initWithFrame:[[UIScreen mainScreen] bounds]] ;  
ViewController\* pViewController  = [[ViewController alloc] init];  
self.window.rootViewController = pViewController;  
[self.window makeKeyAndVisible];

# 11、导入资源

把LayaPlayer-iOS/resource 目录拷贝到项目中，然后在xcode项目下创建resource目录，然后把resource所有文件拖到xcode的resource目录下，选择"Create folder references"如下图所示：

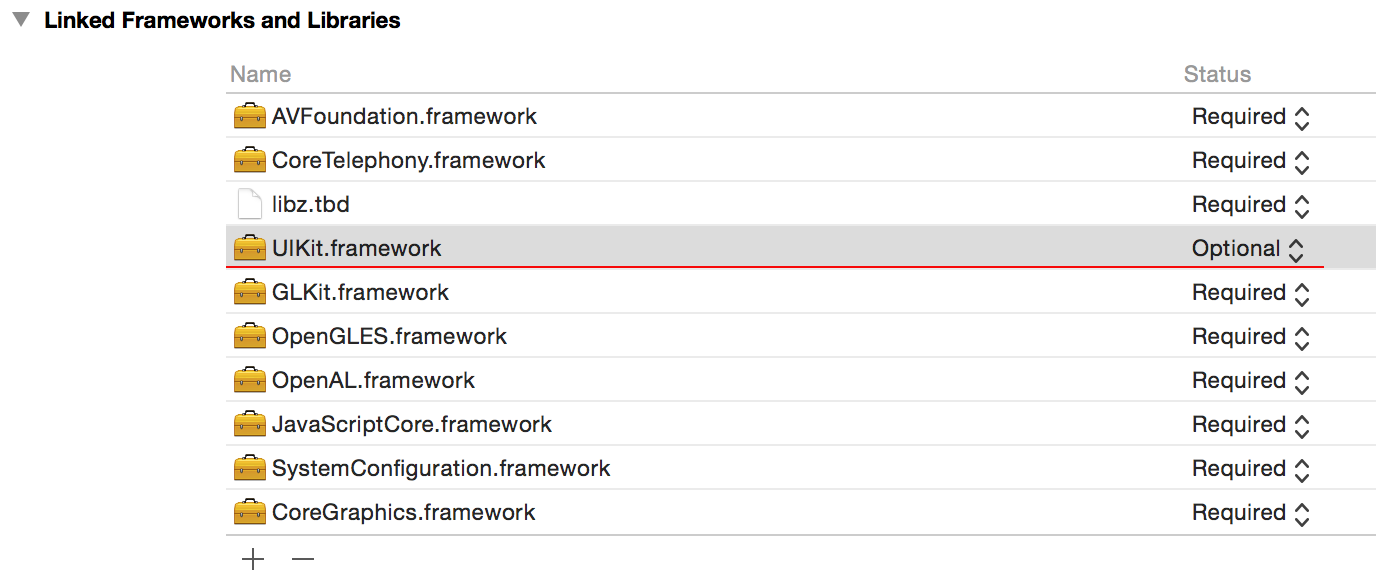


可以在 BuildPhases-->Copy Bundle Resources 目录下是正确的



# 12、编译并运行项目

在ios7.0的设备上，会运行不起来直接报错，这个时候就需要，导入的FrameWork的UIKit.framework的格式为 Optional，如下图所示：



# 13、启动地址的修改

(1)、打开index.js，最下面修改启动地址

loadUrl(conch.presetUrl||"http://10.10.20.200:8899/default.html");

# 14、项目配置，config.ini

orientation 这个屏幕的方向按照如下值进行设定

/\*

UIInterfaceOrientationMaskPortrait, ===2  
 UIInterfaceOrientationMaskPortraitUpsideDown, ===4  
 UIInterfaceOrientationMaskLandscapeLeft, ===8  
 UIInterfaceOrientationMaskLandscapeRight, ===16

\*/

# 15、平台对接

ios-内购和第三方渠道接入（登陆、支付等），需要开发者进行二次开发，LayaPlayer引擎可以提供了一个Market类，这个类可以实现JS和Object-C、Java之间的互调。

**具体对接文档，**请参考其他文档。

# 16、SDK目录描述

LayaBox是一个简单的例子，如果觉得特别上面的步骤特别繁琐，您可以直接使用这个例子，进行开发。

LayaPlayer-iOS是SDK目录和文档

# 17、IPV6

苹果公司在2016年6月1日起，强制执行ipv6标准，所以开发者发布项目的时候，http请求和socket都必须使用域名的方式，不能使用ip地址。

如何测试ipv6网络下是否正常，请参考以下文档

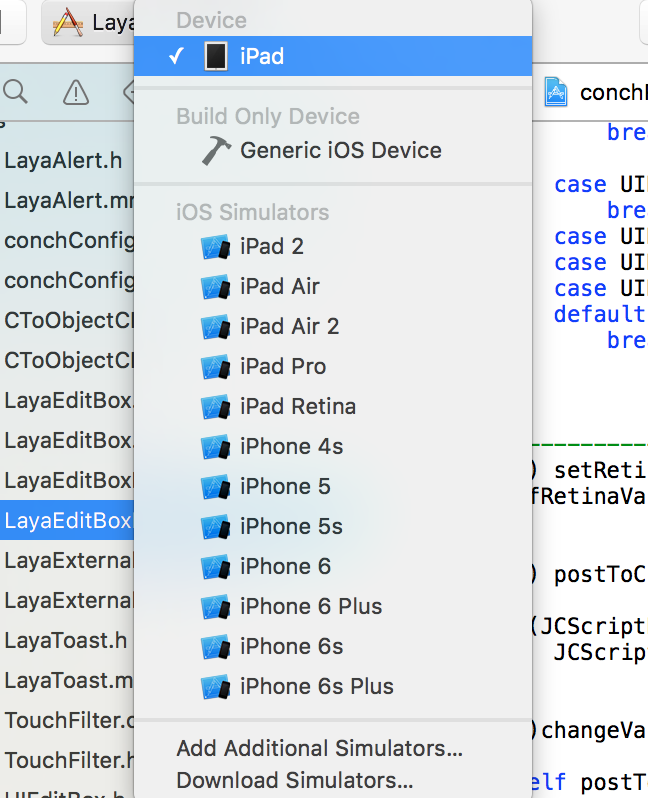
https://developer.apple.com/library/mac/documentation/NetworkingInternetWeb/Conceptual/NetworkingOverview/UnderstandingandPreparingfortheIPv6Transition/UnderstandingandPreparingfortheIPv6Transition.html#//apple\_ref/doc/uid/TP40010220-CH213-SW1

http://www.pchou.info/ios/2016/06/05/ios-supporting-ipv6.html?utm\_source=tuicool&utm\_medium=referral

# 18、编译项目

android项目，直接导入到eclipse或者android-studio即可。

ios项目，打开testLayaBox(或者你自己的项目目录).xcodeproj文件，打开后选择真正的ios设备进行build。（注意：真正的设备是armv7、armv7s、arm64架构，如果是ios Simulator这些都是x86架构，目前LayaPlayer在ios设备上尚未支持x86架构，如果选择模拟器编译是无法通过的）。



ios如果想真机运行调试，需要开发者去苹果购买开发者账号、并设置程序签名等等，这部分请开发者自行学习。

# 19、Logo界面

app启动会先下载html，然后进行解析，再下载、执行js，这个过程需要一点时间，所以在LayaPlayer中，默认项目显示了一个LayaBox的动画。

## 19.1进度条控制

Logo界面默认播放完这个动画，进入游戏，这个logo动画中还可以配置加载进度，如下代码所示

if(window.loadingView)

{

window.loadingView.loading(100);//设置 0 - 100 之间的值，当这个值达到100的时候loadingView消失，显示游戏画面

}

其他设置方法，在script/config.js中，开发者可以根据自己需求进行调用相应方法，如下代码所示：

loadingView.loadingAutoClose=true;

loadingView.bgColor("#ffffff");//设置背景颜色

loadingView.setFontColor("#000000");//设置字体颜色

loadingView.setTips(["新世界的大门即将打开", "敌军还有30秒抵达战场"]);//可以设置一个数组，这个数组中的文字会自动切换。

loadingAutoClose默认这个变量为true,当LayaBox的动画播放完一遍，就会自动把这个logo界面隐藏掉了，如果开发者想要精准控制，需要把这个变量设置为false，通过window.loadingView.loading(100)，进行精确控制。

## 19.2 替换掉开发者的自己的Logo

如果开发者想要把LayaBox的动画替换到成自己的Logo，需要做好一张图片放到resource/logo/logo.png （注意必须为png），这样就不会加载LayaBox的动画了，其他设置和原来的一样，设置背景色、tips、fontColor等。

## 19.3 想要做特别酷炫的进度条怎么办？

在实际项目中，如果开发者想要做出自己喜欢的酷炫的进度条，LayaPlayer现有的方案是不够满足的，建议开发者快速加载LayaAir-JS引擎和必备的图片，通过LayaAir自己实现酷炫的进度条。

## 19.4 后续白名单功能

后续LayaBox会有白名单机制，如果开发者购买了授权或者和LayaBox联合运营产品，便可以去掉LayaBox的logo，如果没有则需要强制增加LayaBox的logo。引擎内部会有检测机制，随机检测，如果检测不通过则无法进入游戏。

# 20、注意事项

文本格式的文件（例如:ini、xml、html、json、js等）都必须是utf8编码格式，因为iOS设备现在尚不支持非utf8格式编码的文件。

在LayaPlayer中背景音乐目前只支持mp3格式，并且不能同时播放多个。 音效可同时播放多个，但是为了效率考虑，目前只支持wav格式，具体格式为PCM格式、单声道、16位、22050采样率

mp3转wav示例教程<http://www.layabox.com/html/docments/71.html>