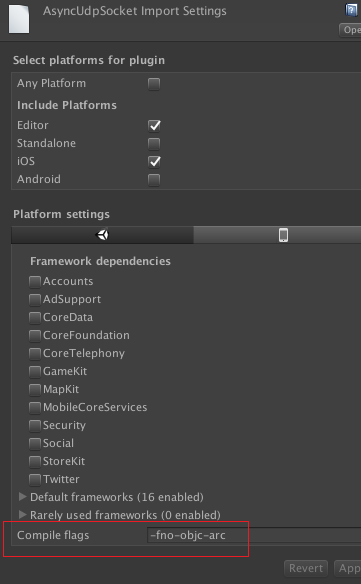
# 1

## REF

### Unity导出XCode修改单个文件编译选项

利用Unity自己的机制：

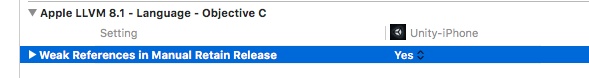


### Unity导出XCode8编译（不要自动签名）

ProvisioningStyle = Manual

<http://blog.icodeten.com/game/2016/12/19/unity-xcode8/>

### [cannot synthesize weak property in file using manual reference counting](http://blog.csdn.net/u010536615/article/details/52170534)



### 去掉Xcdoe8自动签名

<http://www.th7.cn/Program/IOS/201701/1083922.shtml>

<http://stackoverflow.com/questions/39444020/how-to-disable-automatically-manage-signing-in-xcode-8>

xcodebuild -project "/platforms/ios/test.xcodeproj" -sdk "iphoneos10.0" -configuration "Release" -scheme test ENABLE\_BITCODE="NO" install DSTROOT="/target/debug" DEVELOPMENT\_TEAM="teamId" PROVISIONING\_PROFILE\_SPECIFIER="teamId/provisioningProfileName"

### 球球讲座

网络球-->物理球-->渲染球

逻辑和渲染分开，逻辑更新，渲染不一定更新

划分九屏

加载的时候把mesh拿出来，批量绘制

把复杂的球用图片模拟，降低面片数？

### Unity官方指导

1，Lightmap静态烘焙打assetbundle效果异常

2，UI合批 (按钮分开自动合)

3，静态和动态放在不同的canvas下

4，特效放到UI上

5，特效和模型相互包围，想bloom只对模型起效果，对特效不起效果

6，已经发射出的粒子不消失

7，连续动画A-->B-->C B播到一半再从头播会卡顿(Animator)

8，Default UI Shader其实也可以替换的，用更简单的shade

9，FontEffect很耗，建议用更专业的插件

10，UI改用TextureMesh？

### Xcode命令行

security unlock-keychain -p "123456" "/Users/Shared/Jenkins/Library/Keychains/login.keychain"

xcodebuild -target Unity-iPhone -verbose -configuration Release CODE\_SIGN\_IDENTITY="iPhone Distribution: Shanghai Giant Network Technology Co., Ltd." -jobs 8 -project /xxx/yyy/Unity-iPhone.xcodeproj

xcrun -sk iphoneos PackageApplication -v /xxx/yyy/proj\_ios/build/Release-iphones/tank.app -o ../2017.ipa --sign "iPhone Distribution: Shanghai Giant Network Technology Co., Ltd." --embed /path/allToOne.mobileprovision

### XUPorter更改codesign 和Provisioning profile

Project.overwriteBuildSetting(“” , “iPhone Developer:jsjkxkx (xxxx)”, “Debug’)

CODE\_SIGN\_IDENTITY

ROVISIONING\_PROFILE

PROVISIONING\_PROFILE\_SPECIFIER

### XUPorter怎么更改单个文件的编译选项

### svn转Git

<http://www.aikaiyuan.com/6584.html>

git svn clone https://localhost:8443/svn/www/  www

git remote -v

git remote add [git@gitlab.xxx.com:second/test.git](mailto:git@gitlab.xxx.com:second/test.git)

git push

## 语音封装

### IOS录音老出错

要手动设置权限：

<http://www.jianshu.com/p/c212cde86877>

iOS 10 开发适配系列 之 权限Crash问题

解决方案  
1.在项目中找到info.plist文件，右键点击以 Source Code形式打开  
2.添加以下键值对，这里以 PhotoLibrary 作为例子

<key>NSPhotoLibraryUsageDescription</key>

<string>此 App 需要您的同意才能读取媒体资料库</string>

### XCode 同一Project创建多个target

http://blog.163.com/lengfeng\_04/blog/static/80470603201273111549529

## 语音SDK

### 巨人语音

目前问题

* 性能问题
* 杂音
* 背景音转到听筒质量差

registercallback-》initsdk-》setparam-》setuserinfo-》customroomserver->   joinroom-》

### AndroidStudio ndkbuild

<https://developer.android.com/studio/projects/add-native-code.html#link-gradle>

### NDK 中 Android.mk 头文件及代码路径问题

<http://oknostar.blog.163.com/blog/static/294361862012611111539579/>

### Android studio编译  audiowrapper.c出错，找不到头文件

把

#include **"iLBC\_define.h"**#include **"iLBC\_decode.h"**#include **"iLBC\_encode.h"**

改成

#include **"ilbc\_src/iLBC\_define.h"**#include **"ilbc\_src/iLBC\_decode.h"**#include **"ilbc\_src/iLBC\_encode.h"**

就OK了

### [error: 'retain' is unavailable: not available in automatic reference counting](http://blog.csdn.net/wengyupeng/article/details/52005436)

<http://blog.csdn.net/chengyakun11/article/details/8124463>

选中工程->TARGETS->相应的target然后选中右侧的“Build Phases”，向下就找到“Compile Sources”了。然后在相应的文件后面添加:-fobjc-arc参数 、-fno-objc-arc参数

arc项目中引用非arc代码   加上“-fno-objc-arc”

非arc项目中引用arc代码 加上“-fobjc-arc”

### 编译运行单个java文件

<https://zhidao.baidu.com/question/326559137.html>

### Java按钮点击

<http://www.cnblogs.com/McCa/p/5018851.html>

**btnStartRecord** = (Button) findViewById(R.id.***startRecord***);  
**btnStartListen** = (Button) findViewById(R.id.***startListen***);  
**btnStopRecord** = (Button) findViewById(R.id.***stopRecord***);  
**btnStopListen** = (Button) findViewById(R.id.***stopListen***);  
**ipEditText** = (EditText) findViewById(R.id.***edittext\_ip***);

### Java调用C

**public static native int** audio\_codec\_init(**int** mode);  
  
*// encode***public static native int** audio\_encode(**byte**[] sample, **int** sampleOffset,  
 **int** sampleLength, **byte**[] data, **int** dataOffset);  
  
*// decode***public static native int** audio\_decode(**byte**[] data, **int** dataOffset,  
 **int** dataLength, **byte**[] sample, **int** sampleLength);

jint Java\_xmu\_swordbearer\_audio\_AudioCodec\_audio\_1encode(JNIEnv \*env,  
 jobject **this**, jbyteArray sampleArray, jint sampleOffset,  
 jint sampleLength, jbyteArray dataArray, jint dataOffset)

### Android studio查看jar包代码

右键jar包-🡪add to library

### Java端口

A:

socketA = **new** DatagramSocket(**myPortA**);

B:

socketB = **new** DatagramSocket(**myPortB**);

A发送给B

dataPacket = **new** DatagramPacket(data, size, **ipB**, **myPortB**);

socketA.send(**dataPacket**);

A接受B发的信息

DatagramPacket **packet**;

socketA.receive(**packet**);

### UnityPlayer.UnitySendMessage

### 应用TortoiseGit为github账号添加SSH keys

<http://jingyan.baidu.com/article/63f236280f7e750209ab3d60.html>

### [Unity3D中C#调用iOS的静态库(\*.a)](http://blog.csdn.net/skylin19840101/article/details/51039176)

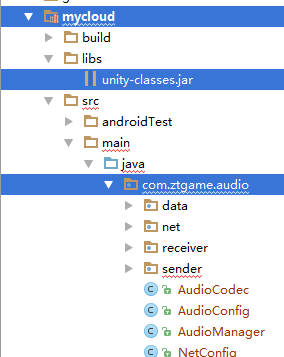
<http://blog.csdn.net/skylin19840101/article/details/51039176>

### [\*\*\*\*\*AS生成有依赖关系的jar包\*\*\*\*\*](http://blog.csdn.net/skylin19840101/article/details/51039176)

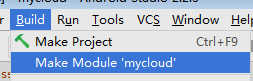
生成new module

添加依赖库unity-classes.jar放在lib目录：

把java代码放到指定的目录com.ztgame.audio

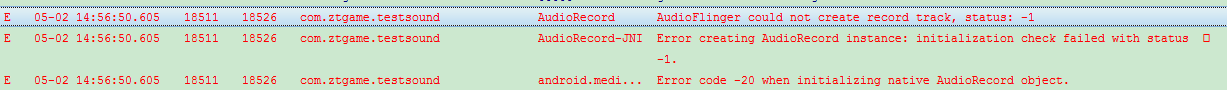


Build🡪make module



# 花时间问题

## UnityDemo调用录音出错



1. 好像是没权限。。

<http://stackoverflow.com/questions/20693001/audioflinger-could-not-create-record-track-status-1-need-help-to-ifx>

\*\*\*\*\*要想起百度或谷歌\*\*\*\*\*

### 111

## 22

### 111

# 优化

## 纹理

1、严格控制RGBA32和ARGB32纹理的使用，在保证视觉效果的前提下，尽可能采用“够用就好”的原则，降低纹理资源的分辨率，以及使用硬件支持的纹理格式。

2、在硬件格式（ETC、PVRTC）无法满足视觉效果时，RGBA16格式是一种较为理想的折中选择，既可以增加视觉效果，又可以保持较低的加载耗时。

3、严格检查纹理资源的Mipmap功能，特别注意UI纹理的Mipmap是否开启。在UWA测评过的项目中，有不少项目的UI纹理均开启了Mipmap功能，不仅造成了内存占用上的浪费，同时也增加了不小的加载时间。

4、ETC2对于支持OpenGL ES3.0的Android移动设备来说，是一个很好的处理半透明的纹理格式。但是，如果你的游戏需要在大量OpenGL ES2.0的设备上进行运行，那么我们不建议使用ETC2格式纹理。因为不仅会造成大量的内存占用（ETC2转成RGBA32），同时也增加一定的加载时间。下图为测试2中所用的测试纹理在三星S3和S4设备上加载性能表现。可以看出，在OpenGL ES2.0设备上，ETC2格式纹理的加载要明显高于ETC1格式，且略高于RGBA16格式纹理。因此，建议研发团队在项目中谨慎使用ETC2格式纹理。