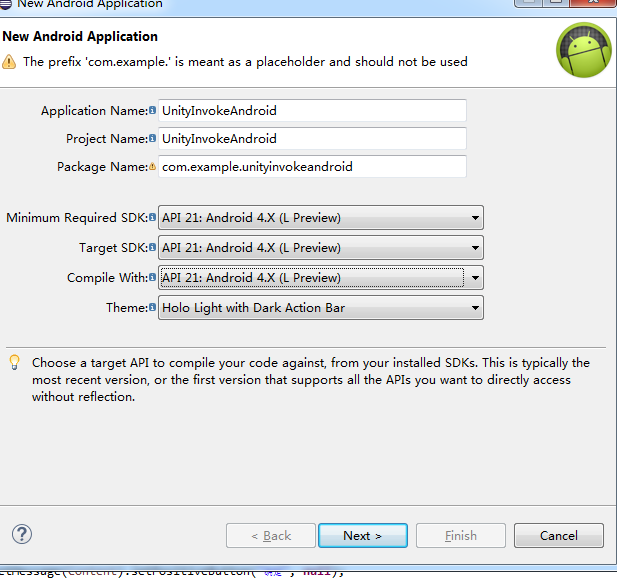
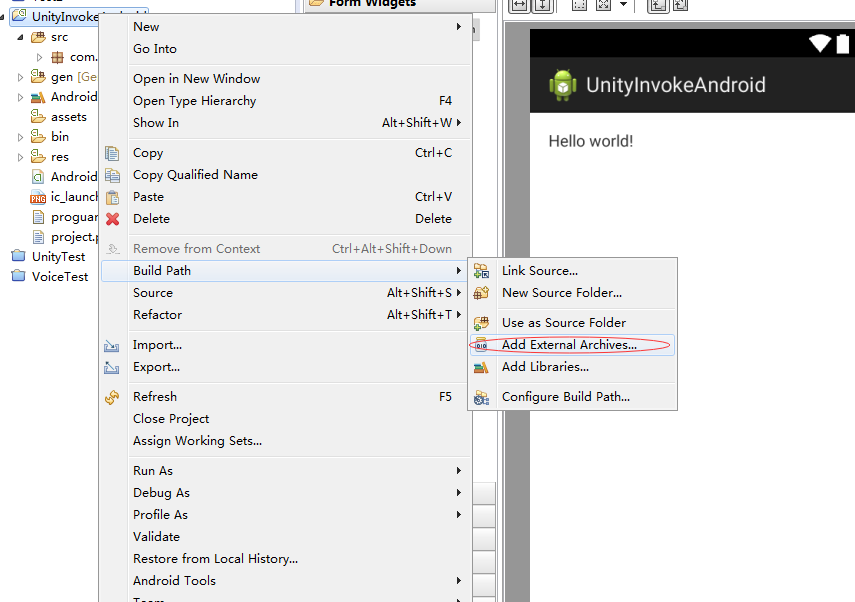
# Unity Invoke Andorid(UIA)

1、新建一个Android项目

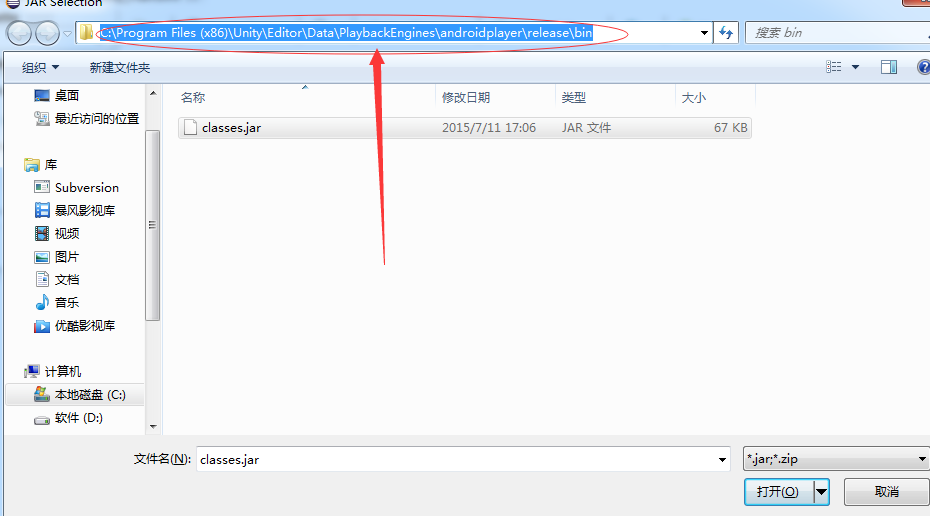


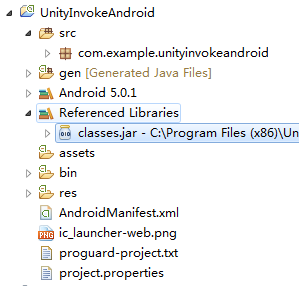
1. 右键项目 build path --- >add external archives



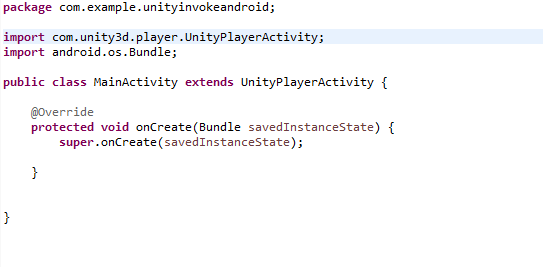
选择unity安装目录下的class.jar文件。参考下图路径找自己的安装路径

C:\Program Files (x86)\Unity\Editor\Data\PlaybackEngines\androidplayer\release\bin，点击打开后自动载入class.jar文件。



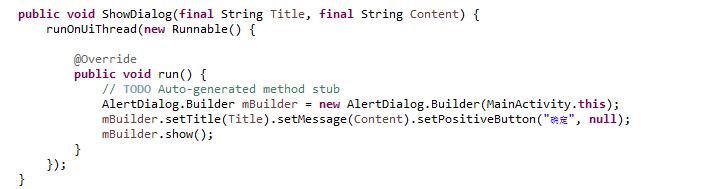


1. 打开MainActivity, 删掉一部分（删掉的部分仅仅是在Android上显示有用），将extends的Activity 改为 UnityPlayerActivity。 如下图所示

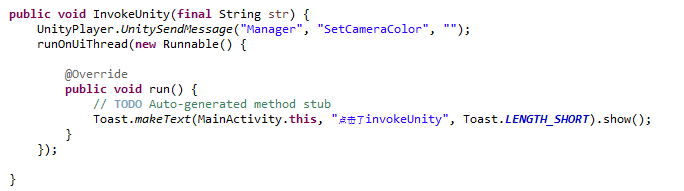


1. 开始编写代码

显示对话框



Android发送消息给unity



Android向Unity工程中发送消息主要是通过下面的接口来实现——

UnityPlayer.UnitySendMessage(“参数一：Unity场景中的GameObject”，“参数二：Unity场景中脚本文件中的回调接口”，“参数三：Android工程向Unity工程发送的字符串”);

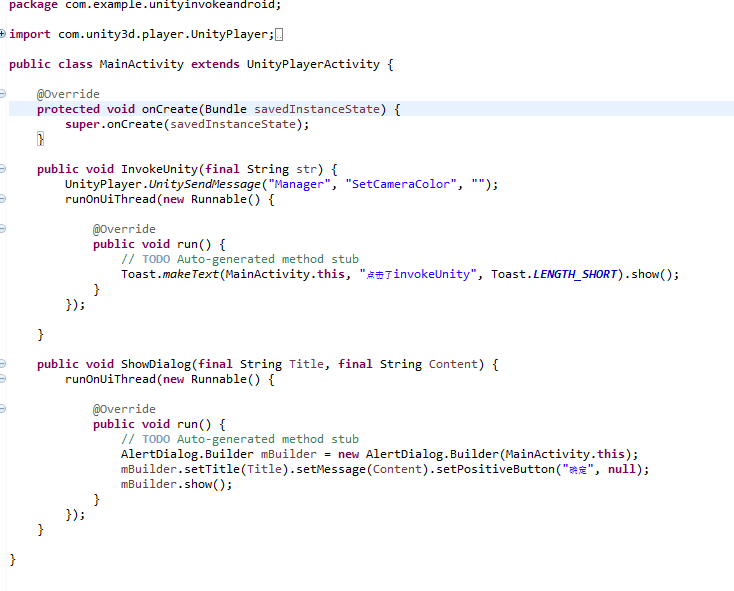
这里需要说明下：

参数一：我们需要在Unity场景中创建一个GameObject并保证它一直存在与Unity场景中，否则可能会出现Android工程在发送消息的时候找不见这么一个对象，造成崩溃；

参数二：我们要为参数一的GameObject绑定一个脚本文件，并在该脚本文件中定义接收Android工程传来消息的接口；

参数三：接收消息的接口参数必须为string类型才能成功接收，其他类型的数据需要通过string类型的转换来传递。

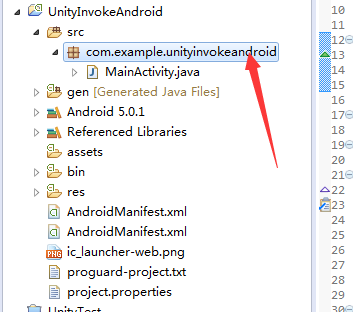
完整代码：



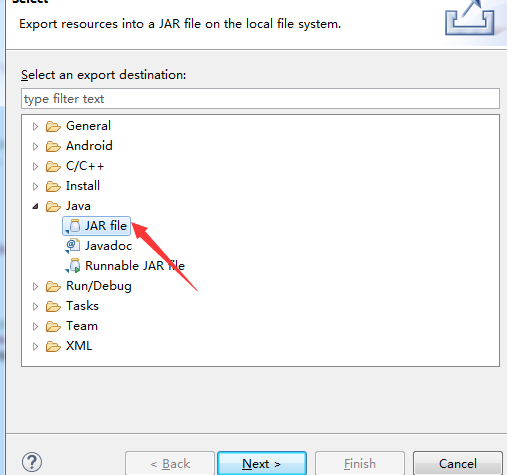
在AndroidManifest中添加上如下代码

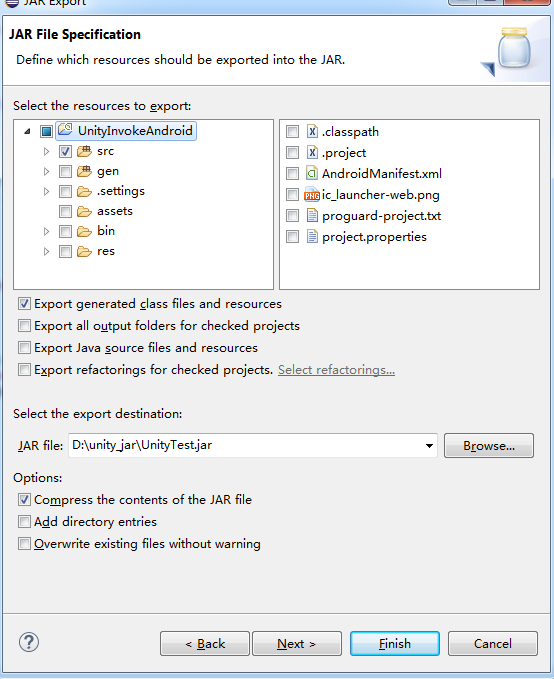


1. 打包。点击项目，选择project，选择clear。然后选择箭头指向的包，



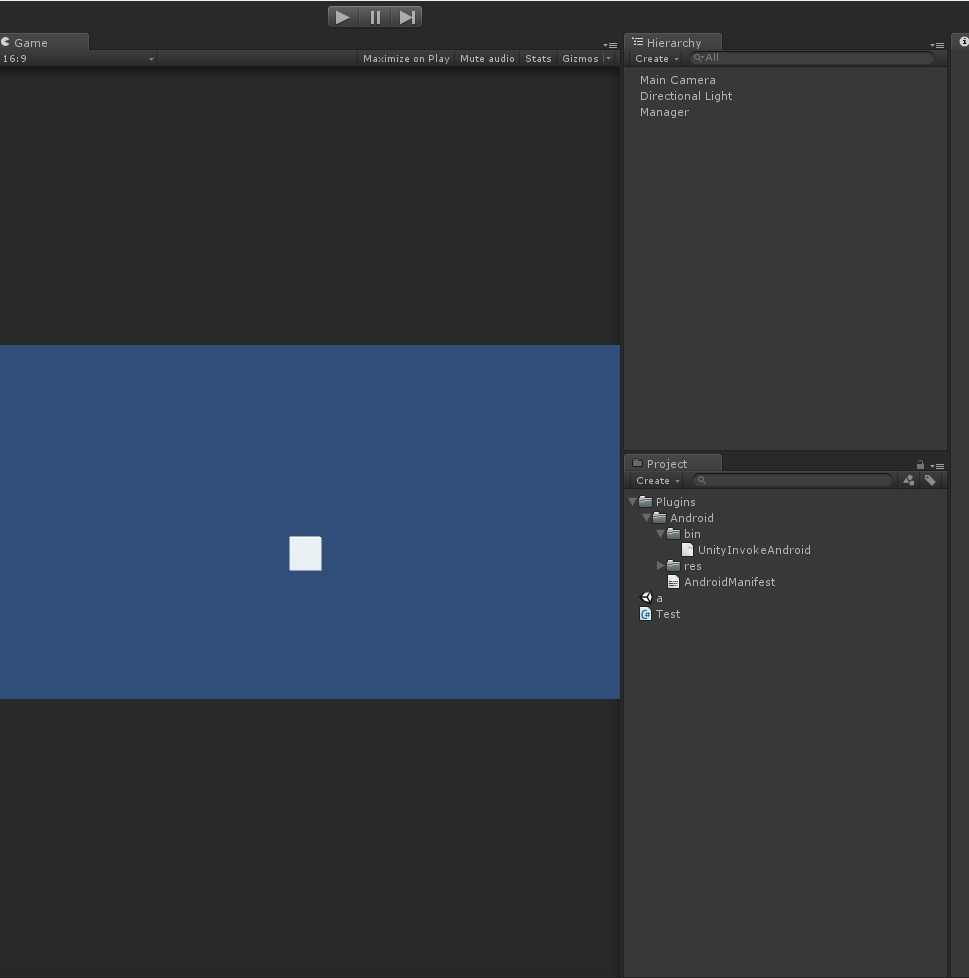
右键，export。选择Java-->jar file





Jar file 的位置自定义，名字也自定义，最后点击finish。

1. Android部分完成，打开unity，新建一个项目



目录结构如下

--Plugins

--Android

--bin

--libs

--res

--AndroidManifest

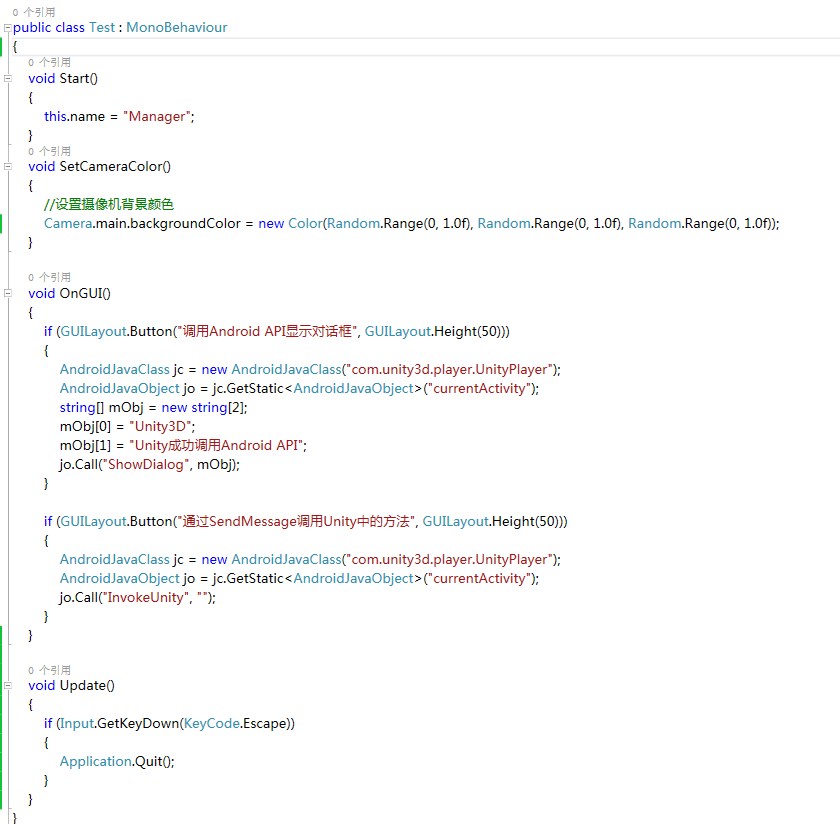
Bin文件夹里放之前打包好的jar文件，示例中是UnityInvokeAndroid.jar

Libs 文件夹是当使用第三方SDK时可用，直接从eclipse中拖过来即可

Res 直接从eclipse中拖过来即可

AndroidManifest 直接从eclipse中拖过来即可

1. 在场景中创建一个名为Manager的游戏对象，然后绑定脚本

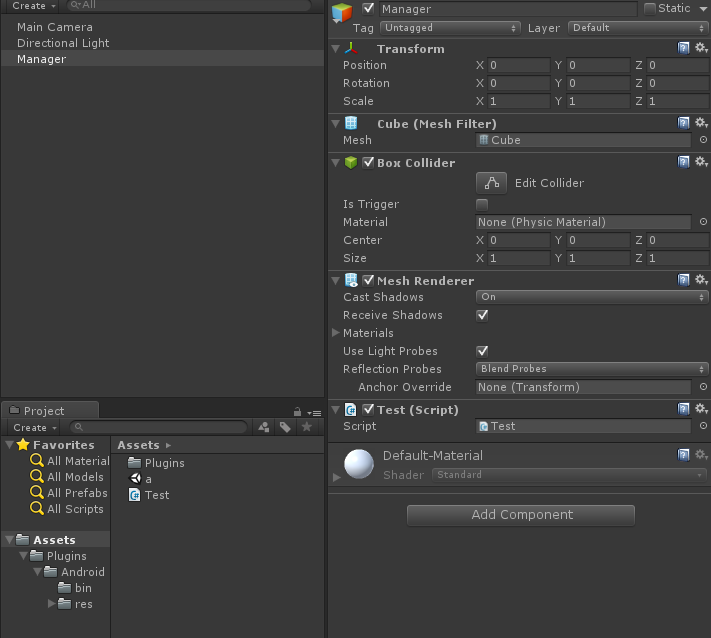


AndroidJavaClass jc = new AndroidJavaClass("com.unity3d.player.UnityPlayer"); (获取unity的类)

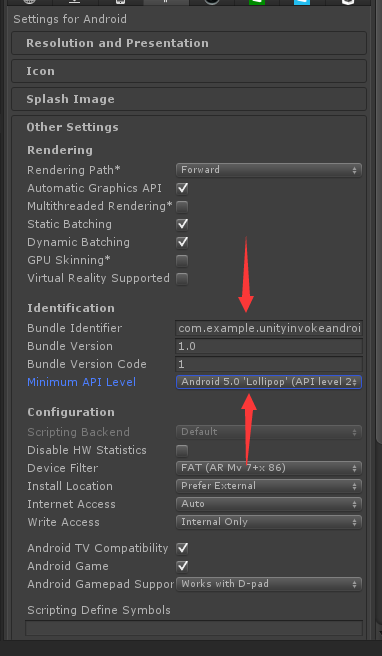
AndroidJavaObject jo = jc.GetStatic<AndroidJavaObject>("currentActivity");(获取当前的Activity)

Jo.call() --- >使用unity调用Android中的方法。jo.Call的第一个参数是Android工程中的接口名，即方法名，第二个参数是需要向Android工程传递的参数，这里需要提一下，Unity向Android传递消息仍然是以string类型来传递的，需要其他类型的话，我们需要在Android工程中进行转换。

绑定脚本。



1. 修改配置



上面的修改成Android工程名，api版本选择21。

10、build and run。一定要在真机上运行。