

参考链接:
<https://medium.com/@davidbeloosesky/embedded-unity-within-android-app-7061f4f473a>

- 软件环境:
1. Unity5.4.5f1
 2. Android Studio 2.3.3

- 整体步骤:
1. 导出Unity工程为Android工程
 2. 用Android Studio导入Unity导出的Android工程, 并导出为AAR文件
 3. 在原生工程导入AAR文件并打包测试

详细步骤:

- 1.用Unity准备一个测试工程, 测试工程中最好添加一个脚本, 监听安卓返回键, 可以退出当前Unity场景.

具体代码为:

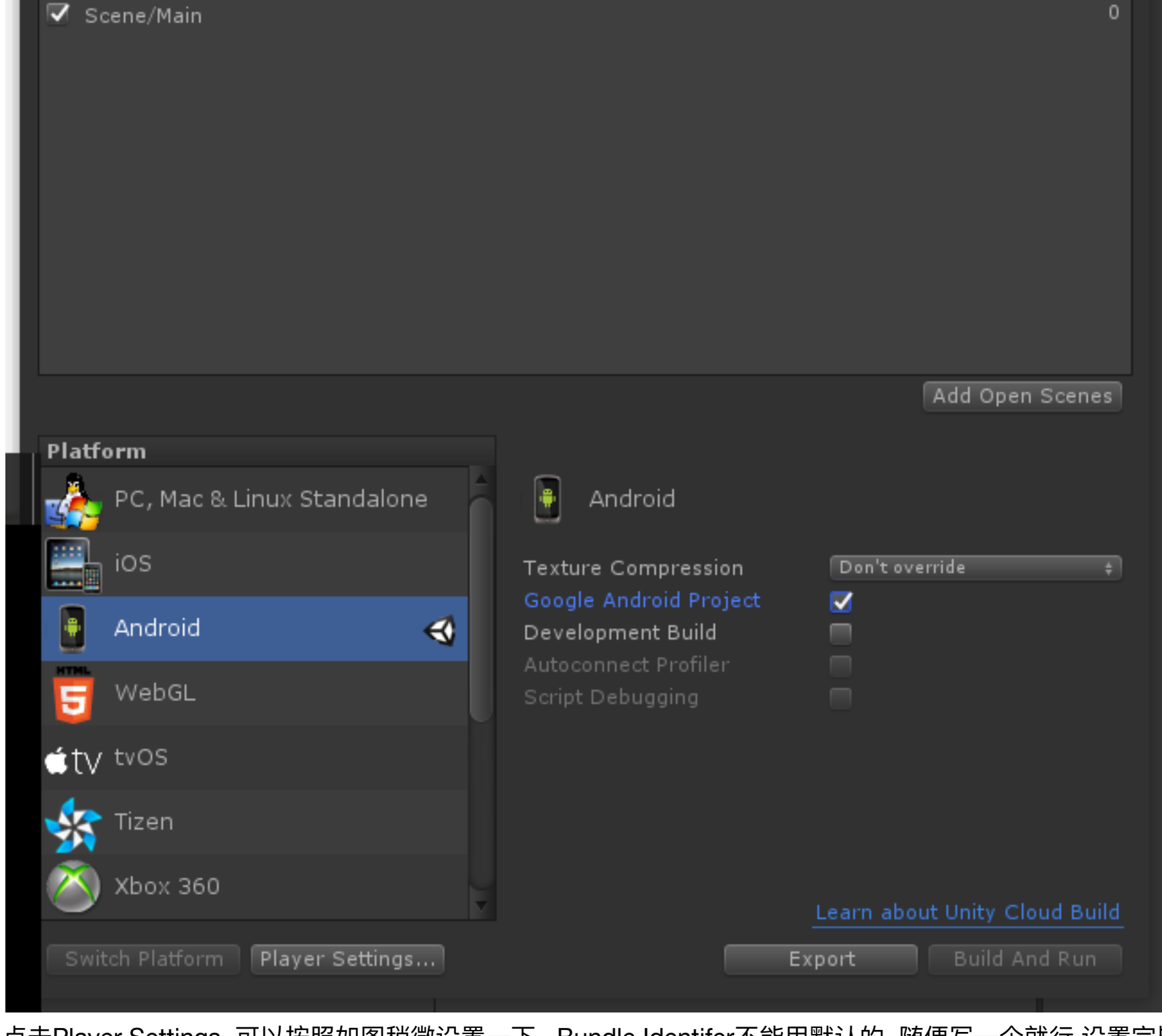
```

Public class SupportBackScript : MonoBehaviour
{
    void Update()
    {
        if(Application.platform == RuntimePlatform.Android)
        {
            if(Input.GetKey(KeyCode.Escape))
            {
                Application.Quit();
            }
        }
    }
}

```

然后准备导出Unity工程, 打开Player Build Settings界面, 呼出路径为, 左上角File>BuildSettings, 记得把测试场景添加上.

平台选择Android, 勾选上Google Android Project 选项如图:



点击Player Settings, 可以按照如图稍微设置一下., Bundle Identifier不能用默认的, 随便写一个就行,设置完毕后, 就可以点击Export导出了.



2.目前Unity版本只能导出为Eclipse工程, 所以需要Android Studio导入然后自动转化一下, 第一次导入的时间比较长.

导入完毕后:

- 打开Mainfest, 首先在activity中增加配置: android:process=":unity", 这个属性的意思是把UnityActivity当成一个新的进程处理.
- 由于UnityActivity退出的时候会杀掉当前进程, 所以把UnityActivity放在新进程就算被杀掉了也不会影响原生App的进程.
- 然后注释掉以下代码:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.your.package.name">
    <supports-screens android:smallScreens="true" android:normalScreens="true" android:largeScreens="true" android:xlargeScreens="true">
    </supports-screens>
    <application android:theme="@style/UnityThemeSelector" android:icon="@drawable/ic_launcher" android:label="@string/app_name">
        <activity android:process=":unity" android:label="IntegrationUnity" android:exported="true">
            <!--<intent-filter>-->
            <!--<action android:name="android.intent.action.MAIN" />-->
            <!--<category android:name="android.intent.category.LAUNCHER" />-->
            <!--</intent-filter>-->
            <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="19" android:targetSdkVersion="25" />
    <uses-feature android:glEsVersion="0x0020000" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-feature android:name="android.hardware.touchscreen" android:required="false" />
    <uses-feature android:name="android.hardware.touchscreen.multitouch" android:required="false" />
    <uses-feature android:name="android.hardware.touchscreen.multitouch.distinct" android:required="false" />
</manifest>

```

- 如图, 打开build.gradle文件, 第一行注释改成第二行的样子, 然后把defaultConfig里面注释掉 applicationId这样.

```

//apply plugin: 'com.android.application'
apply plugin: 'com.android.library'

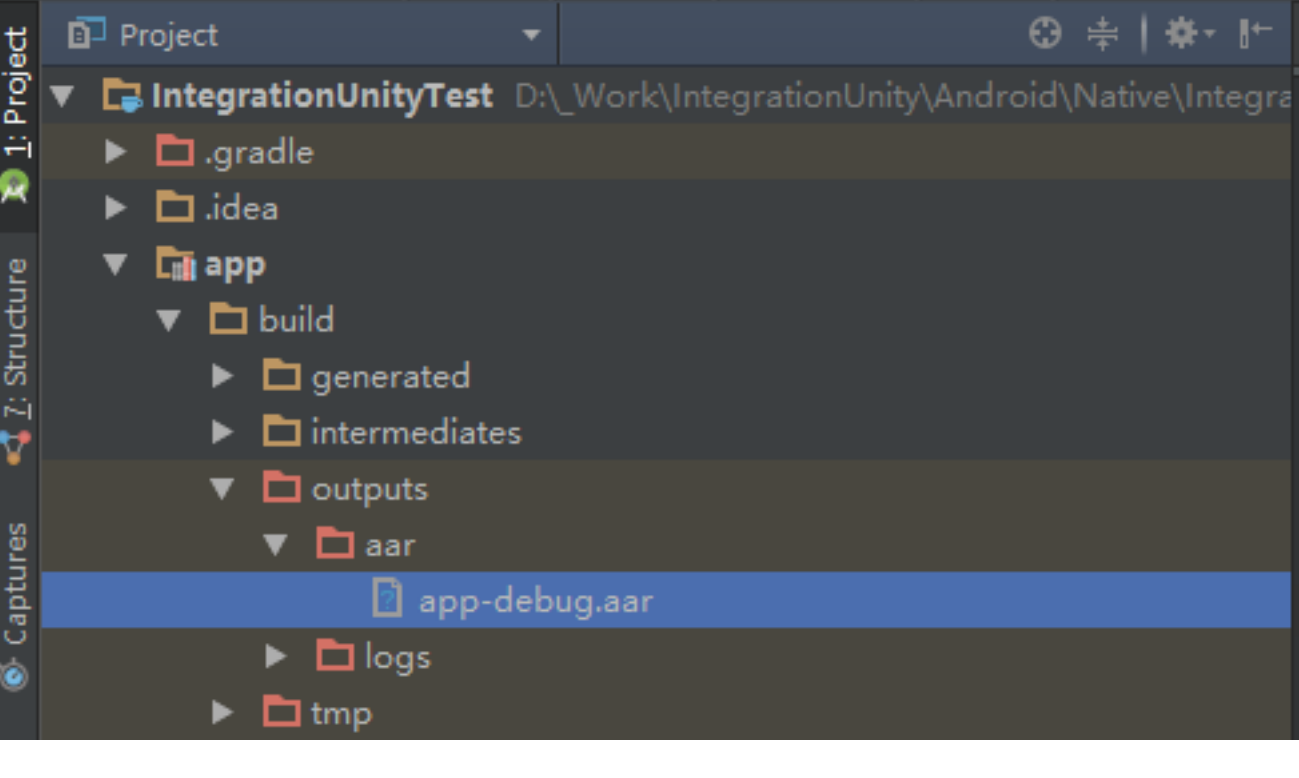
android {
    compileSdkVersion 25
    buildToolsVersion "26.0.1"

    defaultConfig {
        //applicationId "com.your.package.name"
        minSdkVersion 19
        targetSdkVersion 25
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

```

- 接着就可以点击Build-Make Project, 完毕后, 如图就能看到outputs下打包好的文件了.



3.打开一个原生Android应用的工程, 点击File > New Module >Import JAR/AAR Package, 选择你之前Build好的AAR文件.

- 确保settings.gradle文件内是 : include 'app', 'your_aar_file_name'
- 如图, 打开build.gradle(Module:app), 添加红框内的内容: compile project("your_aar_file_name"), 然后会提示Sync gradle, 点击同步即可

```

//apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    buildToolsVersion "26.0.1"
    defaultConfig {
        applicationId "com.puhanda.nativeappaar"
        minSdkVersion 19
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    dependencies {
        compile project(":WebGLTest-debug")
        compile fileTree(dir: 'libs', include: ['*.jar'])
        androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
            exclude group: 'com.android.support', module: 'support-annotations'
        })
        compile 'com.android.support.constraint:constraint-layout:1.0.2'
        testCompile 'junit:junit:4.12'
    }
}

```

由于aar文件内含有unity的mainfest文件, 当run工程的时候会尝试合并mainfest, 但是默认会出现Manifest merger failed with multiple error, 参见logs错误.

需要在原生Android App中的Mainfest中稍微设置一下.

- 添加mainfest tag: xmlns:tools="http://schemas.android.com/tools"
- 添加application tag: tools:replace="android:icon,android:theme"
- 效果就是如下:

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.yourpackage.name">

    <application
        android:allowBackup="true"
        tools:replace="android:icon,android:theme"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        ...
    </application>
</manifest>

```

最后就剩下添加打开UnityActivity的方法了, 打开Activity的代码如下:

```

import com.yourpackage.name.UnityPlayerActivity;
...

// DON'T CALL IT FROM YOUR MAIN ACTIVITY - SEE ABOVE INSTRUCTIONS!
Intent intent = new Intent(mContext, UnityPlayerActivity.class);
startActivity(intent);

```

备注:

如果出现以下为题:

1.出现这个错误, 高版本Unity导出工程的时候会有Gradle选项, 这个相当于是Android studio工程的目录结构. 但是Unity默认的Gradle是2.1.0版本的, 如果出现以下错误, 则只需要把版本改成2.2.0即可, 路径是build.gradle (app module):

```

dependencies {
    classpath 'com.android.tools.build:gradle:2.2.0'
}

```

Error:org.gradle.api.internal.tasks.DefaultTaskInputs\$TaskInputUnionFileCollection cannot be cast to org.gradle.api.internal.file.collections.DefaultConfigurableFileCollection
Possible causes for this unexpected error include:

- Gradle's dependency cache may be corrupt (this sometimes occurs after a network connection timeout.)

2. 如果任何人在将Unity活动导入主项目时遇到问题, 可能是因为在构建期间类文件不会导出到AAR文件.

可以在Unity导出的Android的工程中的proguard-rules.txt文件中加入: -keep public class my.package.* { public; }即可