PlaySDK开发接口文档

(Android版)

2016年01月08日

|  |  |  |
| --- | --- | --- |
| 文件状态：  [] 草稿  [√] 正式发布  [] 正在修改 | 文件标识： |  |
| 当前版本： | V1.4 |
| 完成日期： | 2016-01-08 |

**文档版本历史**

|  |  |  |
| --- | --- | --- |
| 版本 | 起止日期 | 备注 |
| 1.0 | 2015-9-21 | 初稿 |
| 1.1 | 2015-10-13 | 1.新添加停止录制且不立即保存视频之后再调用新接口保存视频，参见4.15  2.新添加停止录制且不立即保存视频之后再调用新接口保存视频且进入视频分享界面, 参见4.16  3.根据游戏界面进行横竖屏适配  4.修复红米HM2A录制视频停滞在第一帧的bug和联想K50录制视频花屏的bug. |
| 1.2 | 2015-12-04 | 文档内容精简，及根据需求方要求规范格式。 |
| 1.3 | 2015-12-18 | 在V1.2基础上，添加一些集成使用实例。 |
| 1.4 | 2016-01-08 | 在V1.3基础上，添加一些集成使用实例(AndroidManifest.xml文件修改)。 |

目录

[1 引言 5](#_Toc440039614)

[1.1介绍 5](#_Toc440039615)

[1.2术语及缩略词 5](#_Toc440039616)

[2SDK的简要介绍 5](#_Toc440039617)

[2.1关于视频的录制模式 6](#_Toc440039618)

[2.2 SDK大致工作流程 6](#_Toc440039619)

[3 SDK的集成方法/步骤 7](#_Toc440039620)

[3.1 SDK集成包文件的说明 7](#_Toc440039621)

[3.2 集成步骤1 –游戏工程中添加SDK文件(复制+粘贴) 7](#_Toc440039622)

[3.3 集成步骤2–修改游戏工程的AndroidManifest.xml文件 7](#_Toc440039623)

[3.4 集成步骤3 –修改游戏工程主Activity的java文件 8](#_Toc440039624)

[4 SDK调用详细接口 9](#_Toc440039625)

[4.1 init -SDK的初始化 9](#_Toc440039627)

[4.2 set/get-RecordingMode -设置/获取录制模式 11](#_Toc440039628)

[4.3 startRecording -启动视频录制 12](#_Toc440039629)

[4.4 stopRecording -停止视频录制并保存视频文件 12](#_Toc440039630)

[4.5 clipRecording -保存精彩片段模式视频片段 13](#_Toc440039631)

[4.6 PlayRecVideoSaveListener接口说明 14](#_Toc440039632)

[4.7 getVideoLocalPath -获取视频的本地地址 14](#_Toc440039633)

[4.8 isRecording -是否处于录制状态 15](#_Toc440039634)

[4.9 exit -SDK退出接口 15](#_Toc440039635)

[4.10 PlayRec.PlayRecListener说明 16](#_Toc440039636)

[4.12 showWin -开启社区界面 16](#_Toc440039637)

[4.13 videoPreview -开启视频预览界面 17](#_Toc440039638)

[4.14 hideFloatingController -隐藏SDK的浮窗 17](#_Toc440039639)

[4.15 showFloatingController -显示SDK的浮窗 18](#_Toc440039640)

[4.16 saveVideoAfterStopRecording -停止录制且不立即保存视频之后再调用此接口保存视频 18](#_Toc440039641)

[4.17 showShareVideoWindowAfterStopRecording -停止录制且不立即保存视频之后再调用此接口保存视频且进入视频分享界面 19](#_Toc440039642)

[5特殊录制场景设置介绍 20](#_Toc440039643)

[6使用示例 20](#_Toc440039644)

[6.1 使用实例1（最简单情况） 20](#_Toc440039645)

[6.2 使用实例2（有初始化调试信息打印） 20](#_Toc440039646)

[6.3使用实例3（有特殊定制设置/示例1） 21](#_Toc440039647)

[6.4使用实例4（对AndroidManifest.xml文件的修改） 22](#_Toc440039648)

# 1 引言

## **1.1介绍**

本文档的目的是为咪咕互动娱乐有限公司的手机游戏视频录制SDK提供集成技术说明指引；为了文档简洁，文档后续简称咪咕互动娱乐有限公司的手机游戏视频录制SDK为PlaySDK。

本文档的内容主要包括几方面的内容：手机游戏视频录制SDK的简要介绍，SDK的集成方法/步骤，SDK调用接口详细介绍，以及特殊录制场景设置介绍。

## **1.2术语及缩略词**

|  |  |
| --- | --- |
| 名词/缩略词 | 描述 |
| CP | Content Provider，内容提供商（就是您游戏集成/开发商） |
| SDK | Software Development Kit 的缩写，中文即“软件开发工具包”。  本文档中的SDK特指咪咕互动娱乐有限公司的手机视频录制SDK。 |

# 2 SDK的简要介绍

咪咕互动娱乐有限公司的视频录制SDK是一个可以动态升级/加载的插件，游戏内容提供商（以下简称CP）的游戏通过几步简单的复制+粘贴以及两处字符串改动后，就可以实现录制游戏视频以及让玩家上传/共享这个视频。以下为我们对视频录制SDK优点的简要描述：

适配度高，无缝适配Android4.0及其以上系统手机；

兼容性好，同时支持cocos2d-x, unity3D, 自主引擎；

录制流畅，边玩边录，视频同步生成，无需等待压缩；

分享快捷，用户把视频分享到QQ、微信等社交平台实现自传播；

集成简便，几行代码完成SDK集成，而且无需root权限；

## **2.1关于视频的录制模式**

视频录制SDK录制游戏视频时，处于两种录制模式中的一种：

* 普通录制模式

用户手动控制(或者CP程序控制)开始/停止录制；当用户手工停止(或CP程序控制停止)录制的同，保存开始录制后缓存下来的视频文件。

* 精彩片段模式

SDK初始化后即自动启动录制（后台循环录制）；当用户点击保存按钮时，保存最近一小段时间的视频。

## **SDK大致工作流程**

视频录制SDK的大致工作流程为：初始化＞ 开始录制 ＞ 停止录制 ＞ 保存视频 ＞ 进入社区(上传/分享)；更详细的流程图以后补充。

# 3 SDK的集成方法/步骤

## **3.1 SDK集成包文件的说明**

SDK集成包(zip)文件包含以下内容：

* 本集成介绍文档；
* 视频录制SDK文件，共两个文件3.2.1和3.2.2用到/提及；

## **3.2 集成步骤1 –游戏工程中添加SDK文件(复制+粘贴)**

**3.2.1**复制playsdk/libs目录下文件(dlsdk-lib-2.jar一个文件)到游戏工程的libs目录下。

**3.2.2** 复制playsdk/assets/playsdk目录下文件（magic\_single\_sdk.bin一个文件）到游戏工程的assets/playsdk目录下(需要您在assets目录下创建playsdk子目录)。

## **3.3** **集成步骤2–修改游戏工程的AndroidManifest.xml文件**

**3.3.1** 添加SDK需要的权限(复制+粘贴)

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.GET\_TASKS" />

**3.3.2** 添加SDK的Activity和Service(复制+粘贴)

<activity android:name="com.wonxing.dynamicload.ProxyActivity"

android:windowSoftInputMode="stateHidden|adjustResize"

android:configChanges="orientation|keyboardHidden|screenSize" >

<intent-filter>

<action android:name="com.wonxing.dynamicload.proxy.activity.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<activity

android:name="com.wonxing.dynamicload.ProxyFragmentActivity" >

<intent-filter>

<action android:name="com.wonxing.dynamicload.proxy.fragmentactivity.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<service android:name="com.wonxing.dynamicload.ProxyService" />

**3.3.3** 在application节点下添加SDK需要的meta-data(复制+粘贴+修改)

<meta-data android:name="com.wonxing.playsdk.game-activity"

android:value="游戏主Activity Class Name" />

游戏主Activity Class Name需要替换为您的游戏主Activity(即需要录制屏幕的那个Activity)的名字，例如 com.cocos2dx.simplegame.SimpleGameActivity 。

## **3.4 集成步骤3 –修改游戏工程主Activity的java文件**

**3.4.1** 在游戏工程的主Activity的onCreate(...)结尾，添加SDK初始化调用(复制+粘贴+修改)

String strGameKey = "563702c1a15e403b65151896"; // GameKey, assigned by Wonxing.

// TODO: Configure your SDK features here

// init PlaySDK

PlayRec.init(this, strGameKey, null);

GameID为咪咕互动娱乐有限公司分配给您游戏的唯一标识，需要您向咪咕互动娱乐有限公司申请分配然后替换代码中的字符串。

**3.4.2** 其他更多调用(可选项，非必需)

如果您只是大致需要游戏视频录制以及分享等功能，3.4.1初始化后就能达到您的要求了；如果您想游戏和视频录制SDK有更深层次的整合或者更多配置，参见第5章特殊录制场景介绍，以及第4章SDK调用接口的详细介绍。

# 4 SDK调用详细接口



## **4.1 init -SDK的初始化**

* 接口描述

在游戏启动后，需要调用初始化函数PlayRec.init，只有确保初始化成功结束后才能调用录制开始/停止等函数；该函数可以在Application.onCreate中调用，也可以在游戏的启动Activity或者主Activity的onCreate中调用。

需要注意的是这个初始化动作是一个异步过程，即这个函数会马上返回(您会得到PlayRec.INIT\_IN\_PROGRESS返回值)；初始化会在后台线程中继续执行SDK的加载/更新动作，加载完成/失败会通过listener参数回调给您。

* 接口定义

|  |
| --- |
| int PlayRec.init(Activity activity, String gameKey, PlayRecInitResultListenerlistener) |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 含义 | 注释 |
| Activity | Activity | 游戏项目的上下文环境 |  |
| gameKey | String | 游戏ID或者标识符，必须能唯一标识一个游戏 |  |
| Listener | PlayRecInitResultListener | 初始化结果侦听器 | 由于init函数的调用为异步方法，在某些情况下，并不会马上返回初始化的最终结果（返回值为INIT\_IN\_PROGRESS时），此时可能需要一个侦听器来侦听初始化的最终结果；PlayRecInitResultListener有两个方法：  1. public void onUpdateDownloading(int currentSize, int totalSize); 正在更新资源文件的回调  2. public void onInitResult(int result);  初始完成的回调,参数result为初始化最终结果 |

* 返回值说明

|  |  |
| --- | --- |
| PlayRec.INIT\_DONE | 初始化成功完成 |
| PlayRec.INIT\_FAILED | 初始化失败，此种情况下不能再调用其他函数 |
| PlayRec.INIT\_IN\_PROGRESS | 初始化中，此情况下，初始化最终的结果会通过listener参数指定的侦听器返回，此种情况下必须等到listener产生回调时才能调用其他函数 |
| PlayRec.INIT\_DEVICE\_UNSUPPORTED | 初始化失败，设备不支持，此种情况下不能再调用其他函数 |

## **4.2 set/get-RecordingMode -设置/获取录制模式**

* 接口描述

关于录制模式，请参见2.1 关于视频录制模式。

本接口可以修改/获取当前录制模式的接口，一旦设置以后将会被保存，下一次游戏启动以后，将使用最后设置的录制模式；调用setRecordingMode之前，必须先停止录制，并确保SDK处于停止录制状态，否则将会调用失败；如果将模式设置成精彩片段模式，会在下一次重启游戏调用init以后自动启动录制，但不会在调用setRecordingMode之后自动启动录制。

* 接口定义

|  |
| --- |
| boolean PlayRec.setRecordingMode(int recordingMode)  int PlayRec.getRecordingMode() |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 含义 | 注释 |
| recordingMode | Int | 录制模式 | PlayRec.RecordingMode\_Normal: 普通模式；  PlayRec.RecordingMode\_Clip: 精彩片段模式 |

* 返回值说明

## **4.3 startRecording -启动视频录制**

* 接口描述

当状态处于录制停止状态时，使用startRecording函数可以启动录制（两种录制模式均可以通过它启动）

* 接口定义

|  |
| --- |
| boolean PlayRec.startRecording() |

* 参数说明

无

* 返回值说明

|  |  |
| --- | --- |
| True | 启动录制成功 |
| False | 启动录制失败 |

## **4.4 stopRecording -停止视频录制并保存视频文件**

* 接口描述

当状态处于录制状态时，使用stopRecording函数可以停止录制（两种录制模式均可以通过它停止）；由于停止录制以及视频文件的生成都是异步，所以需要有侦听器侦听停止(第1个参数)和视频文件保存(第3个参数)结果/状态。

* 接口定义

|  |
| --- |
| boolean PlayRec.stopRecording(PlayRecRecordingStopListener stopListener, boolean saveVideo,  PlayRecVideoSaveListener saveListener) |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 含义 | 注释 |
| stopListener | PlayRecRecordingStopListener | 录制停止侦听器 | 由于录制停止的过程是个异步过程，设置一个侦听器，通知录制的停止，  PlayRecRecordingStopListener只有一个方法：  public void onRecorinngStopped(); |
| saveVideo | Boolean | 是否需要生成视频文件 | 如果需要正常生成视频文件，请使用true，在某些需要强行停止录制，且不需要保存视频文件的情况下使用false，如果录制模式为精彩片段模式，则该参数无效，默认为false |
| saveListener | PlayRecVideoSaveListener | 录制停止和视频文件生成侦听器 | 由于此接口为异步接口，如果saveVideo被设置为true，则需要设置一个侦听器，如果saveVideo被设置为false，则侦听器不会被回调；侦听器详情见PlayRecVideoSaveListener节的详细说明 |

* 返回值说明

|  |  |
| --- | --- |
| True | 停止录制成功 |
| False | 停止录制失败，如处于停止录制的状态下，调用此函数，就会返回false |

## **4.5 clipRecording -保存精彩片段模式视频片段**

* 接口描述

精彩片段模式时SDK总处于录制/缓存状态，可以随时保存录制/缓存下最后一个时间段的精彩片段，通过这个异步调用可以将这个视频片段保存下来。

* 接口定义

|  |
| --- |
| boolean PlayRec.clipRecording(PlayRecVideoSaveListener listener) |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 含义 | 注释 |
| Listener | PlayRecVideoSaveListener | 录制停止和视频文件生成侦听器 | 由于此接口为异步接口，所以需要设置一个侦听器，详情见PlayRecVideoSaveListener节的详细说明 |

* 返回值说明

|  |  |
| --- | --- |
| True | 保存成功 |
| False | 保存失败 |

## **4.6 PlayRecVideoSaveListener接口说明**

下面是PlayRecVideoSaveListener的方法说明：

|  |
| --- |
| public static interface PlayRecVideoSaveListener {  // 视频文件保存成功时回调  // videoId保存的视频ID，如果videoId为null，则表示视频文件保存失败  public void onPlayRecVideoSaved(String videoId);  } |

## **4.7 getVideoLocalPath -获取视频的本地地址**

* 接口描述

游戏视频被保存后，您通过回调接口会得到视频Id，本接口为根据视频Id找到视频文件的函数。

* 接口定义

|  |
| --- |
| URI PlayRec.getVideoLocalPath(String videoId) |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 含义 | 注释 |
| videoID | String | 视频Id | 来至 PlayRecVideoSaveListener接口回调 |

* 返回值说明

返回本地Video文件的路径

## **4.8 isRecording -是否处于录制状态**

* 接口描述

获取SDK状态接口，得到SDK是否正在录制视频。

* 接口定义

|  |
| --- |
| boolean PlayRec.isRecording() |

* 参数说明

无

* 返回值说明

|  |  |
| --- | --- |
| True | 处于录制状态 |
| False | 未处于录制状态 |

## **4.9 exit -SDK退出接口**

* 接口描述

游戏退出时调用

* 接口定义

|  |
| --- |
| void PlayRec.exit() |

* 参数说明

无

* 返回值说明

无

## **4.10 PlayRec.PlayRecListener说明**

* 接口描述

下面是PlayRecListener的方法说明：

|  |
| --- |
| public static interface PlayRecListener {  // 录制启动成功时回调  public void onRecordingStarted();  // 录制停止成功时回调  public void onRecordingStopped();  // 录制过程中由于发生错误而出现录制意外终止，并传入错误代码  public void onRecordingStoppedWithError(int errCode);  // 录制暂停成功时回调  public void onRecordingPaused();  // 录制恢复成功时回调  public void onRecordingResumed();  // 录制进度回调，duration单位是秒  public void onRecordingProgress(long duration);  } |

如果只关心是否处于录制状态，可以只关注onRecordingStarted(),onRecordingStopped()和onRecordingStoppedWithError()三个函数；

如果需要显示录制过程中的时间，则需要关注函数onRecordingProgress(long duration);

## **4.12 showWin -开启社区界面**

* 接口描述

调用这个接口，进入视频社区。

* 接口定义

|  |
| --- |
| void PlayRec.showWin() |

* 参数说明

无

* 返回值说明

无

## **4.13 videoPreview -开启视频预览界面**

* 接口描述

调用这个接口，进入视频社区中的视频预览界面。

* 接口定义

|  |
| --- |
| void PlayRec.videoPreview (String videoId) |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 含义 | 注释 |
| videoId | String | 待预览的视频文件ID |  |

* 返回值说明

无

## **4.14 hideFloatingController -隐藏SDK的浮窗**

* 接口描述

SDK自带的浮窗默认显示在游戏的主Activity之上，如果您想隐藏它请调用这个接口。

* 接口定义

|  |
| --- |
| void PlayRec.hideFloatingController() |

* 参数说明

无

* 返回值说明

无

## **4.15 showFloatingController -显示SDK的浮窗**

* 接口描述

当您调用SDK浮窗隐藏后，可以调用这个接口显示SDK自带浮窗。

* 接口定义

|  |
| --- |
| void PlayRec.showFloatingController() |

* 参数说明

无

* 返回值说明

无

## **4.16 saveVideoAfterStopRecording -停止录制且不立即保存视频之后再调用此接口保存视频**

* 接口描述

调用此接口之前请先调用4.4接口且saveVideo为false，即停止录制且不保存视频的情况下调用

* 接口定义

|  |
| --- |
| boolean PlayRec.saveVideoAfterStopRecording(PlayRecVideoSaveListener listener) |

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 含义 | 注释 |
| Listener | PlayRecVideoSaveListener | 录制停止和视频文件生成侦听器 | 由于此接口为异步接口，所以需要设置一个侦听器，详情见PlayRecVideoSaveListener节的详细说明 |

* 返回值说明

|  |  |
| --- | --- |
| True | 成功 |
| False | 失败 |

## **4.17 showShareVideoWindowAfterStopRecording -停止录制且不立即保存视频之后再调用此接口保存视频且进入视频分享界面**

* 接口描述

调用此接口之前请先调用4.4接口且saveVideo为false，即停止录制且不保存视频的情况下调用

* 接口定义

|  |
| --- |
| boolean PlayRec.showShareVideoWindowAfterStopRecording() |

* 参数说明

无

* 返回值说明

|  |  |
| --- | --- |
| True | 成功 |
| False | 失败 |

# 5特殊录制场景设置介绍

有的时候需要更加深度的集成，如隐藏SDK自带的浮窗，或者默认打开/关闭一些SDK特定的设置；我们的SDK开发了一个接口PlayRec.setExFeature(java.util.Map mapExFeature)作为特殊录制场景设置的接口，在PlayRec.init(…)前调用这个接口，可以对录制特殊场景进行一些定制。

例如我们通过如下的代码调用，可以用来隐藏SDK自带浮窗，以及去掉精彩瞬间模式的支持。

java.util.Map mapExFeatures = new java.util.HashMap();

mapExFeatures.put("flag-floatctrl", "disabled");

mapExFeatures.put("flag-clipmode", "disabled");

PlayRec.setExFeature(mapExFeatures);

特殊录制场景的开发还在进行中，我们可以根据您的需求添加新的定制功能（其实以上两个定制也是CP提出的需求）。

# 6使用示例

## **6.1 使用实例1（最简单情况）**

最简单的集成如3.4.1介绍，在游戏主Activity或者启动Activity的onCreate(..)中，添加一个最简单的PlayRec.init(..)调用，例如：

String strGameKey = "563702c1a15e403b65151896"; // GameKey, by Wonxing.

// TODO: Configure your SDK features here

// init PlaySDK

PlayRec.init(this, strGameKey, null);

## **6.2 使用实例2（有初始化调试信息打印）**

在最简单实例1(上一节6.1)的基础上，我们可以填充PlayRec.init(..)的初始化回调函数，添加一些调试代码方便我们排查集成失败问题，例如：

String strGameKey = "563702c1a15e403b65151896"; // GameKey, by Wonxing.

// TODO: Configure your SDK features here

// init PlaySDK

final String \_TAG\_RecInit = "PlayRec/Init";

int iRet\_RecInit = PlayRec.init(this, strGameKey, new PlayRec.PlayRecInitResultListener() {

@Override

public void onInitResult(int i) {

if (i == PlayRec.INIT\_DONE) {

Log.i(\_TAG\_RecInit, "PlayRec.init success.");

}

else {

Log.e(\_TAG\_RecInit, "PlayRec.init failed(errCode=0x" + Integer.toString(i, 16) + ")!");

}

}

@Override

public void onUpdateDownloading(int i, int i1) {

// just ignore here

}

});

if (iRet\_RecInit == PlayRec.INIT\_DONE) {

Log.i(\_TAG\_RecInit, "PlayRec.init success(direct).");

}

else if (iRet\_RecInit == PlayRec.INIT\_IN\_PROGRESS) {

Log.i(\_TAG\_RecInit, "PlayRec.init in progress ...");

}

else {

Log.e(\_TAG\_RecInit, "PlayRec.init failed(direct, errCode=0x" + Integer.toString(iRet\_RecInit, 16) + ")!");

}

## **6.3使用实例3（有特殊定制设置/示例1）**

在最简单实例1(上一节6.1)的基础上，我们可以做一些比较深度的集成：

在PlayRec.init(..)调用前，我们调用PlayRec.setExFeature(..)，隐藏SDK自带的浮窗。

游戏中，调用PlayRec.startRecording()开始视频录制。

视频录制过程中，调用PlayRec.stopRecording()停止视频录制并保存视频。

见下面代码片段：

// 视频录制SDK初始化

String strGameKey = "563702c1a15e403b65151896"; // GameKey,by Wonxing.

java.util.Map mapExFeatures = new java.util.HashMap();

mapExFeatures.put("flag-floatctrl", "disabled");

mapExFeatures.put("flag-clipmode", "disabled");

PlayRec.setExFeature(mapExFeatures);

// init PlaySDK

PlayRec.init(this, strGameKey, null);

// 调用PlayRec.startRecording()开始视频录制

PlayRec.startRecording();

// 停止视频录制并保存视频

PlayRec.stopRecording(null, true, null);

## **6.4使用实例4（对AndroidManifest.xml文件的修改）**

在3.3继承步骤2中我们详细说明了对AndroidManifest.xml文件所需的修改，现我们举例对一个AndroidManifest.xml文件所需要的所有更改：黑色部分为原始文件，蓝色部分为我们复制+粘贴的修改，红色部分为复制+粘贴+修改的地方。

见下面修改过的AndroidManifest.xml示例文件：

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.wxdl.sh.wukong"

android:versionCode="21"

android:versionName="2.1" >

<uses-feature android:glEsVersion="0x20000" />

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="14" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.GET\_TASKS" />

<application

android:name="com.sifangyou.wukongpao.SFApplication"

android:icon="@drawable/icon"

android:label="@string/app\_name"

android:hardwareAccelerated="true" >

<meta-data

android:name="UMENG\_APPKEY"

android:value="53f32508fd98c556f70005e4" />

<meta-data

android:name="UMENG\_CHANNEL"

android:value="DEMO\_001" />

<activity

android:name="com.sifangyou.wukongpao.GameActivity"

android:configChanges="orientation|keyboardHidden|screenSize"

android:label="@string/app\_name"

android:screenOrientation="landscape"

android:theme="@android:style/Theme.NoTitleBar.Fullscreen" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name="com.tencent.tauth.AuthActivity"

android:launchMode="singleTask"

android:noHistory="true" >

<intent-filter>

<action android:name="android.intent.action.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

<category android:name="android.intent.category.BROWSABLE" />

<data android:scheme="1104119129" />

</intent-filter>

</activity>

<activity

android:name="com.tencent.connect.common.AssistActivity"

android:configChanges="orientation|keyboardHidden|screenSize"

android:theme="@android:style/Theme.Translucent.NoTitleBar" />

<activity android:name="com.wonxing.dynamicload.ProxyActivity"

android:windowSoftInputMode="stateHidden|adjustResize"

android:configChanges="orientation|keyboardHidden|screenSize" >

<intent-filter>

<action android:name="com.wonxing.dynamicload.proxy.activity.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<activity

android:name="com.wonxing.dynamicload.ProxyFragmentActivity" >

<intent-filter>

<action android:name="com.wonxing.dynamicload.proxy.fragmentactivity.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<service android:name="com.wonxing.dynamicload.ProxyService" />

<meta-data android:name="com.wonxing.playsdk.game-activity"

android:value="com.sifangyou.wukongpao.GameActivity"/>

</application>

<supports-screens

android:anyDensity="true"

android:largeScreens="true"

android:normalScreens="true"

android:smallScreens="true" />

</manifest>