**RK3288\_5.1\_3D配置文档**

**1. Demo安装**

1. 先打开android三个文件中的宏

/device/rockchip/common/BoardConfig.mk文件中，增加如下一行

ENABLE\_STEREO\_DEFORM := true

/frameworks/base/policy/src/com/android/internal/policy/impl/PhoneWindowManager.java

修改如下一行

static final boolean DEBUG\_3D\_FUNCTIONS = false;

将false改为true。

2./frameworks/base/services/core/java/com/android/server/wm/WindowManagerService.java

修改如下一行

static final boolean DEBUG\_3D\_FUNCTIONS = false;

将false改为true

3.编译android工程，烧写固件到设备中。

4.将附件中的libvision-jni.so推到机器的system/lib/目录下。

5.将附件中的libGLES.so推到机器的system/lib/egl/目录下。

6.安装Demo apk。

rm rk3066\_lai -rf

|  |  |
| --- | --- |
| **API** | **说明** |
| **void enableStereo(int mode)** | 设置左右分屏开关，mode为1和2，代表分屏显示，1适用物理横屏机器，比如box；2适用物理竖屏机器，比如手机、平板；mode为0，代表关闭分屏显示。 |
| **void setStereoHeight(float height)** | height为一个0.0~ 1.0之间的浮点数，控制左右分屏显示时候画面的高度占屏幕高度的百分比，height为0.5则表示屏幕高度的0.5倍，以此类推。  小于0.0，效果同0.0；大于1.0，效果同1.0。 |
| **void enableDispersion(boolean bool)** | 设置色散开关，bool为true，色散打开，6个畸变参数全部生效；bool为false，色散关闭，只有前两个畸变参数生效。 |
| **void setDeformIntensity(float rk1,float rk2,float gk1,float gk2,float bk1,float bk2)** | 设置畸变色散的强度，以适配不同参数的物理透镜。k1为粗调，k2为细调。参数越大，畸变程度越强。参数的范围建议在0-2之间。 |
| **void addNameToList(String name)** | 增加应用包名到“增强3d模式名单” |
| **void delNameFromList(String name)** | 从“增强3d模式名单”中删除包名 |
| **void setIPDinEnhanced3dMode(float dist)** | 瞳距调节：仅限于增强3d模式中的瞳距调节  dist范围在(-1.0,1.0)。0表示初始居中状态，大于0表示左右画面向外分离，小于0表示向内靠拢。  **以下两个接口的参数标准相同** |
| **void setIPDbyScale(float dist)** | 瞳距调节：左右画面的“往外分离”或“向内聚拢”采用的是画面单边缩放方式，不会损失画面内容。 |
| **void setIPDbyOffset(float dist)** | 瞳距调节：左右画面的“往外分离”或“向内聚拢”采用的是画面平移方式，超出边界的部分丢失。 |
| **void enableAutodetect(boolean i)** | 3d视频自动识别。i为true开启此功能，i为false关闭此功能。 |

三个瞳距调节接口实现的效果各不相同，阅读者可以根据自己的实际需求进行一定地选择调用。三种接口的效果的区别，可以回顾2.4小节的插图。

在所有的接口中，addNameToList和delNameFromList接口在被调用之后，并不是立即生效。举例：调用addNameToList接口增加了包名为com.xxx.yyy的应用到了名单中，那么用户必须重启这个包名为com.xxx.yyy的应用，才能让应用以增强3d模式显示。

其余接口都是实时控制，调用之后即时生效。

**JAVA接口测试方法**

|  |  |
| --- | --- |
| **API** | **测试方法** |
| **void enableStereo(boolean bool)** | **法一**：同上，调用接口之后肉眼判别是否分屏显示  **法二**：getprop | grep 3d 参看属性值sys.game.3d和sys.hwc.force3d.primary是否改变。开启属性值应当为1，反之为0； |
| **void setStereoHeight(float height)** | **法一**：调用此接口之后，开启左右分屏显示，可肉眼判别尺寸设置是否生效；  **法二**：调用此接口后，进入adb shell, getprop | grep sys.3d.height 查看属性值是否与接口设置相同。 |
| **void enableDispersion(boolean bool)** | **法一**：同上，调用之后肉眼判别是否开启色散；  **法二**：getprop | grep dispersion 查看属性是否与接口设置相同。 |
| **void setDeformIntensity(float rk1,float rk2,float gk1,float gk2,float bk1,float bk2)** | **法一**：同上，调用接口之后，开启畸变，肉眼判别畸变程度是否改变；  **法二**：getprop | grep deform 查看6个属性值是否于接口设置相同； |
| **void addNameToList(String name)**  **void delNameFromList(String name)** | **法一**：在adb shell中，cat /data/data/com.rockchip.vison/app\_info,添加名单和删除名单都会实时地反映在这个文件节点上。  调用此接口之后查看这个文件，可知是否生效。 |
| **3个瞳距调节接口** | 调用接口实时调整参数，肉眼即可观察到变化 |

**开发apk**

* + **import开发包**

在MainActivity.java文件头部，加入如下语句：

import com.rockchip.extensions.StereoManager;

final StereoManager obj = new StereoManager();

接下来就可以使用obj对象调用全部的3d显示接口。

UI界面布局、控件使用、应用包名获取等等，开发者可以自行优化设计或借鉴Demo源码。

注意：

如果开发者是在Demo的基础上进行优化开发，demo中的部分函数被封装在libvison-jni.so库中，需要将libvision-jni.so库推到机器的system/lib/目录下才可以调用。

**编译apk**

因为开发包中接口涉及到了和Android系统服务的通信，**因此，开发者无论是自行使用开发包开发还是修改demo进行二次开发，都需要将apk工程放到android 5.1.x工程目录下的packages/apps/目录下进行编译，不可在eclipse下进行编译**。需要注意的有以下几项：

**配置JAVA**

可将开发包中的vision.tar.gz 拷贝到工程的packages/apps/目录下

tar -xzvf vision.tar.gz 解压文件，继续往下：

在packages/apps/MyVisonSetting目录下执行mm -j4 -B之前，需要先配置JAVA，在当前目录下执行如下三条命令后即可编译apk：

**export JAVA\_HOME=/usr/lib/jvm/java-7-sun**

**export PATH=$JAVA\_HOME/bin:$PATH**

**export CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/lib/tools.jar**

**增加系统权限**

修改AndroidManifest.xml文件，来为apk增加系统权限。

具体参考附件demo源码中的AndroidManifest.xml文件中的<uses-permission/>标签。

**修改Android.mk文件**

本文档中所列出的Java接口已经全部被封装到一个jar包当中，而此jar包已经被整合到SDK里，所以编译apk的时候需要修改Android.mk文件，将jar包包含进去。

在Android.mk文件中添加如下两行，将library.jar文件编译到apk当中

LOCAL\_PREBUILT\_STATIC\_JAVA\_LIBRARIES :=library:../../../frameworks/base/stereo/rockchipextensions\_stereo.jar 这里要注意路径

LOCAL\_STATIC\_JAVA\_LIBRARIES := library

Android.mk文件具体写法参考示例demo apk源码。

**监听设备的音量键**

在SDK的如下路径文件文件中

framework/base/policy/src/com/android/internal/policy/impl/PhoneWindowManager.java

的interceptKeyBeforeQueueing函数中，添加相应的代码段。即可实现对设备音量键的监听。

如下是代码示例，示例监听的是快按两次音量加和音量键分别实现的监听，开发者模仿该代码段自行添加想要实现的功能：比如监听不同的键位，采用不同的按键方式监听等等。（这块代码已经好）

//连续两次快按音量加，快按间隔小于0.4秒

if(event.getKeyCode()==KeyEvent.KEYCODE\_VOLUME\_UP && event.isDown()){

if((System.currentTimeMillis()-lastTime) < 400){

//\*\*\*\*\*add what you want to do\*\*\*\* }

lastTime=System.currentTimeMillis();

}

//连续两次快按音量减，快按间隔小于0.4秒

if(event.getKeyCode()==KeyEvent.KEYCODE\_VOLUME\_DOWN && event.isDown()){

//\*\*\*\*add what you want to do\*\*\*\*

}

lastTime=System.currentTimeMillis();}

**常见问题**

**apk无法编译**

* **java配置 (source build.sh)**

export JAVA\_HOME=/usr/lib/jvm/java-7-sun

export PATH=$JAVA\_HOME/bin:$PATH

export CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/lib/tools.jar

编译之前没有执行这三条命令会导致编译不过。

* **检查源码路径**

源码的路径在packages/apps/下，目录名为MyVisionSetting，使用者将开发包拷贝到packages/apps/下直接解压即可，不可自行新建一级目录，在新的目录中解压源码。

* **检查jar包**

apk依赖于一个3D显示接口扩展包，已经整合到SDK中。

所在路径为：frameworks\base\stereo\rockchipextensions\_stereo.jar

如果开发者没有及时更新工程，或只是局部更新工程，则会导致工程该路径下没有jar包，或者使用的是旧版的jar包，导致编译出错或者apk功能不对。

**解决方法**：直接使用附件中的jar包，将jar包复制到工程源码的上述路径中，如果没有stereo目录，则自行创建一个stereo目录。

**5.2 增强3D模式**

* **apk中进入标签“增强3D模式名单”报错退出**

安装apk之前，需要把开发包中的libvision-jni.so推到机器的system/lib/目录下， 如果遗漏了此操作步骤，请重新推送libvision-jni.so库到相应的目录下。

* **增强3D模式瞳距调节无效？**

增强3D模式的瞳距调节，和普通的瞳距调节效果不一样。增强3D模式的瞳距调节，指的是对游戏中的景深场景进行视角的左右移动。实际看到的效果并不是显示图像的左右位移变化，而是显示图像中的内容变化。请仔细核对调节3D模式瞳距之后，游戏的场景内容是否有平移，再判定接口是否生效。

使用者可以自行选择瞳距调节的方式，3D瞳距调节可以和另外两种瞳距调节方式之一搭配使用。

* **增强3D模式名单中勾选之后没有生效？**

**情况1：**

adb shell进入机器

cd system/lib/egl/查看当前目录下是否有一个libGLES.so的库。名称是否正确。

如果没有so库，将开发包附件中的同名so库推到system/lib/egl/目录下。

**情况2：**

增强3D模式是否开启，需要按照2.5节的3D显示逻辑来判断。

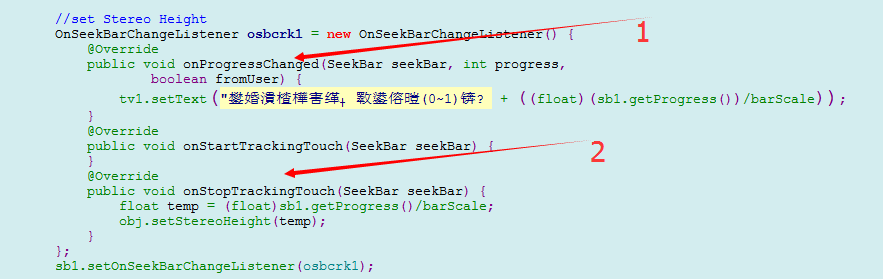
同时，增强3D模式开启的现象是：左右图像产生了“视差”。如果使用者仔细观察的话，会发现增强3D开启和关闭，图像是不一样的。使用者如果没有仔细观察，可能会误认为增强3D没有开启。

**5.3 apk控件无效**

**情况一：**

Android SDK API一直在更新，demo主要用到的控件有textview、seekbar、checkbox等等。控件的使用方式在不同的API版本中可能会有区别，导致控件失效。

举例：



Demo中使用概率极高的seekbar控件，部分客户可以使用，另一部分客户则不行。原因在于，大家使用的Android SDK API不是同一版本，有些客户的API版本的seekbar使用方式不支持图中的2函数，而Demo中的apk使用的方式是接口的调用放在函数2中。

**解决方法：**将函数2中的代码剪切到函数1当中。

**情况二：**

使用者安装的demo必须是将源码拷贝到自己的平台上编译出来得到的apk，因为我们封装的java接口需要和底层进行通信，需要在源码平台上编译，以此获取系统权限。如果使用者用来安装的apk是早期开发包中直接提供的apk文件，可能就会出现控件失效的问题。

**解决方法**：将demo源码拷贝到android源码工程中，自行编译安装。

**5.4 开启分屏显示时黑屏**

部分使用者出现了开启分屏模式1，出现黑屏的情况。

**原因：**分屏模式1的代码是继分屏模式2之后才添加的，如果使用者没有把源码更新到最新版本，会出现开启分屏模式1黑屏的情况。

**解决方法：**更新源码到最新的版本。