1. 概述

PLDroidShortVideo 是七牛推出的一款适用于 Android 平台的短视频 SDK，提供了包括美颜、滤镜、水印、断点录制、回删、本地存储，云端存储在内的多种功能，还可以支持高度定制以及二次开发。

2. 功能列表

- 视频采集
- 音频采集
- 视频 H.264 硬编码
- 音频 AAC 硬编码
- 实时美颜
- 实时滤镜
- 支持第三方美颜
- 支持第三方滤镜
- 自定义时长
- 自定义码率
- 自定义分辨率
- 支持 1:1 录制
- 断点拍摄
- 回删视频
- 视频水印
- 视频存为 .mp4 格式
- 支持 armv7, arm64, i386, x86_64 体系架构

3. 阅读对象

本文档为技术文档，需要阅读者：

- 具有基本的 Android 开发能力
- 准备接入七牛云存储

4 开发准备

4.1 设备以及系统要求

- 系统要求：Android 4.3 (API 18) 及其以上

5. 快速开始

5.1 开发环境

- Android Studio 开发工具，官方下载地址
- Android 官方开发 SDK，官方下载地址。

5.2 下载和导入 SDK

SDK 主要包含 demo 代码、sdk jar 包，以及 sdk 依赖的动态库文件。

其中，release 目录下是需要拷贝到您的 Android 工程的所有文件，请参考 demo 项目的格式。

5.3 修改 build.gradle

双击打开您的工程目录下的 `build.gradle`，确保已经添加了如下依赖，如下所示：

```
dependencies {
    compile files('libs/pldroid-shortvideo-x.y.z.jar')
}
```

5.4 添加相关权限

在 app/src/main 目录中的 AndroidManifest.xml 中增加如下 `uses-permission` 声明：

```
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.FLASHLIGHT" />
```

5.5 开发步骤

5.5.1 视频编辑

5.5.1.1 创建录制对象

```
  PLShortVideoRecorder mShortVideoRecorder = new
PLShortVideoRecorder();
  mShortVideoRecorder.setRecordStateListener(this);
```

5.5.1.2 配置采集参数

`PLShortVideo` 提供了丰富的自定义录制选项，可以通过创建以下对象进行配置：

```
  // 摄像头采集选项
  PLCameraSetting cameraSetting = new PLCameraSetting();

cameraSetting.setCameraId(PLCameraSetting.CAMERA_FACING_ID.CAMERA_FACI
NG_FRONT);

cameraSetting.setCameraPreviewSizeRatio(getPreviewSizeRatio(previewSiz
eRatio));

cameraSetting.setCameraPreviewSizeLevel(getPreviewSizeLevel(previewSiz
eLevel));

  // 麦克风采集选项
  PLMicrophoneSetting microphoneSetting = new PLMicrophoneSetting();

  // 视频编码选项
  PLVideoEncodeSetting videoEncodeSetting = new
PLVideoEncodeSetting();

videoEncodeSetting.setEncodingSizeLevel(getEncodingSizeLevel(encodingS
izeLevel));

  // 音频编码选项
  PLAudioEncodeSetting audioEncodeSetting = new
PLAudioEncodeSetting();

  // 美颜选项
  PLFaceBeautySetting faceBeautySetting = new
PLFaceBeautySetting(1.0f, 0.5f, 0.5f);

  // 录制选项
  PLRecordSetting recordSetting = new PLRecordSetting();

recordSetting.setMaxRecordDuration(RecordSettings.DEFAULT_MAX_RECORD_D
URATION);

recordSetting.setVideoCacheDir(Environment.getExternalStorageDirectory
());
  recordSetting.setVideoFilepath(RECORD_FILE_PATH);
```

5.5.1.2 配置预览窗口

预览窗口为 GLSurfaceView 或自定义的继承 GLSurfaceView 视图，需要在 XML 布局中进行定义。

5.5.1.3 录制准备

```
    mShortVideoRecorder.prepare(preview, cameraSetting,
microphoneSetting,
            videoEncodeSetting, audioEncodeSetting, faceBeautySetting,
recordSetting);
```

5.5.1.4 录制与合并

SDK 支持分段录制，每一段录制开始，调用 `PLShortVideoRecorder#beginSection`；每一段结束时，调用 `PLShortVideoRecorder#endSection`；当所有片段都录制完成后，调用 `PLShortVideoRecorder#concatSections` 将各个片段进行合并，合并完成后，会收到 `onConcatSuccess` 的回调。

5.5.1.5 生命周期

请在 `Activity` 的各个生命周期的回调中，分别调用 `PLShortVideoRecorder` 对应的方法。

5.5.2 视频编辑

5.5.2.1 创建编辑对象

首先创建 `PLShortVideoEditor` 对象，传入 `GLSurfaceView` 对象用于画面预览，以及编辑的配置 `PLVideoEditSetting`，注意，`PLVideoEditSetting` 必须设置视频源的地址，如：

```
    PLVideoEditSetting setting = new PLVideoEditSetting();
    setting.setSourceFilepath(videopath);
    mShortVideoEditor = new PLShortVideoEditor(mPreviewView, setting);
```

5.5.2.2 配置编辑参数

`PLShortVideo` 提供了丰富的自定义编辑选项，可以通过创建以下对象进行配置：

```
    PLVideoEditSetting setting = new PLVideoEditSetting();
    setting.setSourceFilepath(videopath);
    setting.setDestFilepath(outputpath);
    setting.setKeepOriginFile(true);
    setting.setPlaybackLooping(true)
```

5.5.2.3 内置滤镜

`PLShortVideoEditor` 内置了多达 30 多种滤镜，滤镜列表可以通过 `PLShortVideoEditor#getBuiltinFilterList` 获取，可以使用一个 `RecyclerView` 呈现：

```
  RecyclerView mFilterList = (RecyclerView)
findViewById(R.id.recycler_view);
  mFiltersList.setAdapter(new
FilterListAdapter(mEditor.getBuiltinFilterList()));
```

当用户选中对应的内建滤镜后，可以使用 `PLShortVideoEditor#setBuildinFilter` 进行设置。

5.5.2.4 水印功能

`PLShortVideoEditor` 支持水印的添加，可以通过 `PLWatermarkSetting` 对象设置水印：

```
  mWatermarkSetting = new PLWatermarkSetting();
  mWatermarkSetting.setResourceId(R.drawable.qiniu_logo);
  mWatermarkSetting.setPosition(0.01f, 0.75f);
  mWatermarkSetting.setAlpha(128);
  mShortVideoEditor.setWatermark(mWatermarkSetting);
```

5.5.2.5 视频预览

`PLShortVideoEditor` 支持编辑的视频画面预览，可以使用 `PLShortVideoEditor#startPlayback` 与 `PLShortVideoEditor#stopPlayback` 来控制视频的播放与停止。

5.5.2.6 视频保存

编辑后的视频效果，可以保存到本地，调用 `PLShortVideoEditor#save` 即可保存，保存结果通过 `setVideoSaveListener` 设置监听来通知。

6 SDK 接口设计

6.1 核心类

短视频核心类有两个，分别是 `PLShortVideoRecorder` 与 `PLShortVideoEditor` ，分别完成视频的录制与编辑工作。

6.1.1 PLShortVideoRecorder

`PLShortVideoRecorder` 包含了录制短视频所需要的接口，方法列表如下：

```
/**
 * 构造 PLShortVideoRecorder 对象
 */
public PLShortVideoRecorder();

/**
 * 使 PLShortVideoRecorder 暂停工作，通常在 Activity#onPause 中调用
 */
public pause();

/**
 * 使 PLShortVideoRecorder 恢复工作，通常在 Activity#onResume 中调用
 */
public resume();

/**
 * 销毁 PLShortVideoRecorder 对象，通常在 Activity#onDestroy 中调用
 */
public destory();

/**
 * Prepare the recorder
 *
 * @param glSurfaceView      预览的 GLSurfaceView
 * @param cameraSetting      PLCameraSetting 对象
 * @param microphoneSetting  PLMicrophoneSetting 对象
 * @param videoEncodeSetting PLVideoEncodeSetting 对象
 * @param audioEncodeSetting PLAudioEncodeSetting 对象
 * @param faceBeautySetting  PLFaceBeautySetting 对象
 * @param recordSetting      PLRecordSetting 对象
 */
public void prepare(GLSurfaceView glSurfaceView,
                    PLCameraSetting cameraSetting,
                    PLMicrophoneSetting microphoneSetting,
                    PLVideoEncodeSetting videoEncodeSetting,
                    PLAudioEncodeSetting audioEncodeSetting,
                    PLFaceBeautySetting faceBeautySetting,
                    PLRecordSetting recordSetting);

/**
 * 开始录制片段，需要收到 `onReady` 回调之后执行
 */
public boolean beginSection();

/**
 * 停止录制当前片段
```

```
 */
public boolean endSection();

/**
 * 删除上一个录制的片段
 */
public boolean deleteLastSection();

/**
 * 合并录制的片段，SDK 将会在缓存目录中临时创建对应文件
 * @param    用于接收合并回调的 listener
 */
public void concatSections(PLConcatStateListener listener);

/**
 * 切换前后摄像头
 */
public void switchCamera();

/**
 * 设置闪光灯开关
 */
public void setFlashEnabled(boolean enabled);

/**
 * 更新美颜设置
 * @param 美颜设置对象
 */
public final void updateFaceBeautySetting(PLFaceBeautySetting setting);

/**
 * 注册相机预览监听器
 * @param 监听器对象
 */
public final void setCameraPreviewListener(PLCameraPreviewListener
listener);

/**
 * 注册相机参数选定监听器
 * @param 监听器对象
 */
public final void
setCameraParamSelectListener(PLCameraParamSelectListener listener);

/**
 * 注册自定义美颜、滤镜效果的监听器
 * @param 监听器对象
```

```
 */
public final void setVideoFilterListener(PLVideoFilterListener
listener);

/**
 * 注册音频帧监听器
 * @param 监听器对象
 */
public final void setAudioFrameListener(PLAudioFrameListener listener);

/**
 * 注册录制状态监听器
 * @param 监听器对象
 */
public final void setRecordStateListener(PLRecordStateListener
listener);
```

6.1.2 PLShortVideoEditor

`PLShortVideoEditor` 包含了播放短视频所需要的接口，方法列表如下：

```
/**
 * 构造 PLShortVideoEditor 对象
 * @GLSurfaceView 用于预览视频的 View
 * @setting 编辑配置
 */
public PLShortVideoEditor(GLSurfaceView view, PLVideoEditSetting
setting);

/**
 * 播放视频
 */
public void startPlayback();

/**
 * 播放视频，并添加自定义滤镜效果
 */
public void startPlayback(PLVideoFilterListener listener);

/**
 * 停止播放
 */
public void stopPlayback();

/**
 * 获取 SDK 自带的滤镜列表
```

```
     * @return PLBuiltinFilter 数组
     */
    public PLBuiltinFilter[] getBuiltinFilterList();

    /**
     * 应用 SDK 自带的滤镜
     * @param 滤镜名，从 getBuiltinFilter 获取
     */
    public void setBuiltinFilter(String filterName);

    /**
     * 注册保存视频的监听器
     * @param 监听器对象
     */
    public void setVideoSaveListener(PLVideoSaveListener listener);

    /**
     * 应用水印设置
     * @param 水印设置变量
     */
    public void setWatermark(PLWatermarkSetting setting);

    /**
     * 保存视频
     */
    public void save();

    /**
     * 保存视频，并添加自定义滤镜效果
     */
    public void save(PLVideoFilterListener listener);
```

6.2 参数配置

6.2.1 PLCameraSetting

`PLCameraSetting` 类用于配制摄像头采集的相关参数，包含以下方法：

```
/**
 * 设置采集摄像头
 * @param 默认设置为手机的后置摄像头
 */
public PLCameraSetting setCameraId(PLCameraSetting.CAMERA_FACING_ID
cameraId);

/**
 * 设置采集画面的分辨率
 * @param 可以选择从 120P 到 1200P 之间的多种分辨率。默认设置为 480P
 */
public PLCameraSetting
setCameraPreviewSizeLevel(PLCameraSetting.CAMERA_PREVIEW_SIZE_LEVEL
level);

/**
 * 设置采集画面的长宽比
 * @param 可以选择 4:3 或 16:9。默认设置为 16:9
 */
public PLCameraSetting
setCameraPreviewSizeRatio(PLCameraSetting.CAMERA_PREVIEW_SIZE_RATIO
ratio);
```

6.2.2 PLMicrophoneSetting

`PLMicrophoneSetting` 类用于配置麦克风采集的相关参数，包含以下方法：

```
/**
 * 设置音频采集源
 * @param Source ID, 默认为系统的 MediaRecorder.AudioSource.MIC，可时情况进行修改
 */
public PLMicrophoneSetting setAudioSource(MediaRecorder.AudioSource source);

/**
 * 设置音频格式
 * @param Format type, 默认为系统的 AudioFormat.ENCODING_PCM_16BIT，其他格式可能不兼容部分设备，请视情况使用
 */
public PLMicrophoneSetting setAudioFormat(MediaRecorder.AudioSource format);

/**
 * 设置音频声道
 * @param Channel type, 默认为系统的 AudioFormat.ENCODING_CHANNEL_IN_MONO，其他格式可能不兼容部分设备，请视情况使用
 */
public PLMicrophoneSetting setChannelConfig(MediaRecorder.AudioSource channel);

/**
 * 设置音频采样率
 * @param 采样率, 默认为 44100，其他设置可能不兼容部分设备，请视情况使用
 */
public PLMicrophoneSetting setSampleRate(int rate);

/**
 * 开启音频 PTS 优化，有助于降低音画不同步概率，略微消耗计算资源
 * @param 是否开启。默认开启
 */
public PLMicrophoneSetting setAudioPtsOptimizeEnabled(boolean enable);

/**
 * 设置是否开启蓝牙 SCO，如果使用蓝牙耳机采集音频
 * @param 是否开启。默认关闭
 */
public PLMicrophoneSetting setBluetoothSCOEnabled(boolean enable);
```

6.2.3 PLVideoEncodeSetting

`PLVideoEncodeSetting` 类用于配制视频编码的相关参数，包含以下方法：

```
/**
 * 设置编码码率
 * @param 码率，单位为 bps，默认为 1000 * 1000
 */
public PLVideoEncodeSetting setEncodingBitrate(int bitrate);

/**
 * 设置码率模式
 * @param 可选择 QUALITY_PRIORITY 或 BITRATE_PRIORITY
 * @param 在一些情况下 SDK 会分别保证画质而临时提高码率，或保证码率稳定临时降低画质
 * @param 默认设置为 `QUALITY_PRIORITY`
 */
public PLVideoEncodeSetting setEncodingBitrateMode(PLVideoEncodeSetting.BitrateMode mode);

/**
 * 设置编码 fps
 * @param fps 值，默认设置为 30
 */
public PLVideoEncodeSetting setEncodingFps(int fps);

/**
 * 设置编码视频的分辨率
 * @param 可以选择从 120P 到 1200P 之间的多种分辨率，默认设置为 480 * 480
 */
public PLVideoEncodeSetting setEncodingSizeLevel(PLVideoEncodeSetting.VIDEO_ENCODING_SIZE_LEVEL level);

/**
 * 如果 SDK 内置的分辨率列表不能满足需求，可以通过此方法自定义编码视频的分辨率
 * @param width 宽度
 * @param height 高度
 */
public PLVideoEncodeSetting setPrefferedEncodingSize(int width, int height);

/**
 * 设置编码视频关键帧的间隔
 * @param 单位为帧数，默认设置为 30 帧
 */
public PLVideoEncodeSetting setIFrameInterval(int interval);
```

6.2.4 PLAudioEncodeSetting

`PLAudioEncodeSetting` 类用于配制音频编码的相关参数，包含以下方法：

```
/**
 * 设置编码码率
 * @param 码率，单位为 bps，默认为 16 * 1000
 */
public PLAudioEncodeSetting setBitrate(int bitrate);

/**
 * 设置音频声道数量
 * @param 声道数量，默认为 1，建议与采集时的声道数量保持一致
 */
public PLAudioEncodeSetting setChannels(int channels);

/**
 * 设置音频采样率
 * @param 采样率，默认为 44100，建议与采集时的设置保持一致
 */
public PLAudioEncodeSetting setSampleRate(int rate);
```

6.2.5 PLFaceBeautySetting

`PLFaceBeautySetting` 类用于配制美颜相关参数，包含以下方法：

```
 /**
  * 构造 PLFaceBeautySetting 对象
  * @param beauty 美颜程度，0-1 递增
  * @param whiten 美白程度，0-1 递增
  * @param redden 红润程度，0-1 递增
  */
public PLAudioEncodeSetting(float beauty, float whiten, float redden);

/**
 * 设置是否开启美颜
 * @param 是否开启。默认开启
 */
public void setEnable(boolean enable);

/**
 * 设置美颜程度
 * @param 0-1 递增
 */
public void setBeautyLevel(float level);

/**
 * 设置美白程度
 * @param 0-1 递增
 */
public void setWhiten(float level);

/**
 * 设置红润程度
 * @param 0-1 递增
 */
public void setRedden(float level);
```

6.2.6 PLRecordSetting

`PLRecordSetting` 类用于配制录制相关参数，包含以下方法：

```
/**
 * 设置最大录制长度
 * @param 最大录制长度，单位 ms，默认 10*1000
 */
public PLRecordSetting setMaxRecordDuration(int duration);

/**
 * 设置视频缓存目录
 * @param 缓存目录
 */
public PLRecordSetting setVideoCacheDir(File dir);
public PLRecordSetting setVideoCacheDir(String dir);

/**
 * 设置录制的视频文件全路径，默认从缓冲目录随机生成
 * @param 视频文件全路径
 */
public PLRecordSetting setVideoFilepath(String filepath);

/**
 * 设置录制时 Camera preview 在 GLSurfaceView 和编码中的显示模式
 * @param 显示模式，可为 FIT 或 FULL
 */
public PLRecordSetting setDisplayMode(PLDisplayMode displayMode);
```

6.2.7 PLVideoEditSetting

`PLVideoEditSetting` 对象用语配置视频编辑的相关参数，方法如下：

```
/**
 * 设置是否循环播放
 *
 * @param 是否循环播放
 */
public PLVideoEditSetting setPlaybackLooping(boolean looping);

/**
 * 设置编辑保存后是否保留原始文件
 *
 * @param 是否保留原始文件
 */
public PLVideoEditSetting setKeepOriginFile(boolean keepOriginFile);

/**
 * 设置需要编辑的视频文件地址
 *
 * @param 原视频文件地址
 */
public PLVideoEditSetting setSourceFilepath(String filepath);

/**
 * 设置 `save` 保存视频地址
 *
 * @param 保存视频地址
 */
public PLVideoEditSetting setDestFilepath(String filepath);

/**
 * 设置播放视频的显示模式
 *
 * @param 显示模式，可为 FIT 或 FULL
 */
public PLVideoEditSetting setDisplayMode(PLDisplayMode displayMode);
```

6.2.8 PLWatermarkSetting

`PLWatermarkSetting` 对象用于配制视频水印的相关参数，方法如下：

```
/**
 * 设置水印素材 ID
 * @param resource id
 */
public void setResourceId(int resourceId);

/**
 * 设置水印位置
 * @param x 水印左上角在视频 x 轴上的位置，0-1
 * @param y 水印左上角在视频 y 轴上的位置，0-1
 */
public void setPosition(float x, float y);

/**
 * 设置水印透明度
 * @param alpha, 0-255
 */
public void setAlpha(int alpha);
```

6.3 监听器

6.3.1 PLCameraPreviewListener

该接口用于回调相机预览相关的事件，方法如下：

```
/**
 * 当预览画面有视频帧待渲染时触发
 * @param data 视频帧数据
 * @param width 视频帧宽度
 * @param height 视频帧高度
 * @param rotation 视频角度，顺时针 0/90/180/270 度
 * @param fmt 视频格式，在 PLFourCC 类中定义
 * @param timestamp 该帧的时间戳，单位为 ns
 */
boolean onPreviewFrame(byte[] data, int width, int height, int
rotation, int fmt, long timestampNs);
```

6.3.2 PLAudioFrameListener

该接口用于回调音频录制相关的事件，方法如下：

```
/**
 * 录制失败时触发
 * @param 错误码，在 PLErrorCode 中定义
 */
void onAudioRecordFailed(int errorCode);

/**
 * 当有待处理的音频帧时触发
 * @param data PCM 数据
 * @param timestampUs 音频帧时间戳，单位为 ns
 */
void onAudioFrameAvailable(byte[] data, long timestampNs);
```

6.3.3 PLCameraParamSelectListener

该接口用于回调相机参数相关的事件，方法如下：

```
/**
 * 当预览尺寸确定时触发
 * @param 此设备支持的尺寸列表
 * @return 选择的尺寸
 */
Camera.Size onPreviewSizeSelected(List<Camera.Size> list);

/**
 * 当采集视频的 fps 确定时触发
 * @param 此设备支持的 fps 列表
 * @return 选择的 fps
 */
int[] onPreviewFpsSelected(List<int[]> list);
```

6.3.4 PLRecordStateListener

该接口用于回调录制相关的事件，接口如下：

```
/**
 * 当准备完毕可以进行录制时触发
 */
void onReady();

/**
 * 当录制的片段过短时触发
 */
void onDurationTooShort();

/**
 * 录制开始
 */
void onRecordStarted();

/**
 * 录制结束
 */
void onRecordStopped();

/**
 * 当有新片段录制完成时触发
 * @param incDuration   增加的时长
 * @param totalDuration 总时长
 * @param sectionCount  当前的片段总数
 */
void onSectionIncreased(long incDuration, long totalDuration, int
sectionCount);

/**
 * 当有片段被删除时触发
 * @param decDuration   减少的时长
 * @param totalDuration 总时长
 * @param sectionCount  当前的片段总数
 */
void onSectionDecreased(long decDuration, long totalDuration, int
sectionCount);

/**
 * 当录制全部完成时触发
 */
void onRecordCompleted();
```

6.3.5 PLConcatStateListener

该接口用于回调视频拼接相关的事件，接口如下：

```
/**
 * 拼接失败时触发
 * @param 错误码，在 PLErrorCode 中定义
 */
void onConcatFailed(int errorCode);

/**
 * 当拼接进度更新时触发
 * @param num 进度值，0-100
 * @param 总片段数
 */
void onConcatProgressUpdate(int num, int sectionCount);

/**
 * 拼接成功时触发
 * @param 拼接后的文件路径
 */
void onConcatSuccess(String file);
```

6.3.6 PLVideoFilterListener

该接口用于外部对接第三方美颜或者滤镜，接口如下：

```
/**
 * 当有新的 surface 创建时触发
 **/
void onSurfaceCreated();

/**
 * 当 surface 发生改变时触发
 * @param width 宽度
 * @param height 高度
 *
 **/
void onSurfaceChanged(int width, int height);

/**
 * 当 Surface 销毁时触发
 **/
void onSurfaceDestroy();

/**
 * 绘制帧时触发
 * @param texId 待渲染的 SurfaceTexture 对象的 texture ID
 * @param texWidth 绘制的 surface 宽度
 * @param texHeight 绘制的 surface 高度
 * @return 新创建的制定给 SurfaceTexuture 对象的 texture ID
 **/
int onDrawFrame(int texId, int texWidth, int texHeight, long
timeStampNs);
```

6.3.7 PLVideoSaveListener

该接口用于回调视频保存相关的事件，接口如下：

```
/**
 * 保存成功时触发
 * @param 编辑后的视频文件
 */
void onSaveVideoSuccess(File editedFile);

/**
 * 保存失败时触发
 * @param 错误码，在 PLErrorCode 中定义
 */
void onSaveVideoFailed(int errorCode);
```

7. 自定义对象

7.1 PLFourCC

用于定义视频格式，包含如下内容：

- `FOURCC_I420`: I420
- `FOURCC_NV21`: NV21
- `FOURCC_NV12`: NV12
- `FOURCC_ABGR`: ABGR
- `FOURCC_UNKNOWN`: 未知格式

7.2 PLErrorCode

自定义错误码，包含如下内容：

- `ERROR_IO_EXCEPTION`: IO 错误
- `ERROR_WRONG_STATUS`: 状态错误
- `ERROR_EMPTY_SECTIONS`: 无片段
- `ERROR_MUXER_STOP_FAILED`: MUXER 错误

7.3 PLBuiltinFilter

SDK 自带的滤镜对象，包含以下方法：

- `public String getName()`: 获取滤镜名，可用于参数传入 `PLVideoEditor#setBuiltinFilter`
- `public String getAssetFilePath`: 获取滤镜的资源文件路径

8 历史记录

- 1.0.0
    - 发布了 pldroid-player-1.0.0.jar
    - 发布了 libpldroid_beauty.so
    - 完成初版功能