CameraKit for Android

功能

①：在预览模式下无缝地捕捉图片和视屏。

②：自动处理系统权限。

③：自动缩放预览界面。

- 创建一个任意大小的CameraView

- 自动地修剪图像输出以适配你的CameraView边框宽高（可以保证不变形）

④：多样的捕捉方法

- METHOD_STANDARD（标准）：正常地使用cameraAPI来捕捉图像

- METHOD_STILL（静态）：捕捉CameraView预览画面的冻结帧，针对更慢的相机设备，类似于阅后即焚和图片分享。

- METHOD_SPEED（快速）：自动地捕捉图像，基于测量速度决定。

⑤：内置持续聚焦。

⑥：内置手势缩放。

安装和使用

```
implementation 'com.wonderkiln:camerakit:0.13.1'//app的build.gradle
```

```
<com.wonderkiln.camerakit.CameraView
    android:id="@+id/camera"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:adjustViewBounds="true" />
```

```
@Override
protected void onResume() {
    super.onResume();
    cameraView.start();
}

@Override
protected void onPause() {
    cameraView.stop();
    super.onPause();
}
```

捕捉图像

```
//录制音频
camera.setCameraListener(new CameraListener() {
    @Override
    public void onPictureTaken(byte[] picture) {
        super.onPictureTaken(picture);
        // 获取拍摄图像的二进制流,并转换为Bitmap
        Bitmap result = BitmapFactory.decodeByteArray(picture, 0,
picture.length);
    }
});
//捕捉图像
camera.captureImage();

//录制视频
camera.setCameraListener(new CameraListener() {
    @Override
    public void onVideoTaken(File video) {
        super.onVideoTaken(video);
        // 文件的格式是mp4格式
    }
});
//开始录制视频
camera.startRecordingVideo();
//2.5后停止录制视频
camera.postDelayed(new Runnable() {
```

```
    @Override
    public void run() {
        camera.stopRecordingVideo();
    }
}, 2500);
```

其他属性

ckFacing（back front）：设置相机朝脸或朝前，默认为朝前。

cameraView.setFacing(CameraKit.Constants.FACING_BACK);

ckFlash（off on auto）：设置拍照时是否打开闪光灯，默认是关闭。

cameraView.setFlash(CameraKit.Constants.FLASH_OFF);

ckFocus（off continuous tap）：设置是否聚焦，默认是持续。

cameraView.setFocus(CameraKit.Constants.FOCUS_CONTINUOUS);

ckMethod（standard still speed）：设置使用哪种捕捉方法，默认是标准。

//标准模式：使用通常的相机API和快门捕捉图像

cameraView.setMethod(CameraKit.Constants.METHOD_STANDARD);

//静态模式：适合更慢的相机

ckZoom（off pinch）：设置是否支持缩放，默认是否。

//速度模式：在捕捉了六张图像后会回到标准模式或静态模式（取决于相机速度）

cameraView.setMethod(CameraKit.Constants.METHOD_SPEED);

ckPermissions（strict lazy picture）：设置权限申请，默认是全部申请。

ckCropOutput（true false）：设置是否修剪输出流来适配相机边框，默认是否。

ckJpegQuality（0 <= n <= 100）：设置保存的jpeg的图片质量，默认是100。

ckVideoQuality（max480p max720p max1080p max2160p highest lowest）：

设置保存的视频质量，默认是max480p。

自动的权限申请

ameraView.setPermissions(CameraKit.Constants.PERMISSIONS_STRICT);

如果不需要调用CameraView.start()就无需手动申请相机权限。

动态的尺寸调节

cameraView.setCropOutput(true);

你可以将CameraView设置成任意尺寸。当CameraView的尺寸与实际预览界面的宽高比不相匹配时，

预览界面会最低限度地修剪来填充CameraView，类似于ImageView的android:scaleType="centerCrop"，

保证了图像的比例正确。

你可以给cameraView的宽高设置一个固定的尺寸(或match_parent)，

另外还需要使用android:adjustViewBounds="true"属性,这样就可以自动匹配实际预览界面的宽高比，

以获得完整而不变形的图像。

相机事件监听

```
camera.setCameraListener(new CameraListener() {

    @Override
    public void onCameraOpened() {
        super.onCameraOpened();
        //打开相机
    }

    @Override
```

```
    public void onCameraClosed() {
        super.onCameraClosed();
        //关闭相机
    }

    @Override
    public void onPictureTaken(byte[] picture) {
        super.onPictureTaken(picture);
        //获取到拍摄相片
    }

    @Override
    public void onVideoTaken(File video) {
        super.onVideoTaken(video);
        //获取到拍摄视频
    }

});
```