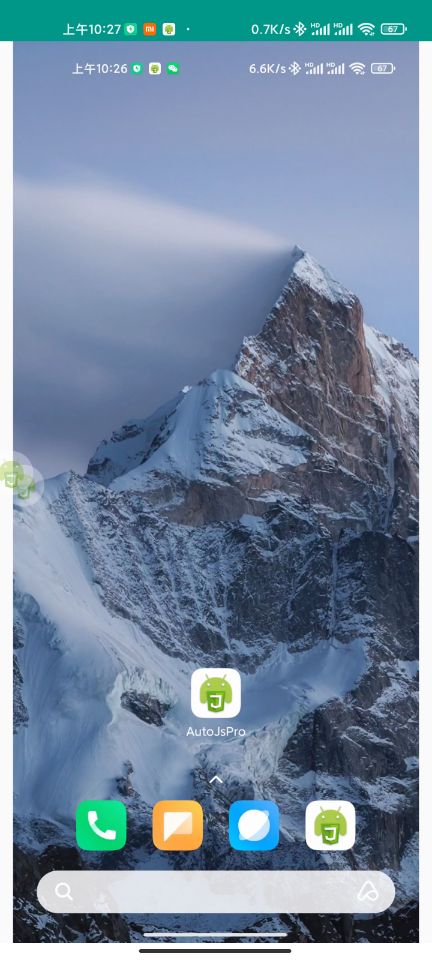
牙叔教程 简单易懂

**效果展示**



**为什么做**

有的app不允许截图, 但允许录屏,

我们要截图的时候, 就先录屏, 然后从MP4文件中提取图片

**环境**

手机: Mi 11 Pro

Android版本: 11

Autojs版本: 9.0.10

**代码流程**

1. 录屏
2. 从mp4提取图片

**备注**

* 申请录屏权限必须开启前台服务
* 录屏一般也需要录音权限
* 获取到权限之后, 可以把autojs本体切换到后台
* 提取图片的时候, 不要用opencv的Videoio, 他提取的图片会变色, 即使COLOR\_BGRA2RGBA, 颜色还是不一样

**名人名言**

思路是最重要的, 其他的百度, bing, stackoverflow, github, 安卓文档, autojs文档, 最后才是群里问问  
--- 牙叔教程

**声明**

部分内容来自网络  
本教程仅用于学习, 禁止用于其他用途

**bilibili**

[牙叔教程](https://space.bilibili.com/26079586)

**微信公众号 牙叔教程**



**QQ群**

747748653  


**完整源码**

"ui";

ui.layout(

<vertical>

<img id="img"></img>

</vertical>

);

function printObj(obj) {

var arr = [];

for (var k in obj) {

arr.push(k);

}

arr.sort();

log(arr);

}

// 先拍视频, 拍一秒, 然后视频中提取一帧

runtime.images.initOpenCvIfNeeded();

importClass(android.content.Context);

importClass(android.media.MediaRecorder);

importClass(java.io.File);

importClass(java.lang.System);

importClass(android.os.Environment);

importClass(android.hardware.display.DisplayManager);

importClass(org.opencv.videoio.VideoCapture);

importClass(org.opencv.core.Mat);

importClass(org.opencv.imgproc.Imgproc);

importClass(org.opencv.videoio.Videoio);

importClass(org.opencv.imgcodecs.Imgcodecs);

importClass(android.media.MediaMetadataRetriever);

/\* -------------------------------------------------------------------------- \*/

$settings.setEnabled("foreground\_service", true);

runtime.requestPermissions(["RECORD\_AUDIO"]);

/\* -------------------------------------------------------------------------- \*/

let videoFilePath = files.path("/sdcard/脚本/record.mp4");

let running = false;

let mediaRecorder;

let virtualDisplay;

let mediaProjection;

let dpi = 6000000;

let width = device.width;

let height = device.height;

let img;

// 最低200毫秒, 再低就报错了

let max\_duration\_ms = 200;

/\* -------------------------------------------------------------------------- \*/

// 申请录屏权限

let userHasAuthorizedClient = false;

let mediaProjectionManager = context.getSystemService(Context.MEDIA\_PROJECTION\_SERVICE);

let screenCaptureIntent = mediaProjectionManager.createScreenCaptureIntent();

let SCREEN\_CAPTURE\_REQUEST\_CODE = 10012;

ui.emitter.on("activity\_result", (requestCode, resultCode, data) => {

log("requestCode = " + requestCode);

log("resultCode = " + resultCode);

log("data = " + data);

if (requestCode === SCREEN\_CAPTURE\_REQUEST\_CODE) {

if (resultCode === -1) {

toastLog("用户 同意 录屏");

userHasAuthorizedClient = true;

mediaProjection = mediaProjectionManager.getMediaProjection(resultCode, data);

if (mediaProjection) {

initMediaRecorder();

virtualDisplay = mediaProjection.createVirtualDisplay(

"牙叔教程",

width,

height,

dpi,

DisplayManager.VIRTUAL\_DISPLAY\_FLAG\_PUBLIC,

mediaRecorder.getSurface(),

null,

null

);

setTimeout(function () {

activity.moveTaskToBack(true);

mediaRecorder.start();

running = true;

}, 2000);

}

} else {

toastLog("用户 不同意 录屏");

}

}

// mediaProjection = mediaProjectionManager.getMediaProjection(resultCode, data);

// if (mediaProjection) {

// startRecord();

// }

});

activity.startActivityForResult(screenCaptureIntent, 10012);

function viewMat(mat) {

let mat2 = mat.clone();

// Imgproc.cvtColor(mat, mat2, Imgproc.COLOR\_BGRA2RGBA);

let tempFilePath = files.join(files.getSdcardPath(), "脚本", "mat.png");

Imgcodecs.imwrite(tempFilePath, mat2);

mat2.release();

app.viewFile(tempFilePath);

}

function showImg() {

let filePath = videoFilePath;

log(files.exists(filePath));

if (!files.exists(filePath)) {

throw new Error("视频文件不存在, " + filePath);

}

let object = new MediaMetadataRetriever();

object.setDataSource(filePath);

//frameTime的单位为us微秒

let frameTime = max\_duration\_ms \* 900;

let frame = object.getFrameAtTime(frameTime, MediaMetadataRetriever.OPTION\_CLOSEST);

log(frame);

log(frame.width);

log(frame.height);

ui.img.setImageBitmap(frame);

}

function showImg2() {

let filePath = videoFilePath;

log(files.exists(filePath));

if (!files.exists(filePath)) {

throw new Error("视频文件不存在, " + filePath);

}

let cap = new VideoCapture(filePath);

log(cap);

log(cap.isOpened());

let frame = new Mat();

cap.set(Videoio.CAP\_PROP\_POS\_AVI\_RATIO, 0.9);

cap.read(frame);

let h = frame.rows();

let w = frame.cols();

log("w = " + w + ", h = " + h);

// Imgproc.cvtColor(frame, frame, Imgproc.COLOR\_BGRA2RGBA);

// img = com.stardust.autojs.core.image.ImageWrapper.ofMat(frame);

// ui.img.setImageBitmap(img.bitmap);

viewMat(frame);

cap.release();

frame.release();

}

function initMediaRecorder() {

mediaRecorder = new MediaRecorder();

mediaRecorder.setOnInfoListener(

new MediaRecorder.OnInfoListener({

onInfo: function (mr, what, extra) {

log("what = " + what);

if (what == MediaRecorder.MEDIA\_RECORDER\_INFO\_MAX\_DURATION\_REACHED) {

stopRecord();

log("1111111111");

toastLog("到达指定时长, 停止录屏");

showImg();

}

},

})

);

// 设置音频来源 需要动态申请 android.permission.RECORD\_AUDIO 权限

mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);

// 设置视频来源

mediaRecorder.setVideoSource(MediaRecorder.VideoSource.SURFACE);

// 设置输出格式

mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG\_4);

// 设置输出文件

let absolutePath = new File(videoFilePath).getAbsolutePath();

mediaRecorder.setOutputFile(absolutePath);

// 设置视频宽高

mediaRecorder.setVideoSize(width, height);

// 设置视频帧率

mediaRecorder.setVideoFrameRate(30);

// 设置视频编码比特率

mediaRecorder.setVideoEncodingBitRate(9000000);

// 设置音频编码

mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);

// 设置视频编码

mediaRecorder.setVideoEncoder(MediaRecorder.VideoEncoder.H264);

mediaRecorder.setMaxDuration(max\_duration\_ms);

// log(r);

mediaRecorder.prepare();

}

events.on("exit", function () {

if (running) {

stopRecord();

}

img && img.recycle();

toastLog("结束运行");

// app.viewFile(videoFilePath);

});

function stopRecord() {

if (!running) {

return true;

}

mediaRecorder.stop();

mediaRecorder.reset();

mediaRecorder.release();

virtualDisplay.release();

mediaProjection.stop();

running = false;

}