

6장 카메라를 이용한 앱 만들기

- 안드로이드 8.0 및 안드로이드 8.1.0 의 이슈 대응

들어가기 전에.

책의 집필이 완료 되었던 17년 10월 30일을 기준으로 안드로이드의 8.0 베타에서 테스트되었던 카메라 예제는 18년 구글과 각 제조사가 안드로이드 8.0 과 오레오의 마이너 업데이트인 안드로이드 8.1.0 을 업데이트 함으로써 일부 기기에 따라 정상 동작하지 않는 이슈가 발생하였습니다.

책에 실린 예제는 하드웨어에서 지원하는 최대 촬영 해상도를 구하여 Camera2 API 를 사용해 비디오 녹화를 시도 합니다. 몇몇의 최신버전의 기기에서 안드로이드 Camera2 API 에 맞게 동작하지 못함을 확인 하였습니다.

만약 여러분의 테스트 기기에서 정상 동작 한다면 책에 실린 예제를 그대로 사용하셔도 좋습니다.

문제가 발생한다면 단순히 동영상의 녹화 해상도를 1280 x 720 으로 낮추면 책의 예제를 그대로 사용할수도 있습니다.

책에 실린 예제는 구글이 공개한 안드로이드의 표준 Camera2 API 의 세부적인 활용과 설정 방법을 설명한것이고 이 API 가 변경된것은 아니기 때문에 해당 사용방법을 숙지하고 자신이 만들 앱에 맞추어 설정을 변경해 가면서 조정하는 것도 하나의 방법 입니다.

안드로이드의 버전은 끊임없이 바뀌고 있고, 특히 하드웨어를 사용하는 앱의 경우, 각 제조사의 차이 와 버전의 차이로 구글에서 제시한 같은 API 도 제조사의 하드웨어 버전 업그레이드 와 함께 동작하지 않거나, 동작하지 않았던 기능도 추후 다시 동작하거나 하는 일들이 발생합니다.

개발자는 지원 기기의 버전 업데이트의 변경 내용을 파악하고 오류없이 지원할수 있도록 항상 자신의 앱을 업데이트 해 가야 합니다.

18년 3월을 기준으로 이슈가 발생하는 기기를 포함하여 , 앱이 실행되는 현재 기기의 최적의 사이즈를 찾아서 정상 동작하도록 하는 추가 카메라 예제를 다음과 같이 설명드립니다.

정상 동작하지 않는 기기의 목록은 따로 공개하지 않으며, 다음의 내용은 책과 같은 어조로 서술함을 이해 부탁드립니다.

2018년 3월 7일 저자. 임규민, 박헌우.

6장 4절까지 완성된 예제에서 RecordActivity 의 startRecording 메서드를 아래와 같이 수정한다.

<RecordActivity.java>

```
package com.practice.sample.cameramedia;

...

public class RecordActivity extends Activity {

    ...

    private void startRecording() {
        recordButton.setText("중지");

        if (mediaRecorder == null) {
            mediaRecorder = new MediaRecorder();
        }

        String recordFilePath = getOutputMediaFile().getAbsolutePath();

        mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        mediaRecorder.setVideoSource(MediaRecorder.VideoSource.SURFACE);

        // 현재 앱이 실행되는 카메라가 지원하는
        // 최대 화질의 설정 프로필을 구한다.
        // 안드로이드 기기에 탑재되는 카메라에 따라
        // 결정되는 녹화 사이즈 등은 모두 달라진다.
01 CamcorderProfile camcorderProfile
01     = CamcorderProfile.get(CamcorderProfile.QUALITY_HIGH);

        // 일부 기종에서 최대 화질 프로파일 과
        // 카메라 하드웨어 가 지원하는 최대 사이즈가 다른 문제가 발생한다.
        // 실제 카메라 하드웨어로 부터 구해진 previewSize 와
        // 최대 화질 프로파일 API 를 통해 구해진 최대 지원 사이즈를 비교해서
        // 더 작은 쪽으로 설정을 바꾼다.
02 if (camcorderProfile.videoFrameWidth > previewSize.getWidth()
02     || camcorderProfile.videoFrameHeight > previewSize.getHeight()) {
03     camcorderProfile.videoFrameWidth = previewSize.getWidth();
04     camcorderProfile.videoFrameHeight = previewSize.getHeight();
04 }

05 mediaRecorder.setProfile(camcorderProfile);
06 mediaRecorder.setOutputFile(recordFilePath);
06 mediaRecorder.setOrientationHint(90);

        // 넥서스 5x 의 OS 버전 8.1.0 에서는
        // 다른 기종들과는 반대로 화면을 회전해야 하는 문제가 발생한다.
        // 테스트폰이 넥서스 5x 라면 다음 줄을 주석 해제 한다.
07 // mediaRecorder.setOrientationHint(270);

        try {
            mediaRecorder.prepare();
        } catch (IOException e) {
            e.printStackTrace();
            return;
        }

        List<Surface> surfaces = new ArrayList<>();

        SurfaceTexture surfaceTexture = textureView.getSurfaceTexture();
        Surface previewSurface = new Surface(surfaceTexture);
        surfaces.add(previewSurface);
    }
}
```

	<pre> Surface mediaRecorderSurface = mediaRecorder.getSurface(); surfaces.add(mediaRecorderSurface); try { previewBuilder = cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_RECORD); previewBuilder.addTarget(previewSurface); previewBuilder.addTarget(mediaRecorderSurface); cameraDevice.createCaptureSession(surfaces, captureStateCallback, null); mediaRecorder.start(); } catch (CameraAccessException e) { e.printStackTrace(); } } ... } </pre>
	<ul style="list-style-type: none"> 01. 시스템으로부터 동영상 녹화 설정값 가져옴 02. 구해진 동영상 녹화 설정값이 하드웨어가 지원하는 최대 해상도보다 클 경우 03. 동영상 녹화 설정값의 동영상 가로 크기를 하드웨어 지원 해상도로 변경 04. 동영상 녹화 설정값의 동영상 세로 크기를 하드웨어 지원 해상도로 변경 05. MediaRecorder 에 동영상 녹화 설정값을 지정함 06. 기존 예제와 다르지 않음 07. 넥서스5x 8.1.0 의 경우 타 기종과 다르게 반대로 회전을 해야 정상 녹화됨

CamcorderProfile 클래스의 get() 메서드는 사용자가 원하는 동영상 퀄리티에 따라 자동으로 녹화 설정값을 CamcorderProfile 객체에 담아서 반환한다. 해당 메서드를 사용하여 기기의 종류에 상관없이 해당 기기가 지원하는 녹화 설정을 자동으로 얻어 낼수 있다.

CamcorderProfile 의 get() 메서드에 사용할수 있는 동영상 퀄리티는 다음과 같다.

옵션	설명	지원 API
QUALITY_1080P	영상의 퀄리티를 1080p (1920x1080) 해상도로 설정한다	11
QUALITY_2160P	영상의 퀄리티를 2160p (3840x2160) 해상도로 설정한다	21
QUALITY_480P	영상의 퀄리티를 480p (720x480) 해상도로 설정한다	11
QUALITY_720P	영상의 퀄리티를 720p (1280x720) 해상도로 설정한다	11
QUALITY_CIF	영상의 퀄리티를 CIF (352x288) 해상도로 설정한다	11
QUALITY_HIGH	영상의 퀄리티를 기기가 사용가능한 최고 해상도로 설정한다	8

QUALITY_HIGH_SPEED_1080P	영상의 퀄리티와 초당 프레임 수를 1080p 해상도, 고속(100fps 이상)으로 설정한다	21
QUALITY_HIGH_SPEED_2160P	영상의 퀄리티와 초당 프레임 수를 2160p 해상도, 고속(100fps 이상)으로 설정한다	21
QUALITY_HIGH_SPEED_480P	영상의 퀄리티와 초당 프레임 수를 480p 해상도, 고속(100fps 이상)으로 설정한다	21
QUALITY_HIGH_SPEED_720P	영상의 퀄리티와 초당 프레임 수를 720p 해상도, 고속(100fps 이상)으로 설정한다	21
QUALITY_HIGH_SPEED_HIGH	영상의 퀄리티를 기기가 사용가능한 최고 해상도와 고속(100fps 이상)으로 설정한다	21
QUALITY_HIGH_SPEED_LOW	영상의 퀄리티를 기기가 사용가능한 최저 해상도와 고속(100fps 이상)으로 설정한다	21
QUALITY_LOW	영상의 퀄리티를 기기가 사용가능한 최저 해상도로 설정한다	8
QUALITY_QCIF	영상의 퀄리티를 QCIF (176x144) 해상도로 설정한다	11
QUALITY_QVGA	영상의 퀄리티를 QVGA (320x240) 해상도로 설정한다	15
QUALITY_TIME_LAPSE_1080P	영상의 퀄리티를 1080p 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_LAPSE_2160P	영상의 퀄리티를 2160p 해상도의 타임랩스 용으로 설정한다	21
QUALITY_TIME_LAPSE_480P	영상의 퀄리티를 480p 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_LAPSE_720P	영상의 퀄리티를 720p 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_LAPSE_CIF	영상의 퀄리티를 CIF 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_LAPSE_HIGH	영상의 퀄리티를 기기가 사용가능한 최고 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_LAPSE_LOW	영상의 퀄리티를 기기가 사용가능한 최저 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_LAPSE_QCIF	영상의 퀄리티를 QCIF 해상도의 타임랩스 용으로 설정한다	11
QUALITY_TIME_	영상의 퀄리티를 QVGA 해상도의 타임랩스 용으로 설정한다	15

LAPSE_QVGA		
------------	--	--

표와 같이 동영상 퀄리티에 맞는 많은 설정값을 가져올수 있지만, 이 모든 옵션이 모든 기기에 다 동작하지 않는다. 현재 이 앱이 동작하는 하드웨어 지원 해상도 와 기기의 OS 버전에 따라 사용할수 있는 옵션이 다르니 반드시 확인하고 사용해야 한다.

이 예제에서는 안드로이드 2.2 부터 사용할수 있었던 설정값인 QUALITY_HIGH 를 이용해 대다수의 기기에서 돌아갈수 있도록 설정값을 구하도록 하였다.

CamcorderProfile 클래스의 get() 메서드를 이용해 CamcorderProfile 객체를 반환받으면 이 객체에는 동영상 녹화에 필요한 설정값들이 저장되어 있다. 다음의 표는 CamcorderProfile 객체의 설정값 필드 목록이다.

필드명	설명	지원 API
audioBitRate	오디오 출력 비트레이트 (비트/초)	8
audioChannels	오디오의 채널 수	8
audioCodec	인코딩에 사용되는 오디오 코덱	8
audioSampleRate	오디오의 샘플레이트	8
duration	세션이 종료되기 전의 기본 레코딩 지속 시간(초)	8
fileFormat	생성되는 파일의 출력 형식	8
quality	생성되는 파일의 퀄리티	8
videoBitRate	비디오 출력 비트레이트 (비트/초)	8
videoCodec	인코딩에 사용되는 비디오 코덱	8
videoFrameHeight	영상의 프레임 높이 (픽셀)	8
videoFrameRate	영상의 프레임 속도(초당 프레임 수)	8
videoFrameWidth	영상의 프레임 너비 (픽셀)	8

구글의 API 가이드 상으로 CamcorderProfile 클래스의 get() 메서드로 구해진 동영상 녹화 설정값은 그대로 사용이 가능해야 하지만, 일부 기종에서 자신의 카메라 하드웨어가 지원하지 못하는 해상도를 설정값으로 반환하는 버그가 발견되었다.

해당 기종의 버그는 분명 해당 제조사가 수정해야 할 버그이지만 제조사 수정 이전에 개발자는 이 앱이 모든 사용자의 기기에서 정상동작하도록 개발해야 한다.

해당 버그를 회피하여 정상동작하는 앱을 만들어 보자.

미리 구해놓은 preview 변수에 저장된 카메라 하드웨어의 최대 지원 해상도와 CamcorderProfile 객체에 저장된 동영상 녹화 가로, 세로 사이즈 값인 videoFrameWidth, videoFrameHeight 를 서로 비교해서 CamcorderProfile 객체에 지정된 사이즈가 더 크다면, 해당 기종은 영상을 녹화하지 못하고 앱이 강제 종료되게 된다.

이 경우 videoFrameWidth 와 videoFrameHeight 의 값을 카메라 하드웨어의 최대 해상도 값이 저장된 preview 의 값으로 변경한다.

이렇게 구해진 설정값을 MediaRecorder 의 setProfile() 메서드에 넘겨주면 기본 설정은 완료 된다. 책의 예제보다 간단하고 범용성도 가질수 있지만 개발자가 기본 설정할수 있는 많은 옵션들이 사라지는것에 주의해야 한다.

Tip.

동영상의 퀄리티 프로파일은 다음의 링크에서 모두 확인 가능하다.

<https://developer.android.com/reference/android/media/CamcorderProfile.html>

만약 여러분의 테스트 환경이 구글의 넥서스 5x 이고, 안드로이드 버전 8.1.0 이라면 해당 소스로 동영상을 촬영할 경우, 화면이 뒤집히는 반전 현상이 일어난다.

해당 현상은 구글의 안드로이드 폰인 픽셀, 픽셀 XL, 넥서스 6p 에서는 발생하지 않음을 확인하였고, 오직 넥서스 5x 8.1.0 버전에서 발생한다.

해당 이슈는 구글에서 업데이트로 수정해야 겠지만 이 경우도 회피할수 있는 코드를 여기에 제시한다.

MediaRecorder 의 setOrientationHint() 메서드는 현재 동영상의 녹화 방향을 지정한다. 넥서스 5x 를 제외한 안드로이드 기기에서는 세로 촬영할 경우 90도를 회전하면 영상이 정상 녹화 된다. 하지만 넥서스 5x 의 경우는 그 반대로 회전을 해야 하므로 270도를 회전하면 정상 녹화 된다.

화면이 뒤집힌다면 270도 설정 코드의 주석을 해제 하여 사용하도록 하자.

해당 예제를 포함하여 모든 예제는 저자의 깃허브를 통하여 다운로드 할수 있다.

<https://github.com/Fluxus-M/Android-Tutorials>

실무에 바로 적용 할 수 있는 안드로이드 8.0 앱 프로그래밍(Oreo 버전) NCS
국가직무능력표준 기준안 반영



실생활에 꼭 필요한 기능을 하는 예제를 만들며 배우는 안드로이드 프로그래밍!

구글 플레이 스토어의 상위권 앱들과 같은 동작을 하는 예제를 만들면서 초보자가 궁금해했던 부분을 쉽게 설명하고 있다. 각 장의 예제는 실무에 꼭 필요한 기술과 이론을 다루면서 초보자를 고급 개발자로 발전시킬 수 있는 길을 마련해준다.

각 장은 국가 직무 능력 표준인 NCS의 과정을 참고하여 작성되었으며 NCS 과정에 맞추어 기술을 익히고자 하는 이들에게 올바른 가이드라인이 되어 줄 것이다.

앱의 제작과 마켓을 이용한 배포까지 모든 과정을 한 권에 담고 있으므로 자신의 앱을 만들고 서비스를 하고자 하는 이들에게 큰 도움이 될 것이다.

이 책을 꼭 읽어야 할 분

- 자신만의 앱을 만들어 보고 싶은 분
- 구글 플레이 스토어의 앱들이 어떻게 만들어지는지 궁금한 분
- 안드로이드 개발을 처음 접하는 분
- 안드로이드 기술의 기본을 탄탄히 다지고 싶은 분
- 실무에 쓰이는 기술이 무엇인지 궁금한 분
- 오픈 API란 무엇이며 오픈 API를 어떻게 사용해야 하는지 궁금한 분
- NCS란 무엇이며 NCS 과정에 맞는 기술을 익히고자 하는 분



국립중앙도서관
수집번호
20180000000000000000

안드로이드 8.0

앱 프로그래밍

NCS 국가직무능력표준
기초안
20180000000000000000



NCS 국가직무능력표준
National Competency Standards
기초안
20180000000000000000



안드로이드 8.0 앱 프로그래밍

최신 안드로이드 Oreo 버전, 안드로이드 스튜디오 3.0 반영
NCS 기준에 맞춰 실무에 바로 적용할 수 있는 앱 소스 공개



박헌우, 임규민 지음 | 다본 | 2018년 02월 20일 출간