CHAP 15. 멀티미디어

안드로이드의 멀티미디어 지원

- □ 멀티미디어 재생->**MediaPlayer** 클래스를 사용
- □ 멀티미디어 녹화-> **MediaRecorder** 클래스를 사용

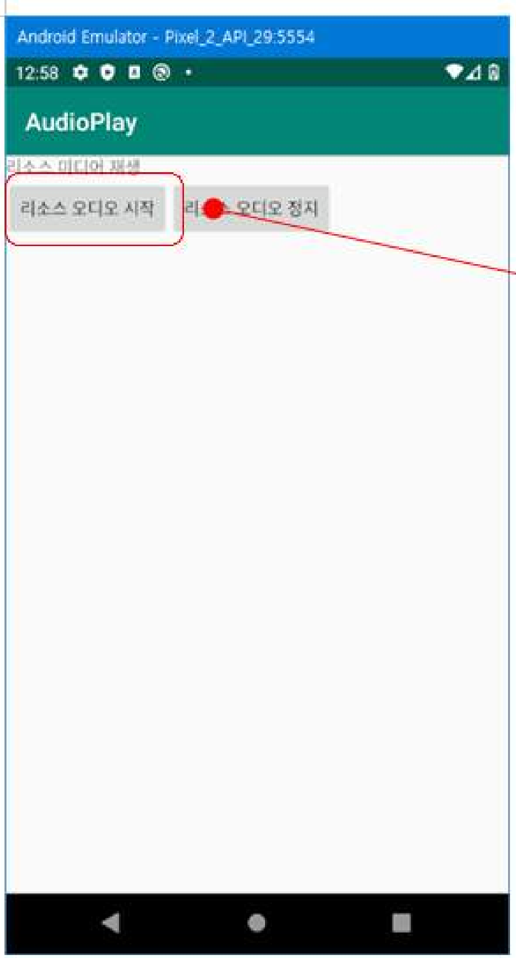
안드로이드 지원 파일 형식

종류	형식/코덱	엔코더	디코더	지원파일종류(파일확장자)
오디오	AAC LC/LTP	•	•	3GPP (.3gp) 와 MPEG-4 (.mp4, .m4a). raw AAC (.aac) 는 지원하지 않음
	HE-AACv1 (AAC+)		•	
	HE-AACv2 (enhanced AAC+)		•	
	AMR-NB	•	•	3GPP (.3gp)
	AMR-WB	•	•	3GPP (.3gp)
	MP3		•	MP3 (.mp3)
	MIDI		•	Type 0과 1 (.mid, .xmf, .mxmf). RTTTL/RTX (.rtttl, .rtx), OTA (.ota)와 iMelody (.imy)
	Ogg Vorbis		•	Ogg (.ogg)
	PCM/WAVE		•	WAVE (.wav)
이미지	JPEG	•	•	JPEG (.jpg)
	GIF		•	GIF (.gif)
	PNG	•	•	PNG (.png)
	BMP		•	BMP (.bmp)
비디오	H.263	•	•	3GPP (.3gp) 와 MPEG-4 (.mp4)
	H.264 AVC	•	•	3GPP (.3gp) 와 MPEG-4 (.mp4)
	MPEG-4 SP		•	3GPP (.3gp)
	VP8		•	WebM (.webm)

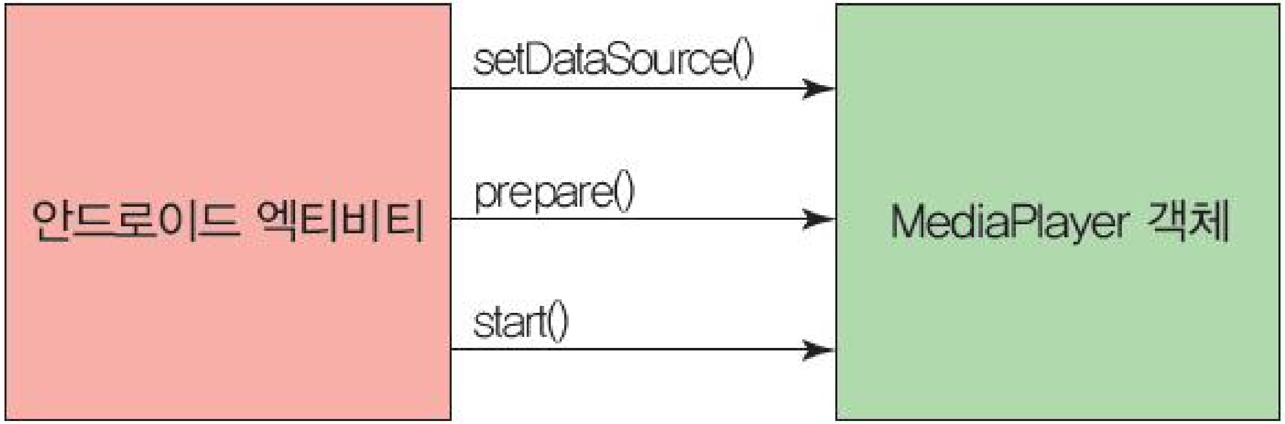
오디오 재생

- 2가지의 방법
  - 인텐트 사용
  - **MediaPlayer** 클래스 사용

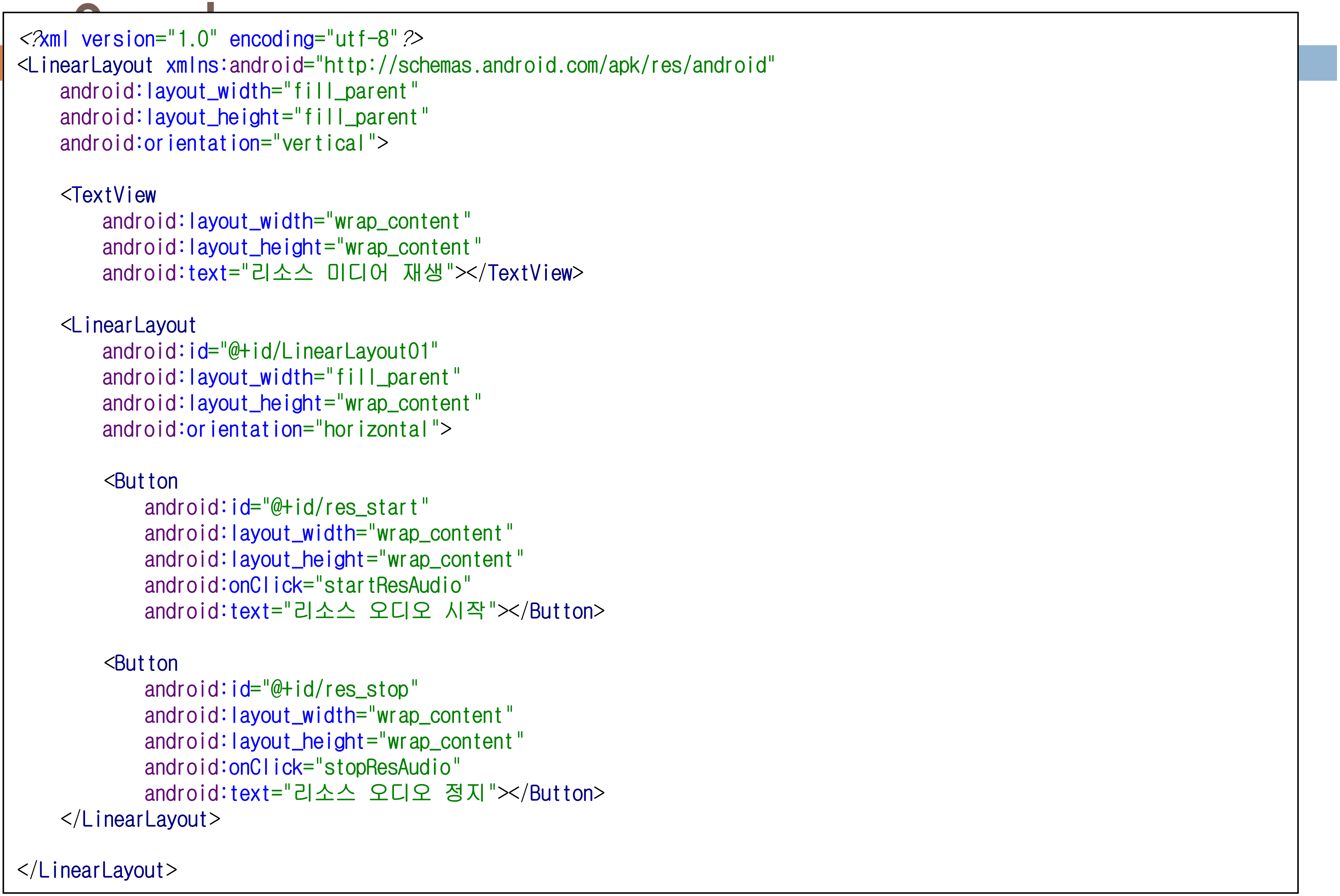
MediaPlayer 클래스를 사용한 오디오 예제



MediaPlayer 클래스



사용자 인터페이스



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="리소스 미디어 재생"></TextView>

    <LinearLayout
        android:id="@+id/LinearLayout01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/res_start"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="startResAudio"
            android:text="리소스 오디오 시작"></Button>

        <Button
            android:id="@+id/res_stop"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="stopResAudio"
            android:text="리소스 오디오 정지"></Button>
    </LinearLayout>

</LinearLayout>
```

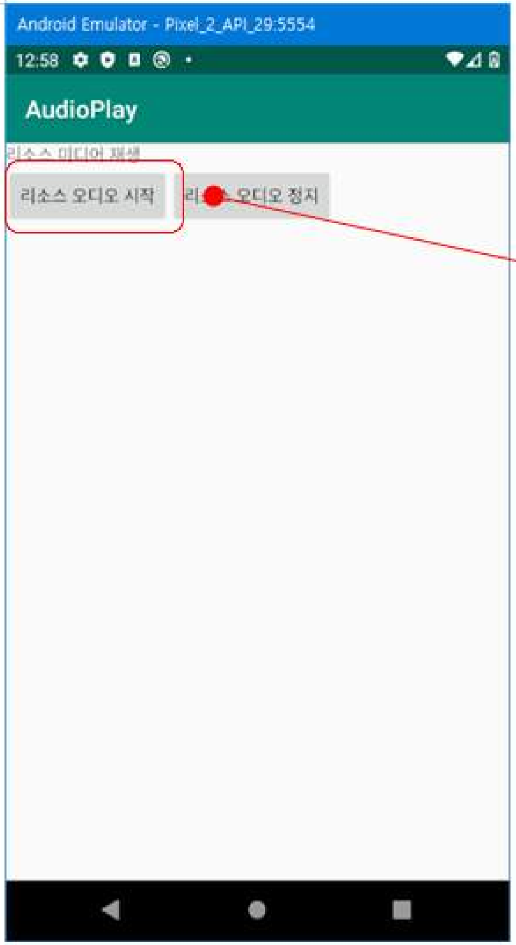
코드 작성

```java
public class MainActivity extends AppCompatActivity {
    MediaPlayer mp = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void startResAudio(View v) {
        mp = MediaPlayer.create(this, R.raw.old_pop);
        mp.start();
    }

    public void stopResAudio(View v) {
        if (mp != null) {
            mp.stop();
            mp.release();
        }
        mp = null;
    }

}
```

매니페스트 파일

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.audioplay">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

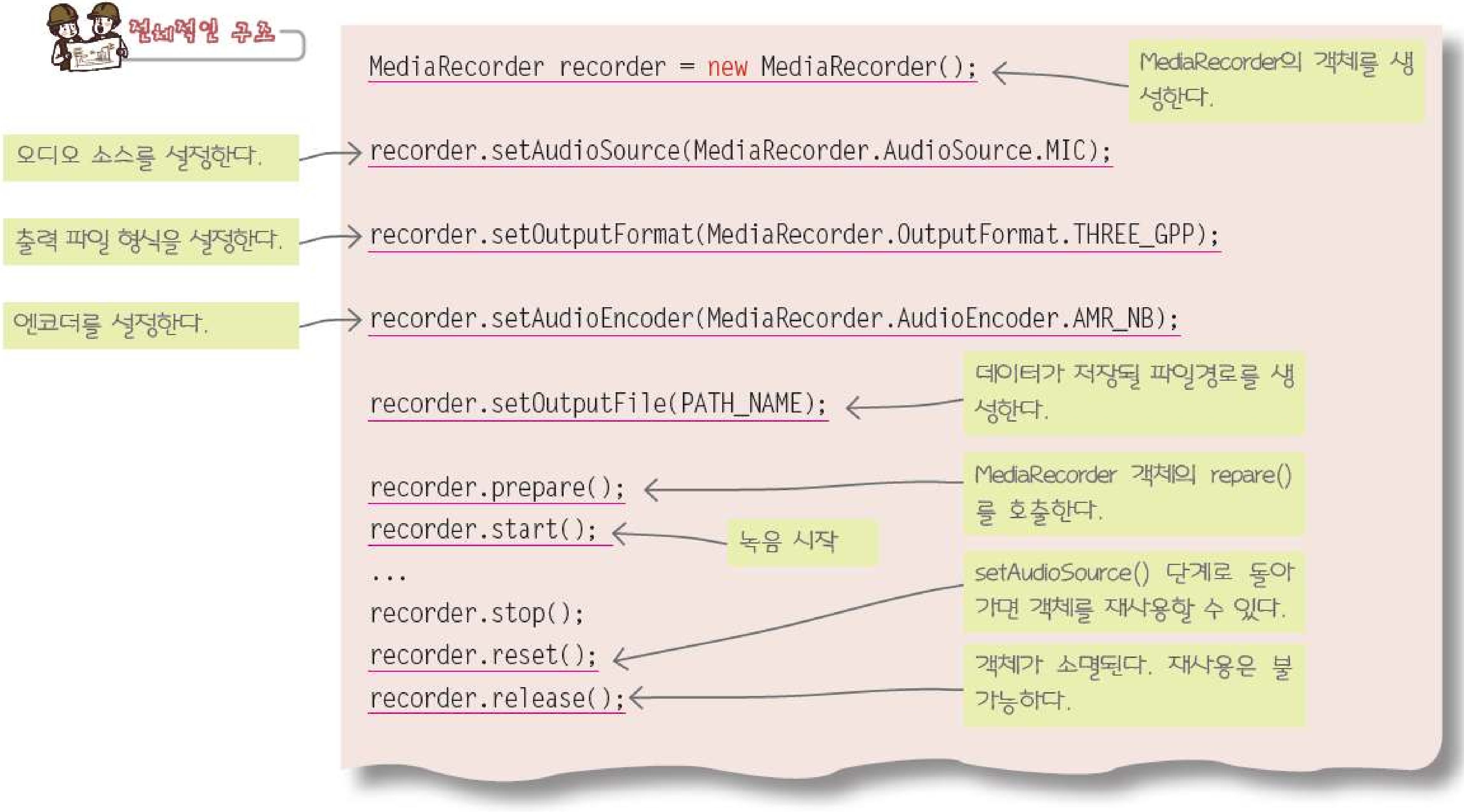
실행 결과


Android Emulator - Pixel_2_API_29:5554
12:58
AudioPlay
리소스 미디어 재생
리소스 오디오 시작
리소스 오디오 정지

오디오 녹음

- 2가지의 방법
  - 인텐트 사용
  - **MediaRecorder** 클래스 사용

MediaRecorder를 사용하여서 오디오 녹음



전체적인 구조

```
MediaRecorder recorder = new MediaRecorder();
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
recorder.setOutputFile(PATH_NAME);
recorder.prepare();
recorder.start();
...
recorder.stop();
recorder.reset();
recorder.release();
```

MediaRecorder의 객체를 생성한다.

오디오 소스를 설정한다.

출력 파일 형식을 설정한다.

엔코더를 설정한다.

데이터가 저장될 파일경로를 생성한다.

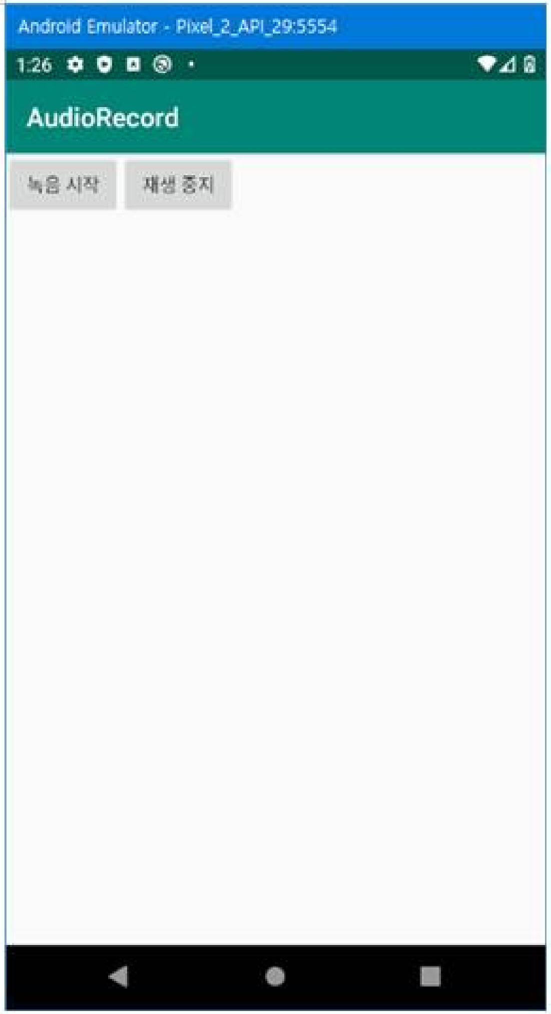
MediaRecorder 객체의 repare()를 호출한다.

녹음 시작

setAudioSource() 단계로 돌아가면 객체를 재사용할 수 있다.

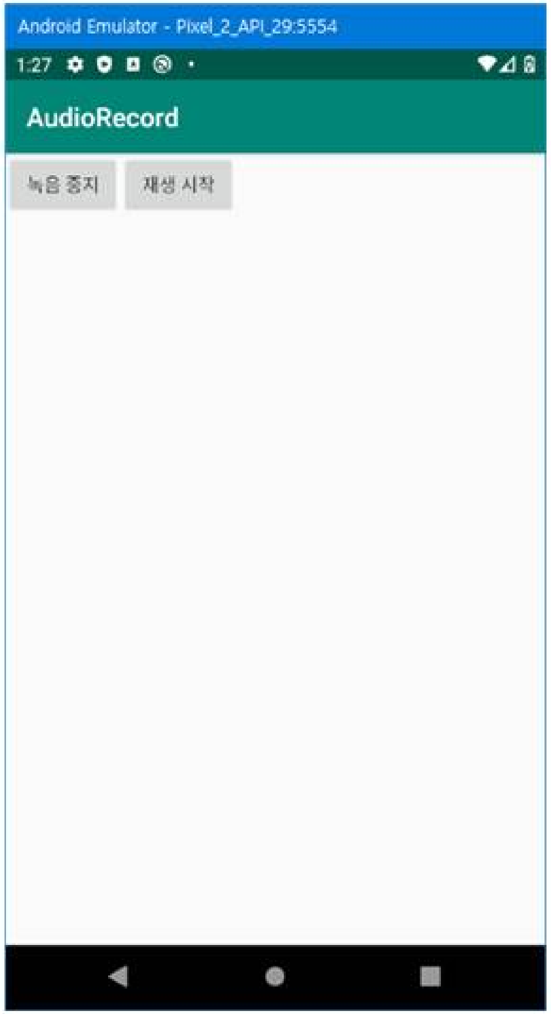
객체가 소멸된다. 재사용은 불가능하다.

실행 결과


Android Emulator - Pixel_2_API_29:5554
1:26
AudioRecord
녹음 시작
재생 중지

->

->


Android Emulator - Pixel_2_API_29:5554
1:27
AudioRecord
녹음 중지
재생 시작

예제

```java
public class MainActivity extends AppCompatActivity {

    private static final String LOG_TAG = "AudioRecordTest";
    private static final int REQUEST_RECORD_AUDIO_PERMISSION = 200;
    private static String fileName = null;

    private RecordButton recordButton = null;
    private MediaRecorder recorder = null;

    private PlayButton   playButton = null;
    private MediaPlayer   player = null;

    // Requesting permission to RECORD_AUDIO
    private boolean permissionToRecordAccepted = false;
    private String [] permissions = {Manifest.permission.RECORD_AUDIO};
```

예제

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    switch (requestCode){
        case REQUEST_RECORD_AUDIO_PERMISSION:
            permissionToRecordAccepted  = grantResults[0] == PackageManager.PERMISSION_GRANTED;
            break;
    }
    if (!permissionToRecordAccepted ) finish();

}

private void onRecord(boolean start) {
    if (start) {            startRecording();        }
    else {            stopRecording();        }
}

private void onPlay(boolean start) {
    if (start) {            startPlaying();        }
    else {            stopPlaying();        }
}
```

예제

```
private void startPlaying() {
    player = new MediaPlayer();
    try {
        player.setDataSource(fileName);
        player.prepare();
        player.start();
    } catch (IOException e) {
        Log.e(LOG_TAG, "prepare() failed");
    }
}

private void stopPlaying() {
    player.release();
    player = null;
}
```

예제

```
private void startRecording() {
    recorder = new MediaRecorder();
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    recorder.setOutputFile(fileName);
    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);

    try {              recorder.prepare();        }
    catch (IOException e) {            Log.e(LOG_TAG, "prepare() failed");        }

    recorder.start();
}

private void stopRecording() {
    recorder.stop();
    recorder.release();
    recorder = null;
}
```

```java
class RecordButton extends AppCompatButton {
    boolean mStartRecording = true;

    OnClickListener clicker = new OnClickListener() {
        public void onClick(View v) {
            onRecord(mStartRecording);
            if (mStartRecording) {                    setText("녹음 중지");                }
            else {                    setText("녹음 시작");                }
            mStartRecording = !mStartRecording;
        }
    };

    public RecordButton(Context ctx) {
        super(ctx);
        setText("녹음 시작");
        setOnClickListener(clicker);
    }
}

class PlayButton extends AppCompatButton {
    boolean mStartPlaying = true;

    OnClickListener clicker = new OnClickListener() {
        public void onClick(View v) {
            onPlay(mStartPlaying);
            if (mStartPlaying) {                    setText("재생 중지");                }
            else {                    setText("재생 시작");                }
            mStartPlaying = !mStartPlaying;
        }
    };

    public PlayButton(Context ctx) {
        super(ctx);
        setText("재생 시작");
        setOnClickListener(clicker);
    }
}
```

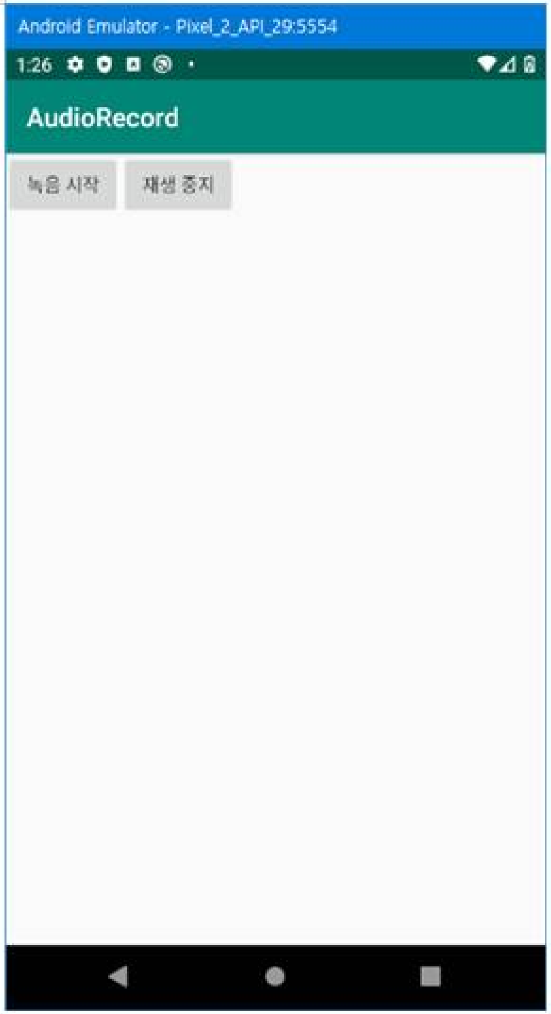
```
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);

    // 외부 캐시 디렉토리에 저장한다.
    fileName = getExternalCacheDir().getAbsolutePath();
    fileName += "/audiorecordtest.3gp";

    ActivityCompat.requestPermissions(this, permissions, REQUEST_RECORD_AUDIO_PERMISSION);

    LinearLayout ll = new LinearLayout(this);
    recordButton = new RecordButton(this);
    ll.addView(recordButton,
            new LinearLayout.LayoutParams(
                    ViewGroup.LayoutParams.WRAP_CONTENT,
                    ViewGroup.LayoutParams.WRAP_CONTENT,
                    0));
    playButton = new PlayButton(this);
    ll.addView(playButton,
            new LinearLayout.LayoutParams(
                    ViewGroup.LayoutParams.WRAP_CONTENT,
                    ViewGroup.LayoutParams.WRAP_CONTENT,
                    0));
    setContentView(ll);
}
```

```
    @Override
    public void onStop() {
        super.onStop();
        if (recorder != null) {
            recorder.release();
            recorder = null;
        }

        if (player != null) {
            player.release();
            player = null;
        }
    }
}
```

매니페스트 파일

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.audiorecord">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
    <uses-permission android:name="android.permission.RECORD_AUDIO"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

실행 결과



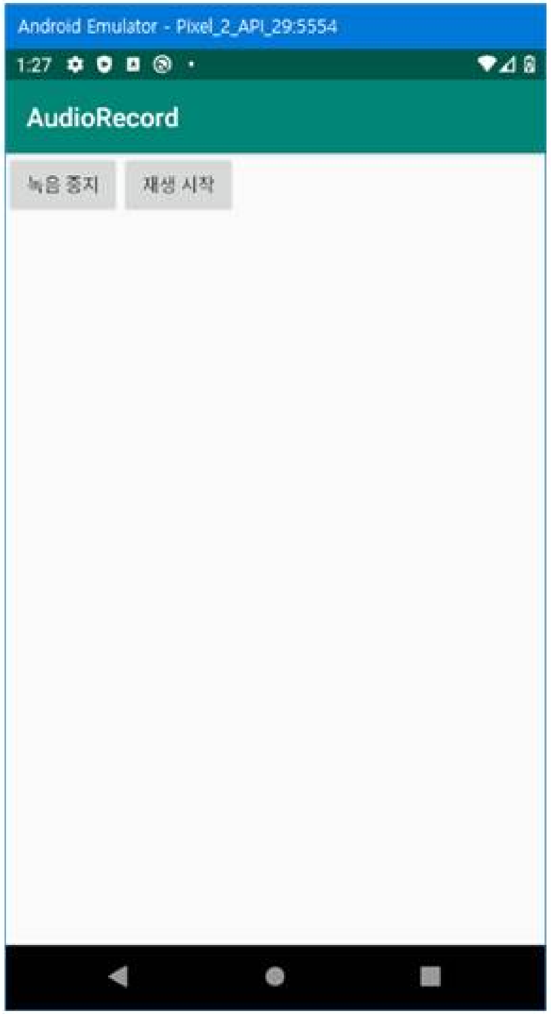
Android Emulator - Pixel_2_API_29:5554

1:26

AudioRecord

->

->



Android Emulator - Pixel_2_API_29:5554

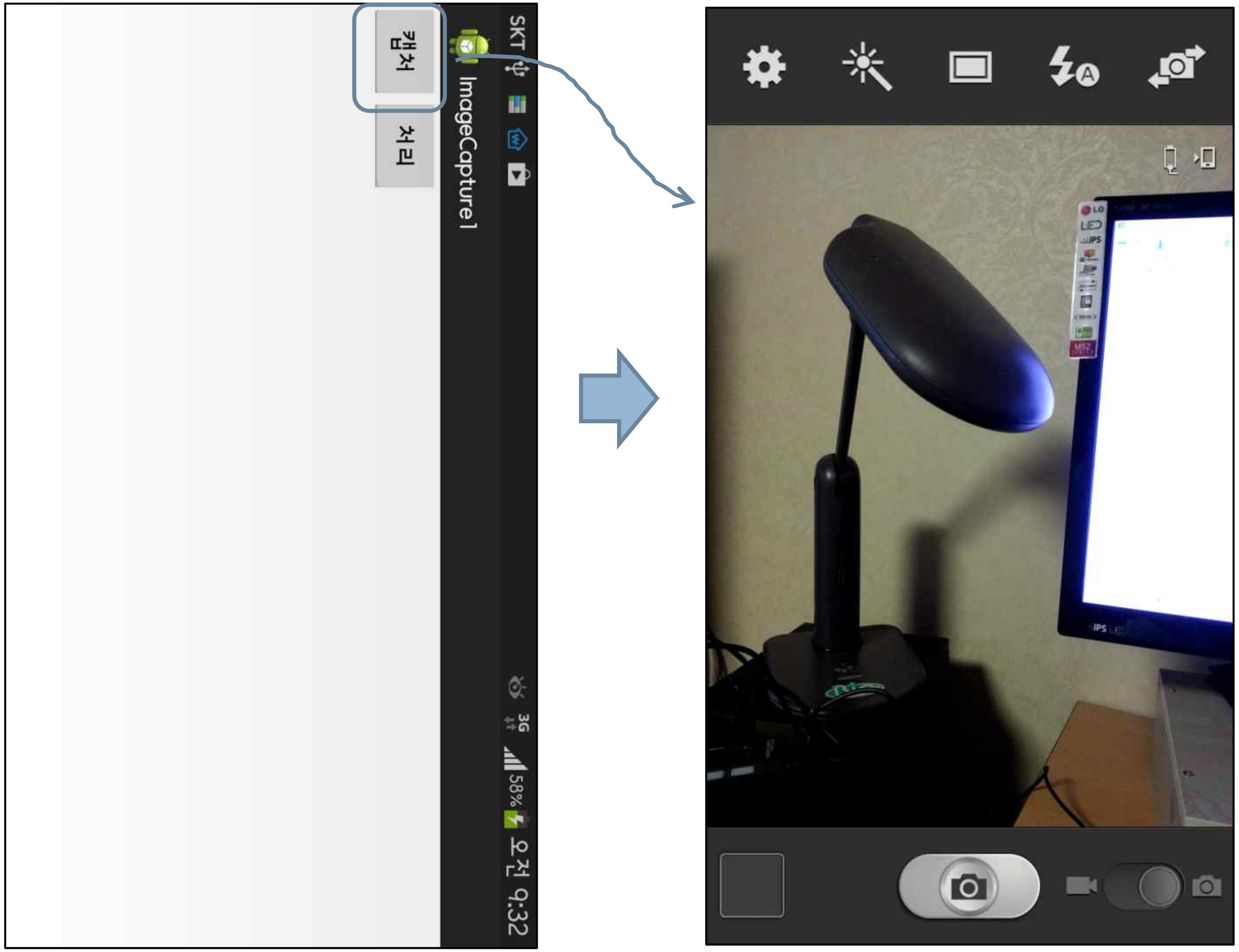
1:27

AudioRecord

이미지 캡처

- 2가지의 방법
  - 인텐트 사용
  - **Camera** 클래스 사용

예제. 인텐트를 이용한 이미지 획득



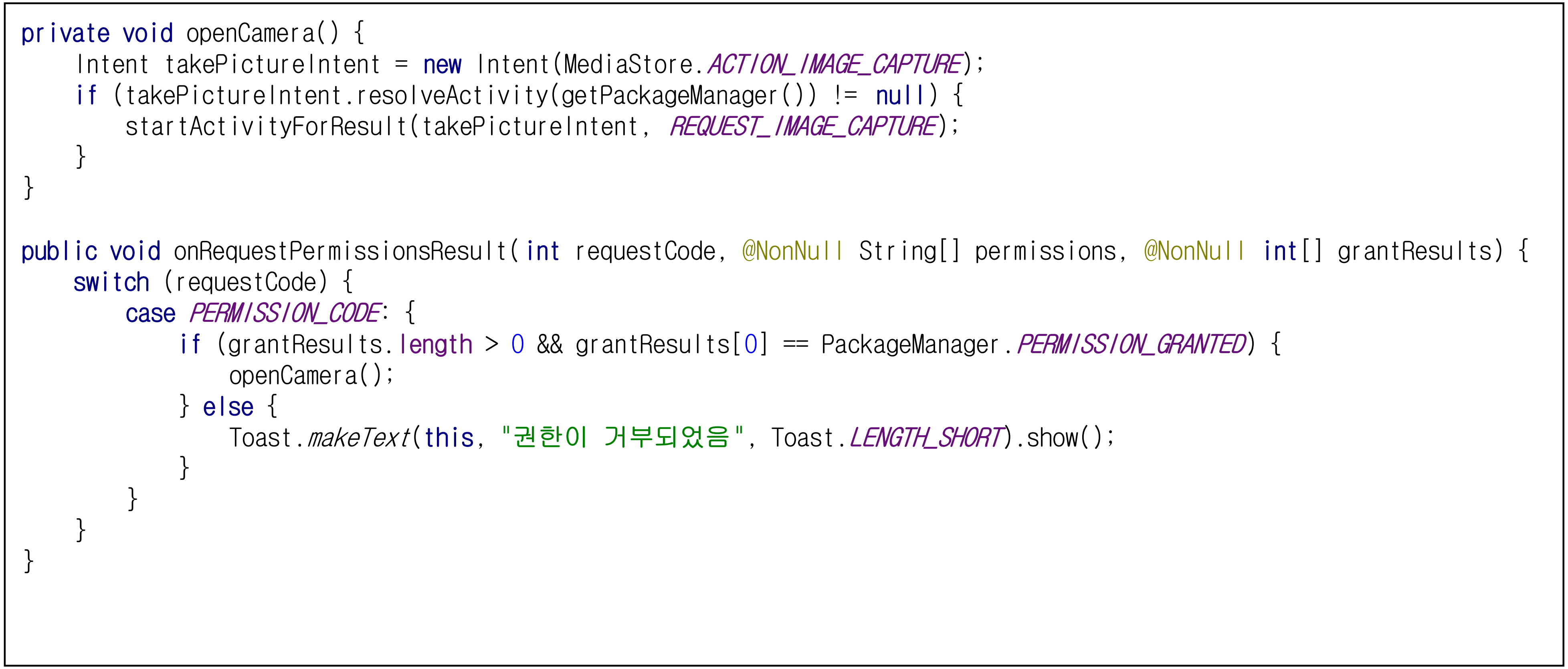
사용자 인터페이스 작성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:gravity="center_horizontal"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="카메라로부터 이미지를 캡처하려면 누르세요!"/>

    <Button
        android:id="@+id/btnCaptureImage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="capture"
        android:text="이미지 캡처" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="300dp"
        android:layout_height="500dp"
        android:src="@drawable/ic_launcher_foreground"
        android:layout_marginTop="3dp"/>
</LinearLayout>
```

코드 작성

```java
public class MainActivity extends AppCompatActivity {
    private static final int PERMISSION_CODE = 10;
    Button mCaptureBtn;
    ImageView imageView;
    static final int REQUEST_IMAGE_CAPTURE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView = findViewById(R.id.imageView);
    }

    public void capture(View v) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            String[] permission = {Manifest.permission.CAMERA, Manifest.permission.WRITE_EXTERNAL_STORAGE};
            requestPermissions(permission, PERMISSION_CODE);
        } else {
            openCamera();
        }
    }
```

코드 작성



```
private void openCamera() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
    }
}

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_CODE: {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                openCamera();
            } else {
                Toast.makeText(this, "권한이 거부되었음", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

코드 작성

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
            Bundle extras = data.getExtras();
            Bitmap imageBitmap = (Bitmap) extras.get("data");
            imageView.setImageBitmap(imageBitmap);
        }
    }
}
```

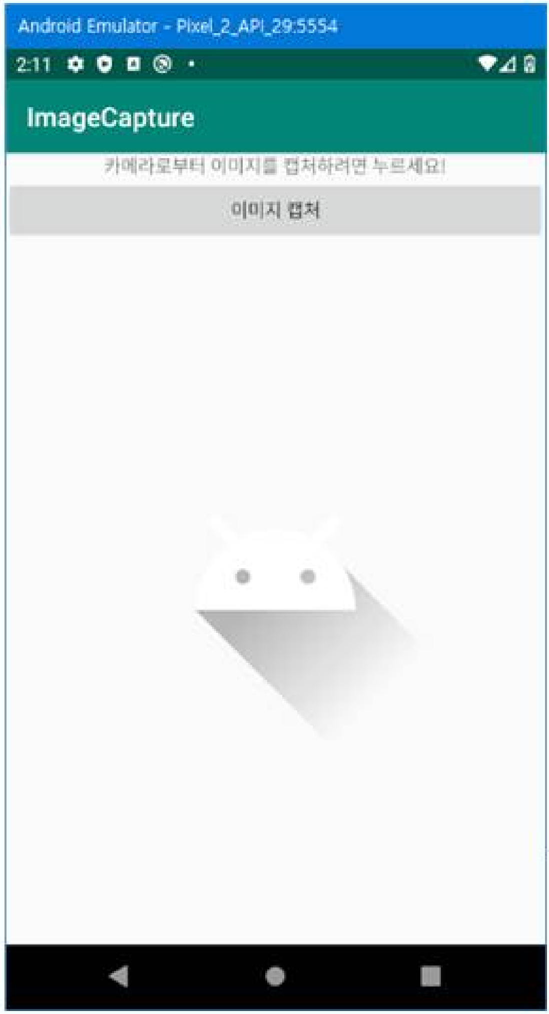
코드 작성

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
            Bundle extras = data.getExtras();
            Bitmap imageBitmap = (Bitmap) extras.get("data");
            imageView.setImageBitmap(imageBitmap);
        }
    }
}
```

매니페스트 파일

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.imagecapture">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

실행 결과



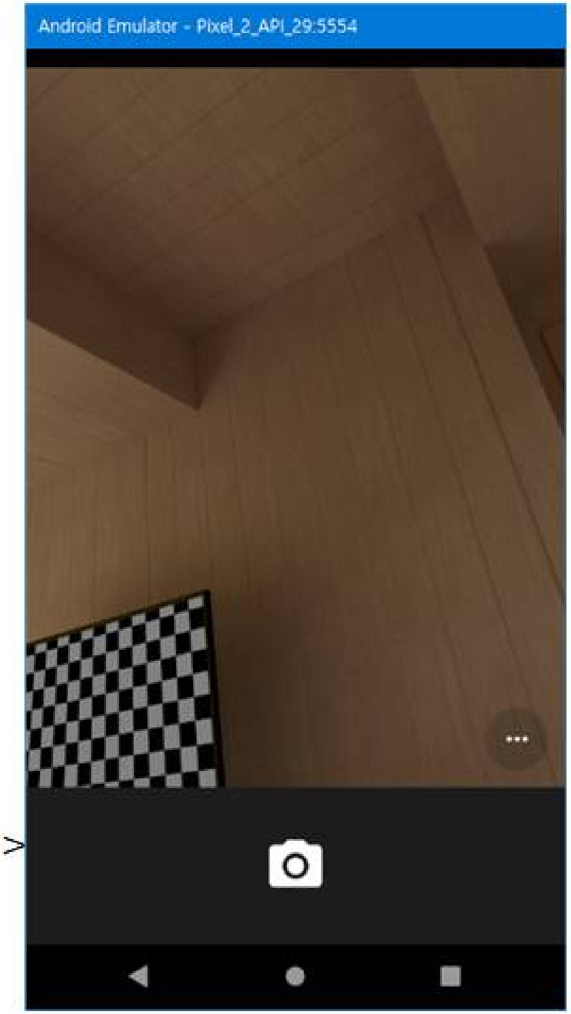
Android Emulator - Pixel_2_API_29:5554

2:11

ImageCapture

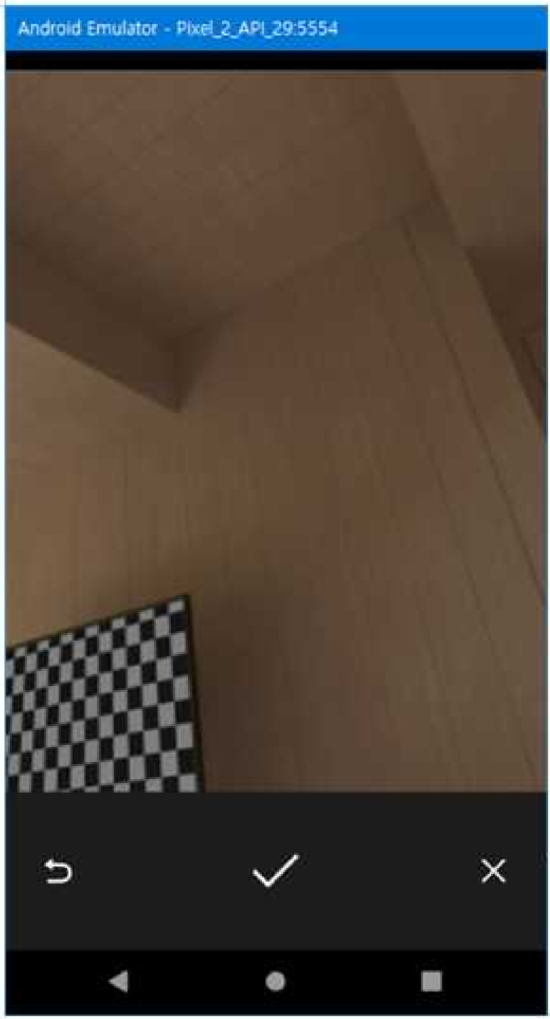
카메라로부터 이미지를 캡처하려면 누르세요!

->

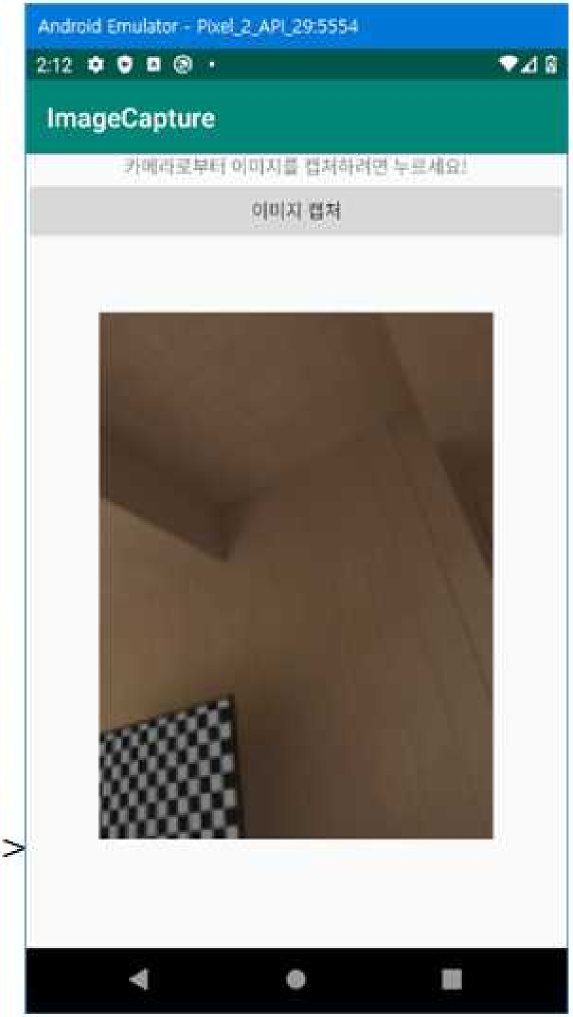


Android Emulator - Pixel_2_API_29:5554

실행 결과



->

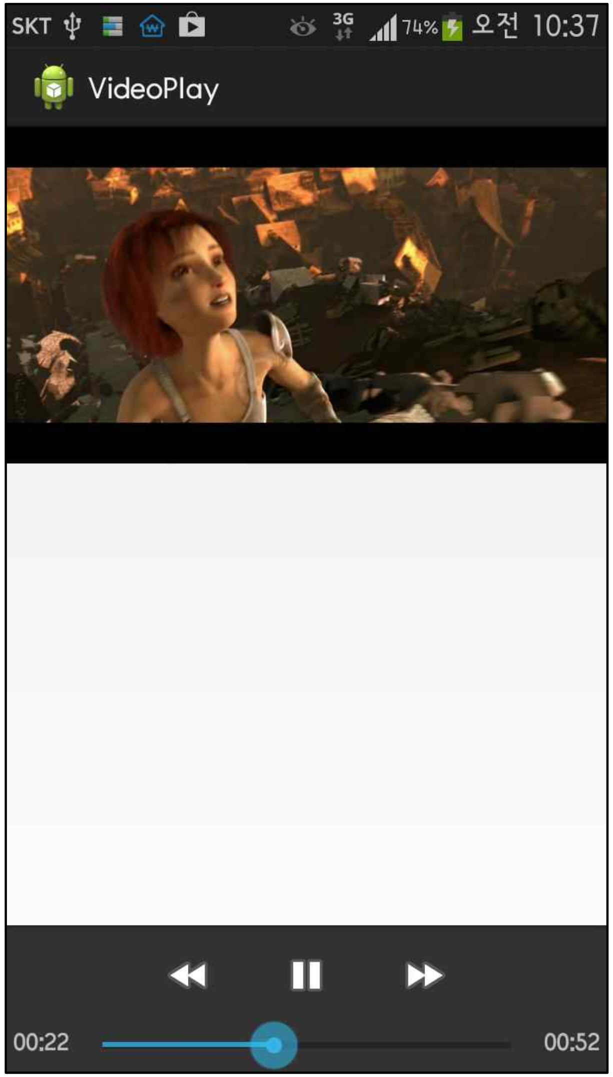


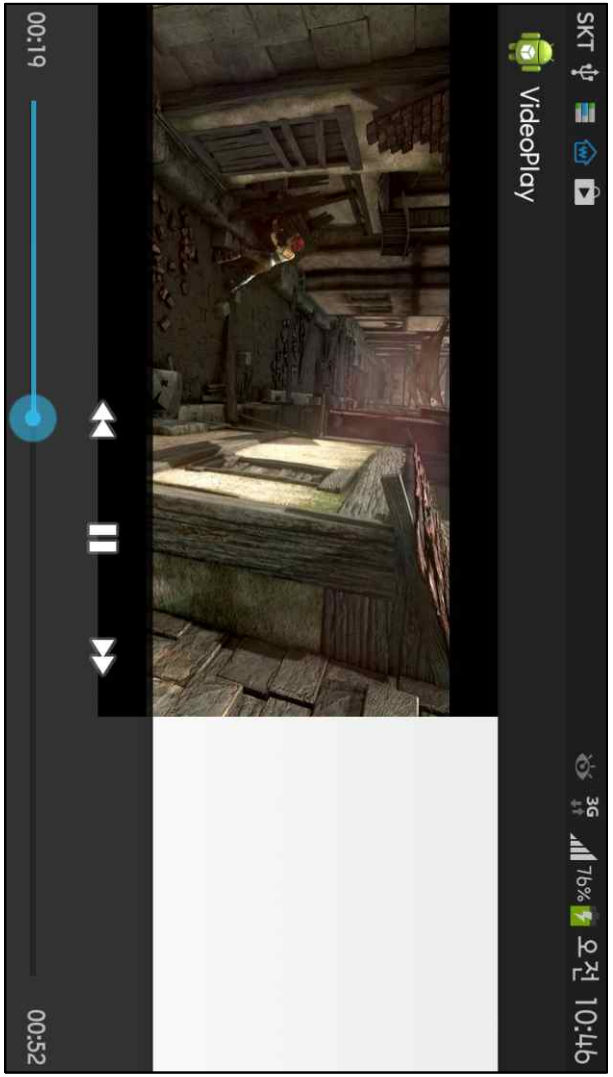
카메라로부터 이미지를 캡처하려면 누르세요!

비디오 재생

- MediaPlayer 클래스는 오디오 재생 뿐만 아니라 비디오 재생도 담당
- VideoView 클래스는 MediaPlayer 객체의 생성과 초기화를 담당한다.

예제. 비디오 재생


SKT
74%
오전 10:37
VideoPlay
00:22
00:52


SKT
76%
오전 10:46
VideoPlay
00:19
00:52

사용자 인터페이스 작성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <VideoView
        android:id="@+id/videoview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

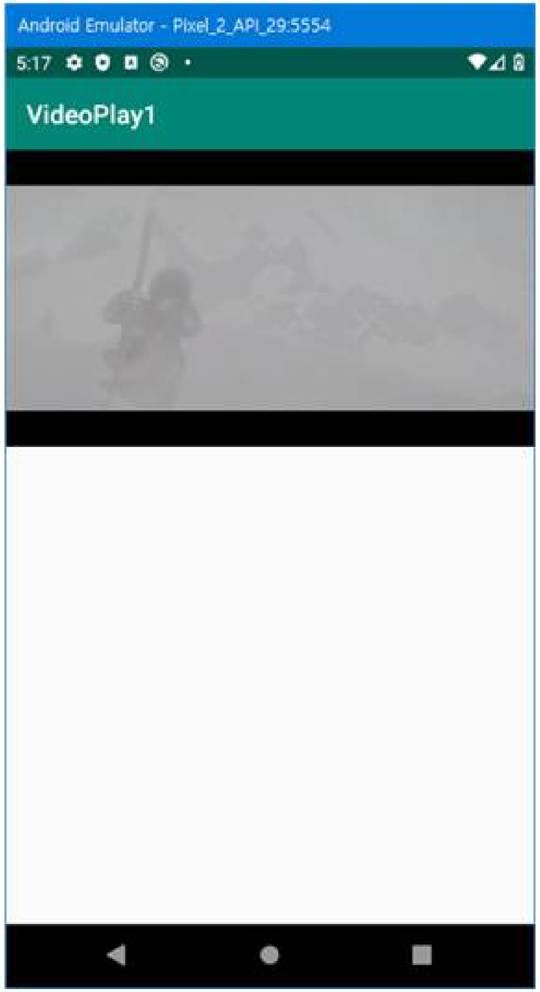
코드 작성

```java
public class MainActivity extends AppCompatActivity {
    VideoView videoview;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.setContentView(R.layout.activity_main);

        videoview = (VideoView) this.findViewById(R.id.videoview);
        MediaController mc = new MediaController(this);
        videoview.setMediaController(mc);
        videoview.setVideoURI(Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.trailer));
        videoview.start();
    }
}
```

매니페스트 파일

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.videoplay1">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

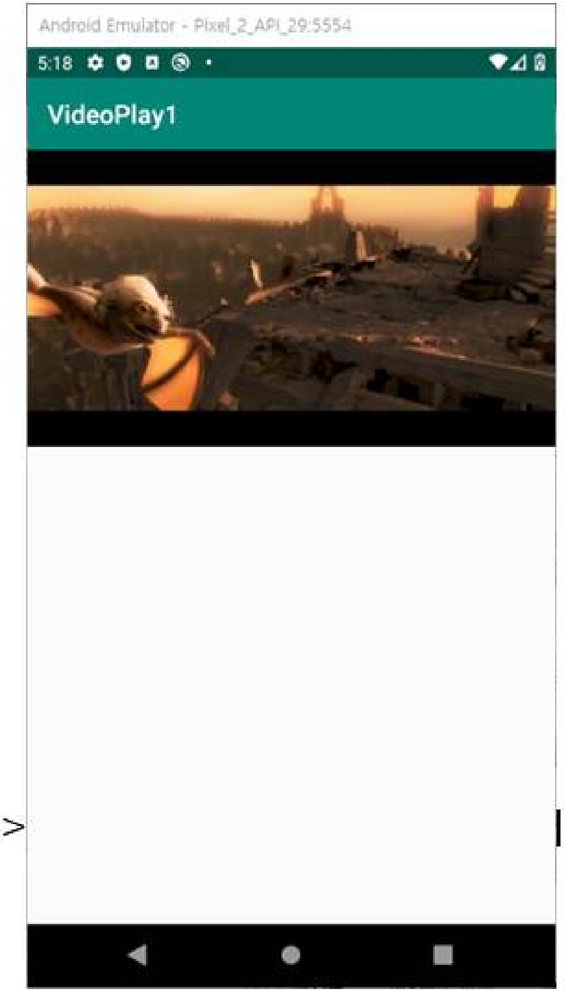
실행 결과


Android Emulator - Pixel_2_API_29:5554

5:17

VideoPlay1

->

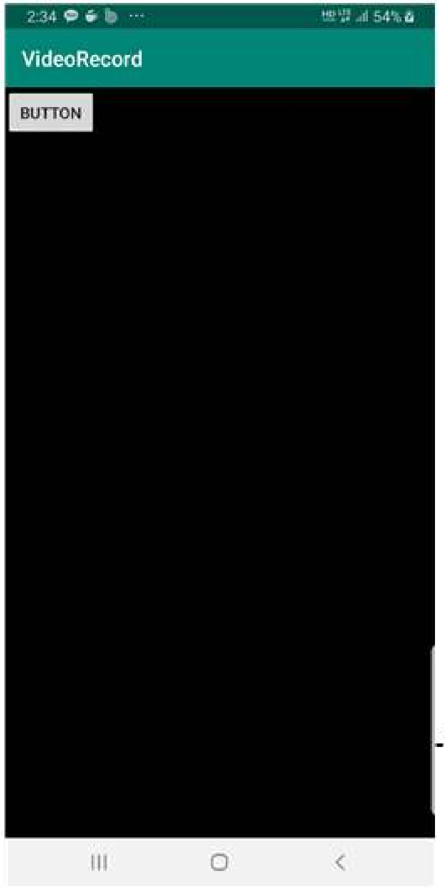

Android Emulator - Pixel_2_API_29:5554

5:18

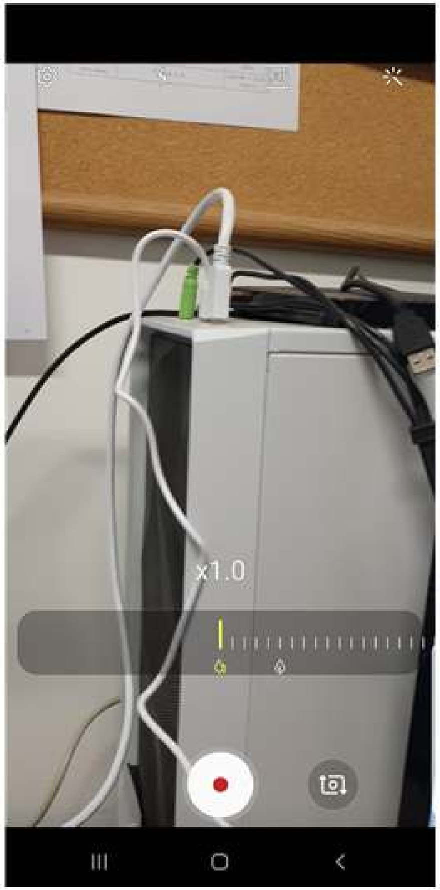
VideoPlay1

비디오 녹화

- □ 인텐트를 이용한 비디오 녹화



->



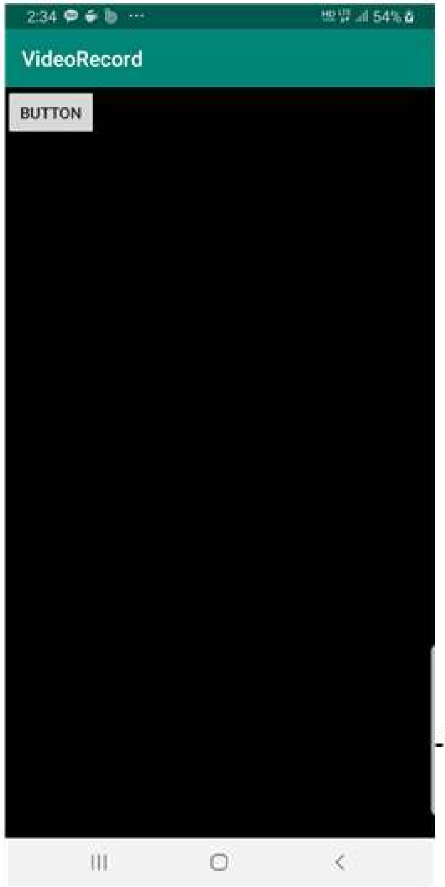
사용자 인터페이스 작성

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="takevideo"
        android:text="Button"
        tools:layout_editor_absoluteX="63dp"
        tools:layout_editor_absoluteY="650dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

코드 작성

```
public class MainActivity extends AppCompatActivity {
    VideoView videoView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        videoView = (VideoView) this.findViewById(R.id.videoView);
    }
    static final int REQUEST_VIDEO_CAPTURE = 1;

    public void takevideo(View v) {
        Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
        if (takeVideoIntent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(takeVideoIntent, REQUEST_VIDEO_CAPTURE);
        }
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
        super.onActivityResult(requestCode, resultCode, intent);
        if (requestCode == REQUEST_VIDEO_CAPTURE && resultCode == RESULT_OK) {
            Uri videoUri = intent.getData();
            videoView.setVideoURI(videoUri);
        }
    }
}
```

실행 결과



->