

CHAP 15. 멀티미디어

안드로이드의 멀티미디어 지원

2

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

안드로이드 지원 파일 형식

3

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)
이미지	PCM/WAVE		•	WAVE (.wav)
	JPEG	•	•	JPEG (.jpg)
	GIF		•	GIF (.gif)
	PNG	•	•	PNG (.png)
비디오	BMP		•	BMP (.bmp)
	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

4

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

(1) 인텐트를 사용하는 방법

5

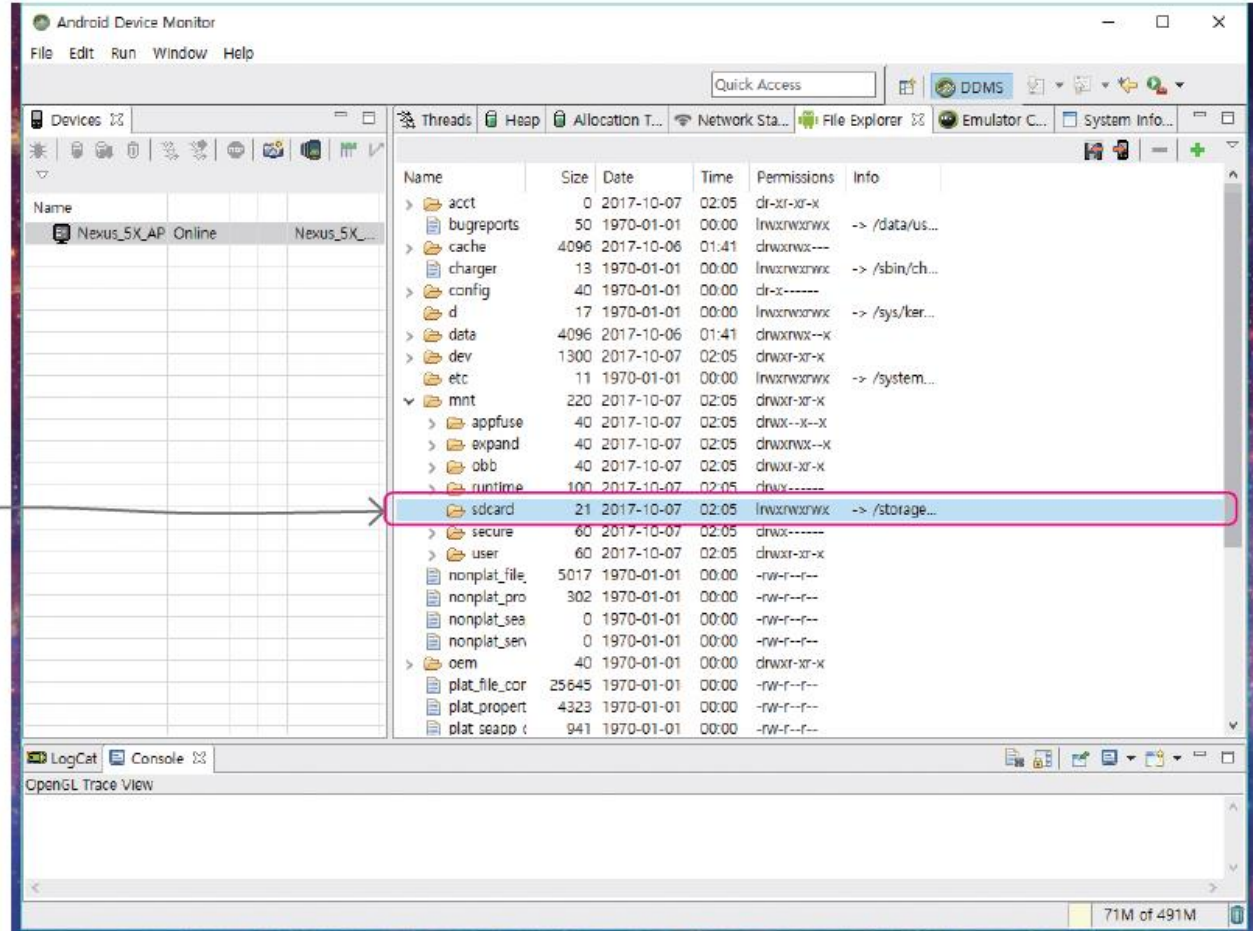
```
public class MainActivity extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button playButton = (Button) findViewById(R.id.play);
        playButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_VIEW);
                Uri uri = Uri.parse("file:///sdcard/old_pop.mp3");
                intent.setDataAndType(uri, "audio/mp3");
                startActivity(intent);
            }
        });
    }
}
```

음악 파일 복사

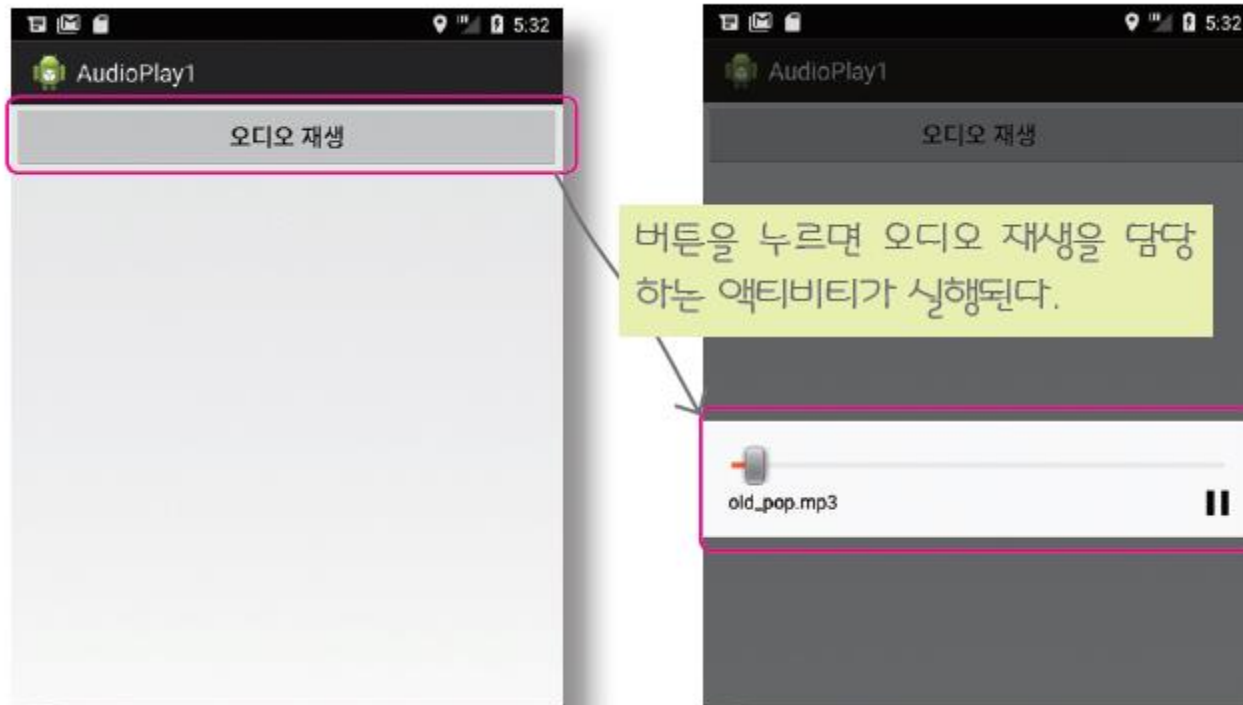
6

이 폴더로 외부 음악 파일을 복사한다.



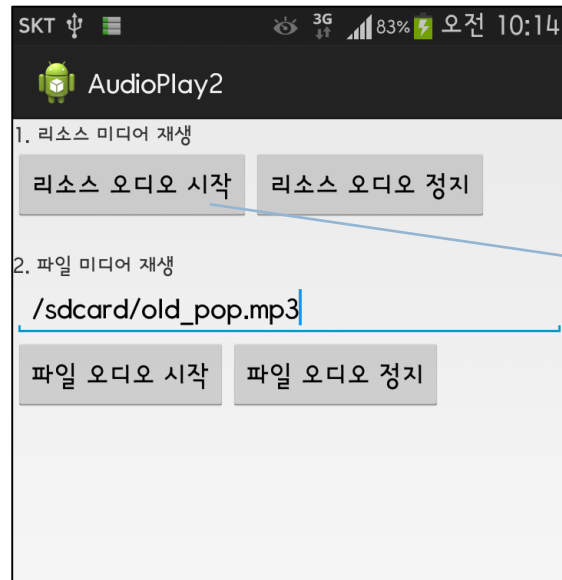
실행 결과

7



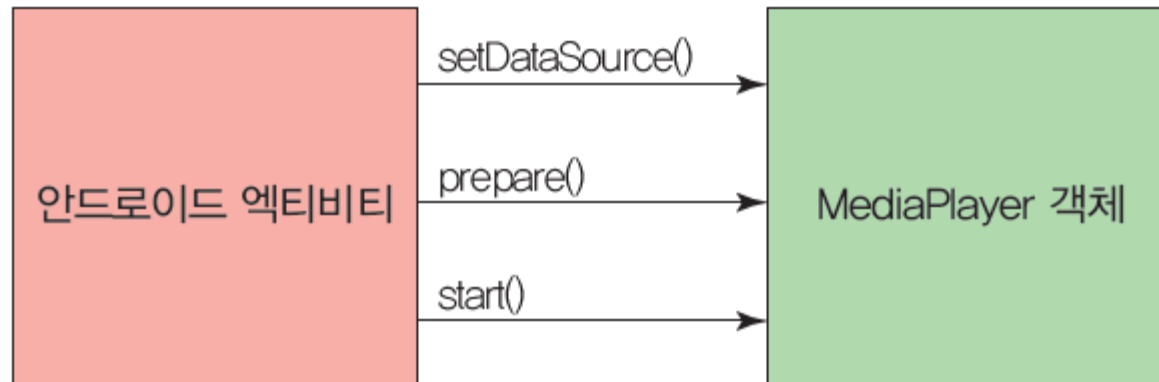
(2) MediaPlayer 클래스를 사용한 오디오 예제

8



MediaPlayer 클래스

9



코드 작성

10

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

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        edit = (EditText) findViewById(R.id.path);
    }

    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;
    }
}
```

```
public void startFileAudio(View v) {
    String file;
    file = edit.getText().toString();
    mp = new MediaPlayer();
    try {
        mp.setDataSource(file);
        mp.prepare();
    } catch (Exception e) {
        e.printStackTrace();
    }
    mp.start();
}

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

매니페스트 파일

12

AndroidManifest.xml

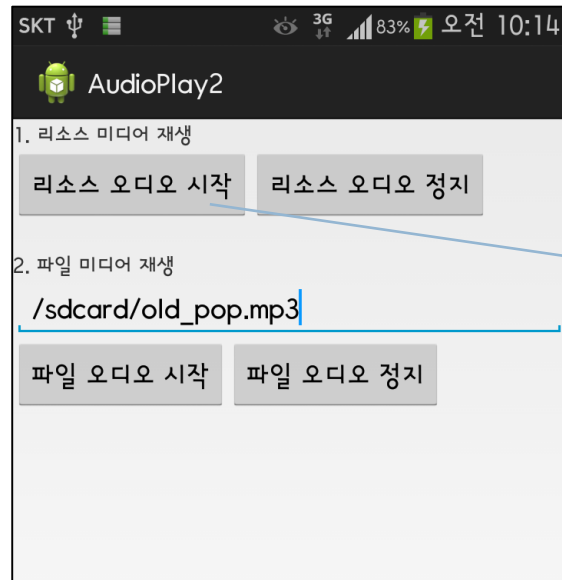
...

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE">  
</uses-permission>
```

...

실행 결과

13



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

MediaRecorder를 사용하여 오디오 녹음

15



필수적인 코드

오디오 소스를 설정한다.

```
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
```

출력 파일 형식을 설정한다.

```
recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
```

엔코더를 설정한다.

```
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
```

```
recorder.setOutputFile(PATH_NAME);
```

데이터가 저장될 파일경로를 생성한다.

```
recorder.prepare();
```

MediaRecorder 객체의 prepare()를 호출한다.

```
recorder.start();
```

녹음 시작

...

```
recorder.stop();
```

```
recorder.reset();
```

setAudioSource() 단계로 돌아가면 객체를 재사용할 수 있다.

```
recorder.release();
```

객체가 소멸된다. 재사용은 불가능하다.

예제

16

```
record.setOnClickListener(new Button.OnClickListener() {  
    public void onClick(View v) {  
        if (recorder == null) {  
            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();  
            record.setText("녹음 중지");  
        } else {  
            recorder.stop();  
            recorder.release();  
            recorder = null;  
            record.setText("녹음 시작");  
        }  
    }  
});  
}
```

MediaPlayer의 객체를 생성한다.

녹음 준비 단계에서는 예외가 발생할 수 있다.

녹음을 시작한다.

매니페스트 파일

17

AndroidManifest.xml

...

<uses-permission

android:name="android.permission.WRITE_EXTERNAL_STORAGE" ></uses-permission>

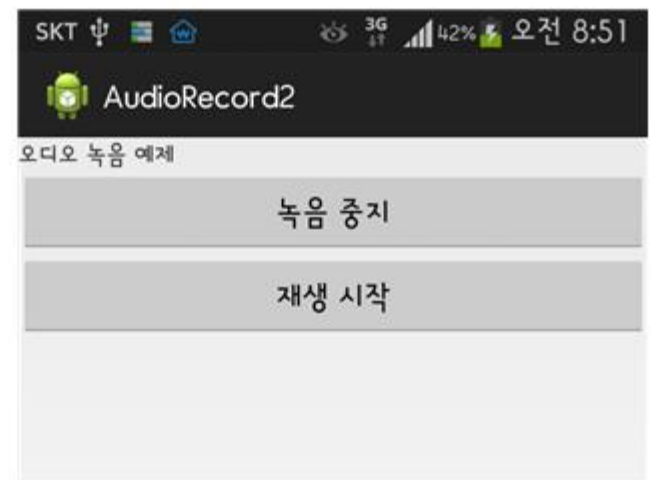
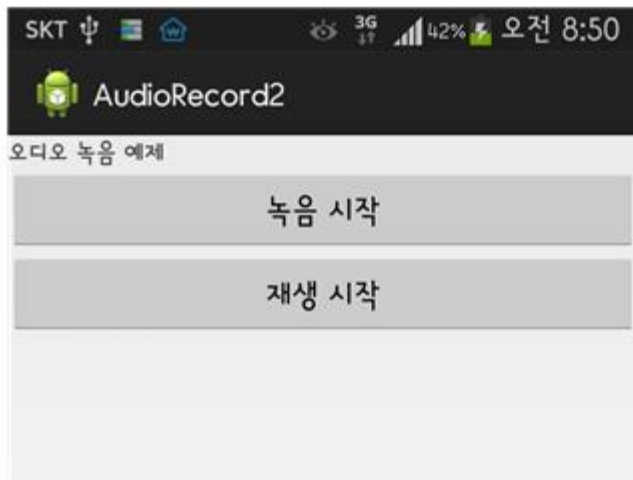
<uses-permission

android:name="android.permission.RECORD_AUDIO" ></uses-permission>

...

실행 결과

18



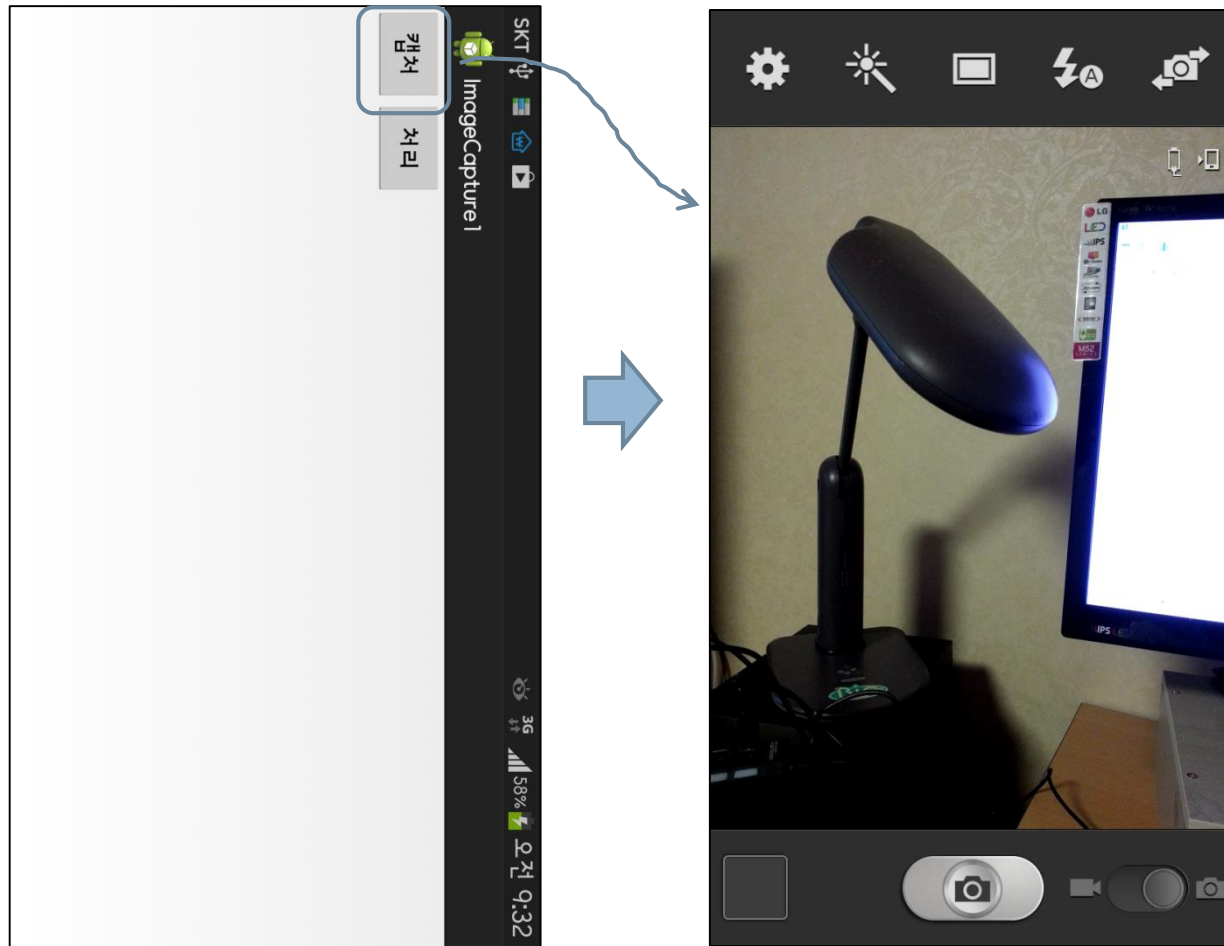
이미지 캡처

19

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

예제. 인텔리гент 이용한 이미지 획득

20



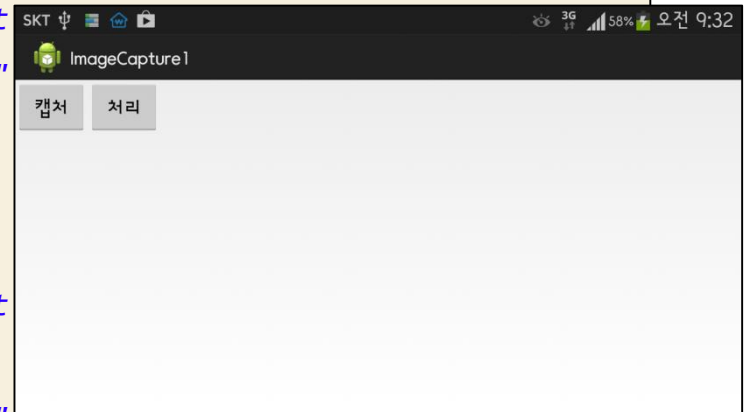
사용자 인터페이스 작성

21

```
<?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">
    <Button
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="캡처"
        android:id="@+id/capture" />

    <Button
        android:layout_height="wrap_content"
        android:text="처리"
        android:layout_width="wrap_content"
        android:id="@+id/process" />

    <ImageView
        android:id="@+id/imageview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```



```
public class MainActivity extends AppCompatActivity {
    private static final int CAMERA_CAPTURE = 0;

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

        Button captureButton = (Button) findViewById(R.id.capture);
        Button processButton = (Button) findViewById(R.id.process);
        captureButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Intent i = new Intent(
                    android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                i.putExtra(android.provider.MediaStore.EXTRA_OUTPUT,

                    Uri.fromFile(new File("/sdcard/image.jpg")));
                startActivityForResult(i, CAMERA_CAPTURE);
            }
        });
    }
}
```

```
processButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        Bitmap captureBmp = null;

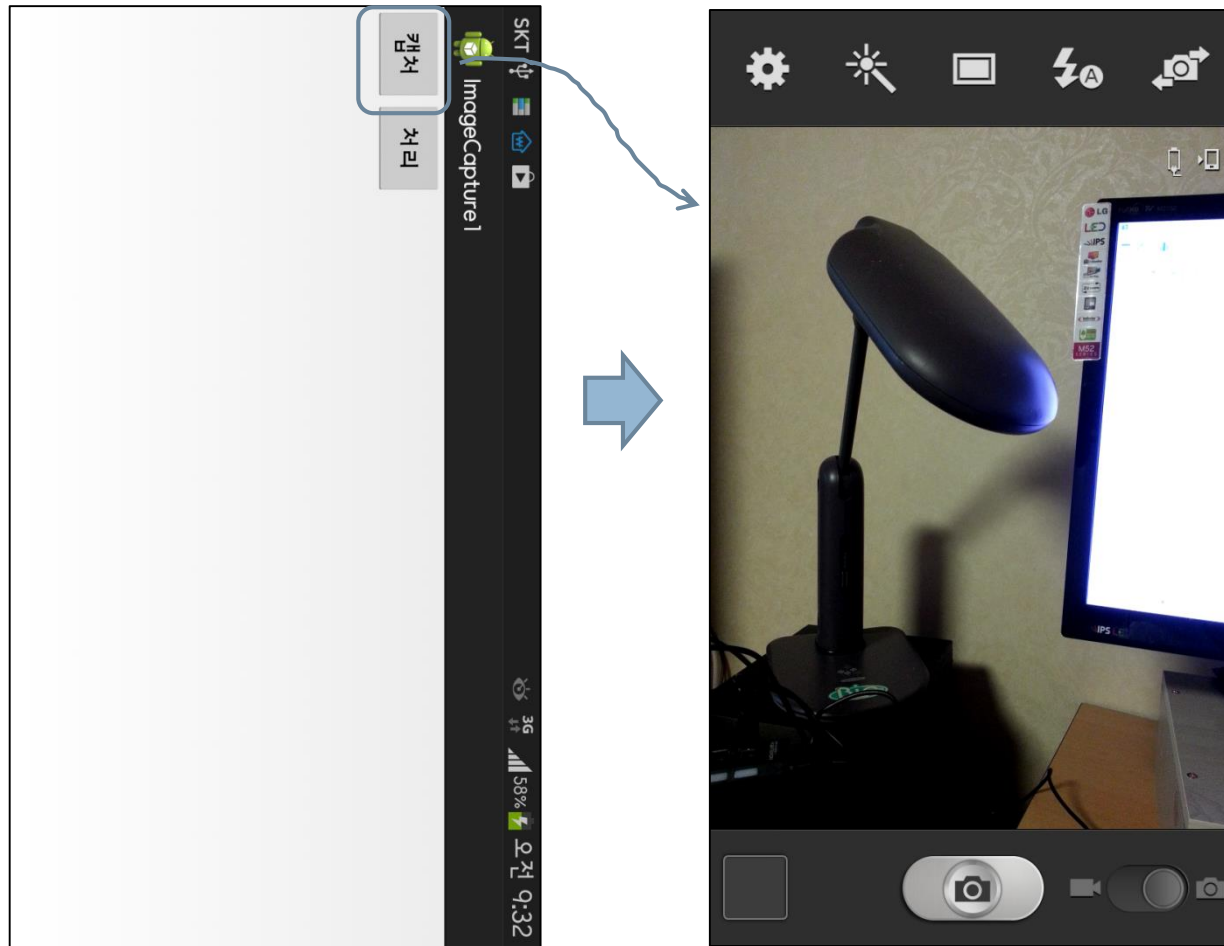
        File file = new File("/sdcard/image.jpg");
        try {
            captureBmp = Images.Media.getBitmap(getContentResolver(),
                Uri.fromFile(file));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        int width = captureBmp.getWidth();
        int height = captureBmp.getHeight();
        Bitmap tmpBmp = captureBmp.copy(Bitmap.Config.ARGB_8888, true);
        for (int y = 0; y < height; y++) {
            for (int x = 0; x < width; x++) {
                int value = captureBmp.getPixel(x, y);
                if (value < 0xff808080)
                    tmpBmp.setPixel(x, y, 0xff000000);
                else
                    tmpBmp.setPixel(x, y, 0xffffffff);
            }
        }
        ImageView imgView = (ImageView) findViewById(R.id.imageview);
        imgView.setImageBitmap(tmpBmp);
    }
});
}
```

@Override

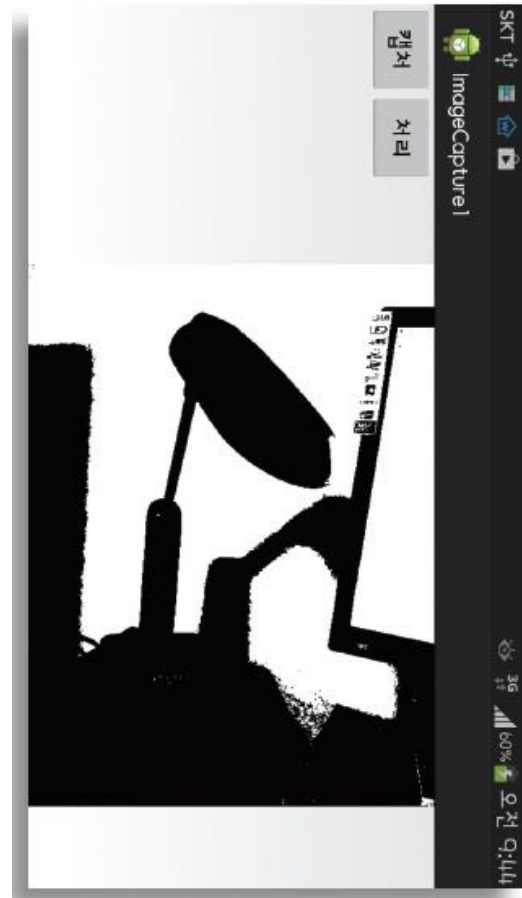
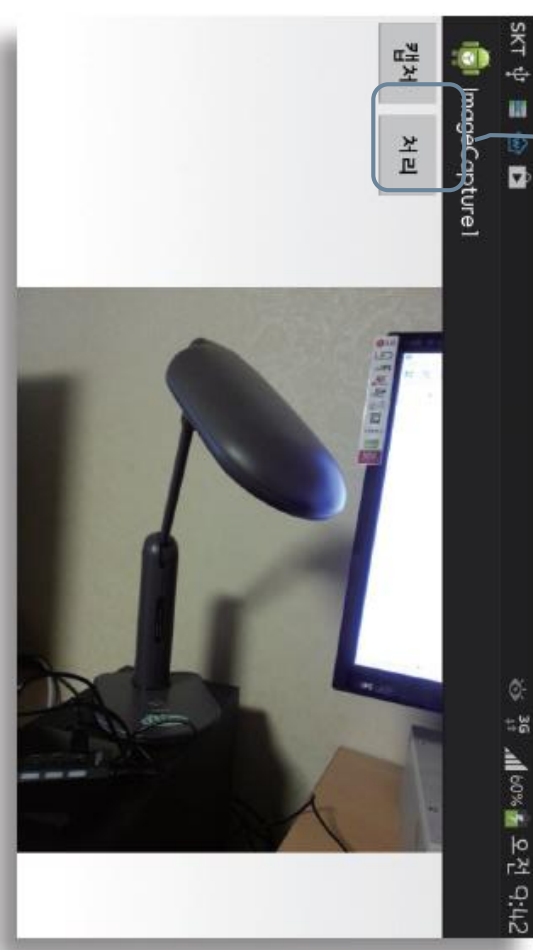
```
protected void onActivityResult(int requestCode, int resultCode,
    Intent intent) {
    Bitmap captureBmp = null;

    if (resultCode == RESULT_OK && requestCode == CAMERA_CAPTURE) {
        File file = new File("/sdcard/image.jpg");
        try {
            captureBmp = Images.Media.getBitmap(getContentResolver(),
                Uri.fromFile(file));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        ImageView imgView = (ImageView) findViewById(R.id.imageview);
        imgView.setImageBitmap(captureBmp);
    }
}
```

실행 결과

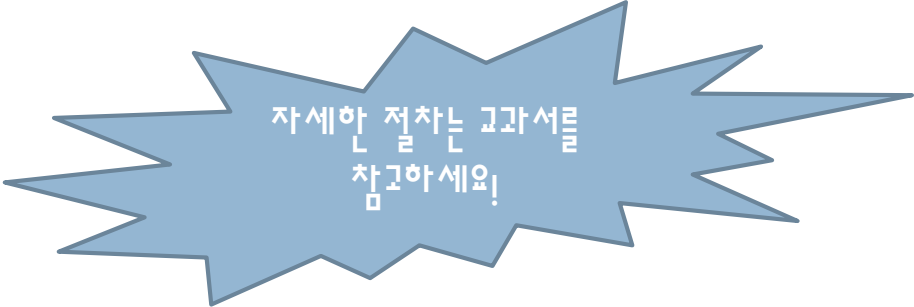
26



Camera 클래스를 이용한 영상 캡처

27

- 만약 안드로이드에서 기본적으로 제공하는 카메라 서비스를 이용하지 않고 자신만의 카메라 애플리케이션을 작성하고자 한다면 **Camera** 클래스를 이용하면 된다.
- **Camera** 클래스를 이용하면 이미지 캡처 설정 변경, 이미지 프리뷰, 비디오에서 프레임 추출 등을 할 수 있다.



자세한 절차는 교과서를
참고하세요!

처리단계

28

단계	설명
1	카메라 하드웨어 감지 - 카메라의 존재를 검사하고 카메라 객체에 대한 참조를 얻는다.
2	프리뷰 클래스 생성 - SurfaceView를 상속받는 카메라 프리뷰 클래스를 생성하고 SurfaceHolder 인터페이스를 구현한다. 이 클래스는 카메라로부터 오는 실시간 영상을 보여준다. 만약 영상을 미리 보는 기능이 필요 없으면 구현하지 않아도 된다.
3	레이아웃을 작성한다 - 프리뷰를 추가할 수 있는 프레임 레이아웃과 원하는 UI 컨트롤들을 포함 시킨 레이아웃을 작성한다.
4	캡처를 담당하는 리스너를 설정한다 - 이미지 캡처를 처리하는 리스너를 UI 컨트롤에 연결한다.
5	캡처 및 저장 - 사진이나 비디오가 캡처되면 데이터를 파일로 저장한다.
6	카메라를 해제한다 - 카메라를 사용한 후에는 애플리케이션에서 다른 애플리케이션이 사용할 수 있도록 카메라를 해제하여야 한다.

사용자 인터페이스 작성

29



사용자
인터페이스작성



activity_main.xml

```
<?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" >
```

```
<FrameLayout
    android:id="@+id/camera_preview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1" />
```

여기에 프리뷰가 추가된다.

```
public class MainActivity extends AppCompatActivity {
    private Camera mCamera;
    private CameraPreview mPreview;

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

        mCamera = getCameraInstance();

        mPreview = new CameraPreview(this, mCamera);
        FrameLayout preview = (FrameLayout) findViewById(R.id.camera_preview);
        preview.addView(mPreview);

        Button captureButton = (Button) findViewById(R.id.button_capture);
        captureButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mCamera.takePicture(null, null, mPicture);
            }
        });
    }
}
```

```

private PictureCallback mPicture = new PictureCallback() {

    @Override
    public void onPictureTaken(byte[] data, Camera camera) {

        File pictureFileDir = getDir();
        if (!pictureFileDir.exists() && !pictureFileDir.mkdirs()) {
            Log.d("Camera", "Can't create directory.");
            Toast.makeText(getApplicationContext(),
                "Can't create directory.", Toast.LENGTH_LONG).show();
            return;
        }

        SimpleDateFormat dateFormat = new
SimpleDateFormat("yyyymmddhhmmss");
        String date = dateFormat.format(new Date());
        String photoFile = "MyPic" + date + ".jpg";

        String filename = pictureFileDir.getPath() + File.separator
            + photoFile;

        File pictureFile = new File(filename);

        try {
            FileOutputStream fos = new FileOutputStream(pictureFile);
            fos.write(data);
            fos.close();
            Toast.makeText(getApplicationContext(),
                "New Image is saved:" + photoFile, Toast.LENGTH_LONG)
                .show();
        } catch (Exception error) {
            Log.d("Camera", "File" + filename + "Image was not saved: "
                + error.getMessage());
            Toast.makeText(getApplicationContext(), "Image was not saved.",
                Toast.LENGTH_LONG).show();
        }
    }
}

```

```
    }  
}  
  
private File getDir() {  
    File sdDir = Environment  
.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);  
    return new File(sdDir, "CameraCapture");  
}  
};  
  
public static Camera getCameraInstance() {  
    Camera c = null;  
    try {  
        c = Camera.open(); // Camera 객체의 인스턴스를 얻는다.  
    } catch (Exception e) { // 카메라를 사용할 수 없음  
    }  
    return c;  
}  
}
```



```

public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback {
    private SurfaceHolder mHolder;
    private Camera mCamera;

    public CameraPreview(Context context, Camera camera) {
        super(context);
        mCamera = camera;

        mHolder = getHolder();
        mHolder.addCallback(this);
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
    }

    public void surfaceCreated(SurfaceHolder holder) {
        try {
            mCamera.setPreviewDisplay(holder);
            mCamera.setDisplayOrientation(90);
            mCamera.startPreview();
        } catch (IOException e) {
            Log.d("Camera", "Error setting camera preview: " + e.getMessage());
        }
    }

    public void surfaceDestroyed(SurfaceHolder holder) {
    }
}

```

매니페스트 파일

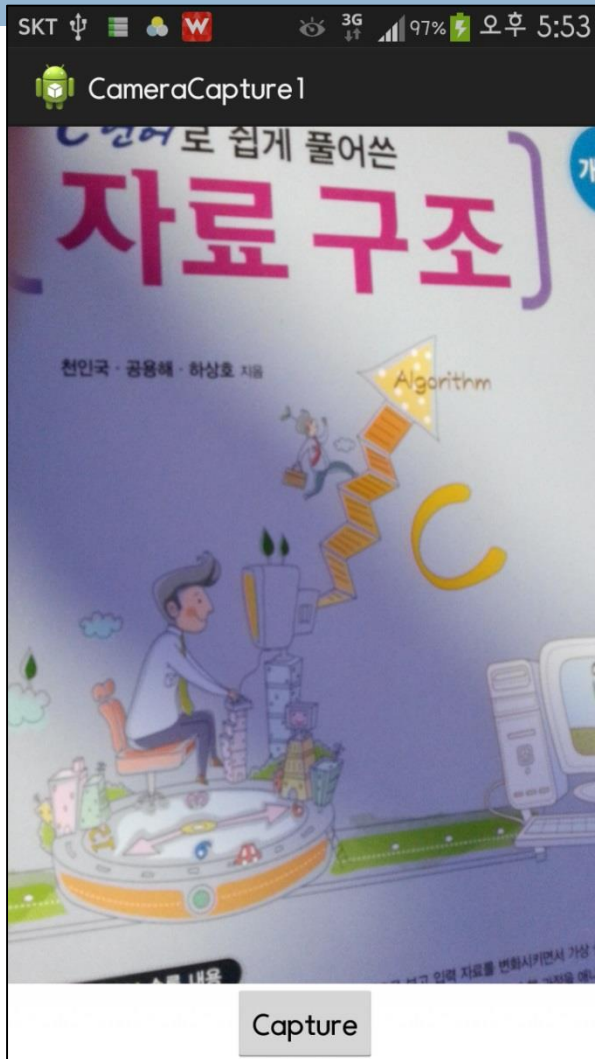
34

AndroidManifest.xml

```
...  
  
    <uses-permission android:name="android.permission.CAMERA" />  
    <uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
  
    <uses-feature android:name="android.hardware.camera" />  
    <uses-feature android:name="android.hardware.camera.autofocus" />  
  
...  
  
    <application  
        android:allowBackup="true"  
        android:icon="@drawable/ic_launcher"  
        android:label="@string/app_name"  
        android:screenOrientation="landscape"  
        android:theme="@style/AppTheme" >  
  
...
```

실행 결과

35



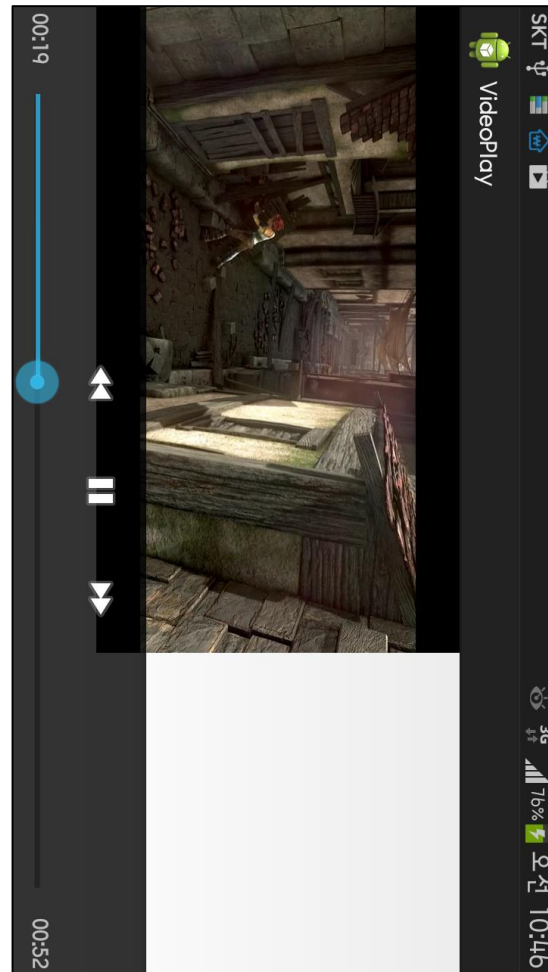
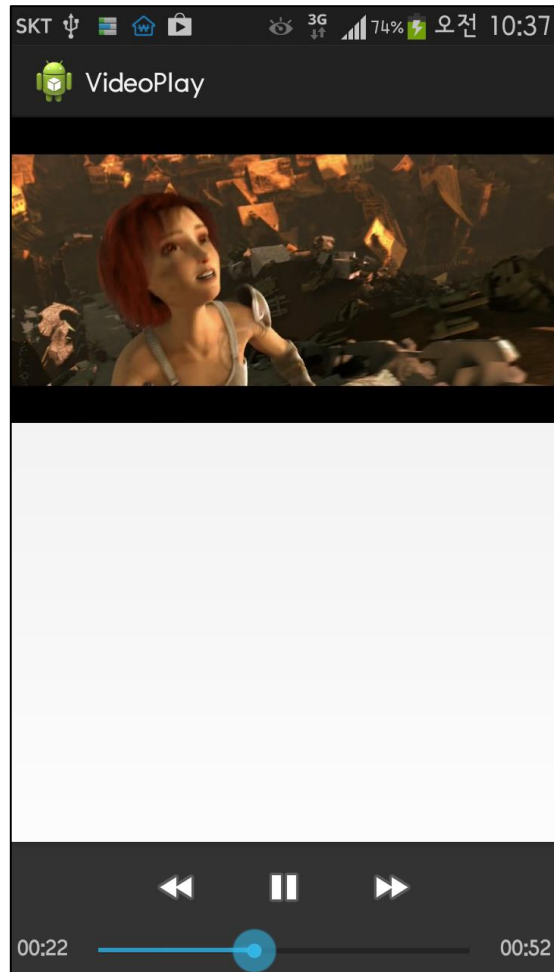
비디오 재생

36

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

예제: 비디오 재생

37



소프트웨어학부

사용자 인터페이스 작성

38

```
<?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:layout_height="match_parent"
    >
<VideoView android:id="@+id/videoview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
</LinearLayout>
```

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

        VideoView videoview = (VideoView) this.findViewById(R.id.videoview);
        MediaController mc = new MediaController(this);
        videoview.setMediaController(mc);
        String folder = Environment.getExternalStorageDirectory().getAbsolutePath();

        videoview.setVideoPath(folder + "/trailer.mp4");
        videoview.requestFocus();
        videoview.start();
    }
}
```

매니페스트 파일

40

AndroidManifest.xml

...

<uses-permission

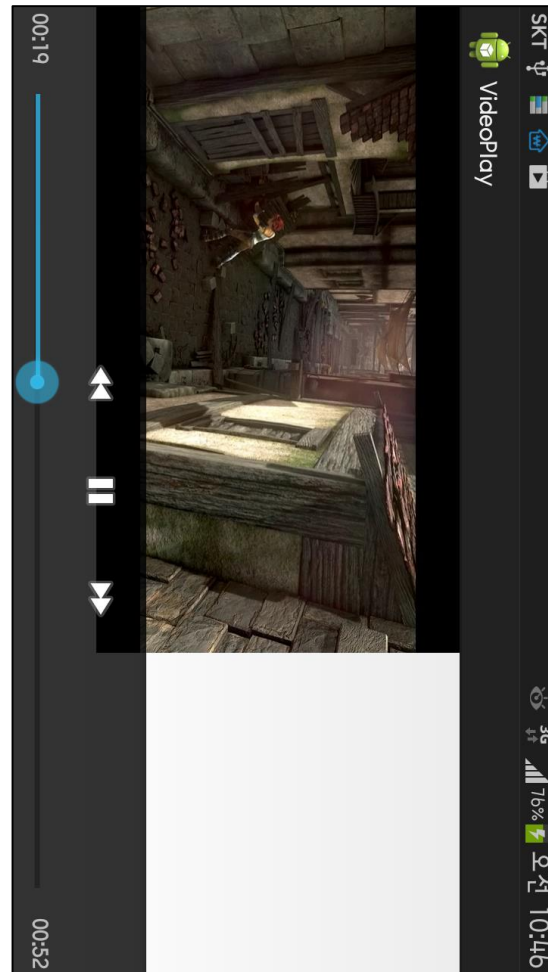
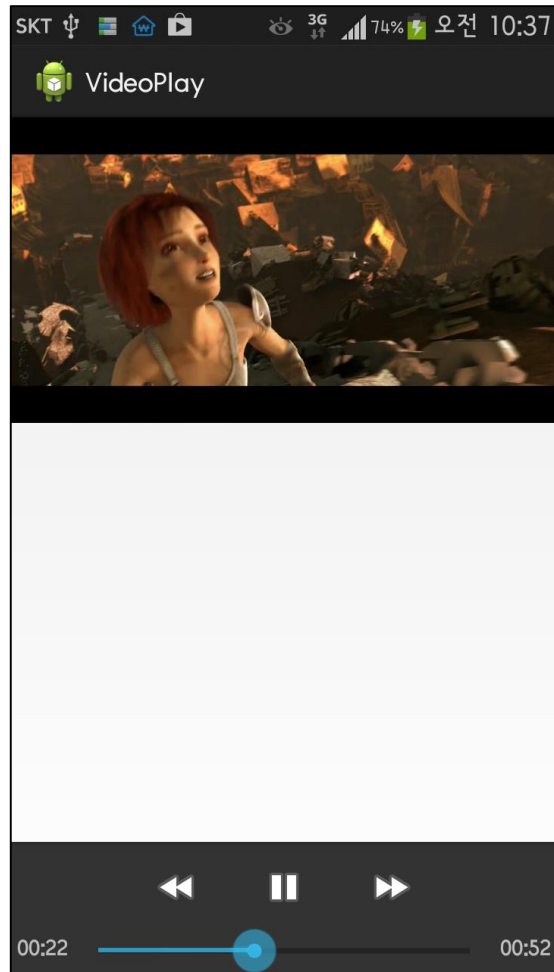
 android:name="android.permission.READ_EXTERNAL_STORAGE">

</uses-permission>

...

실행 결과

41

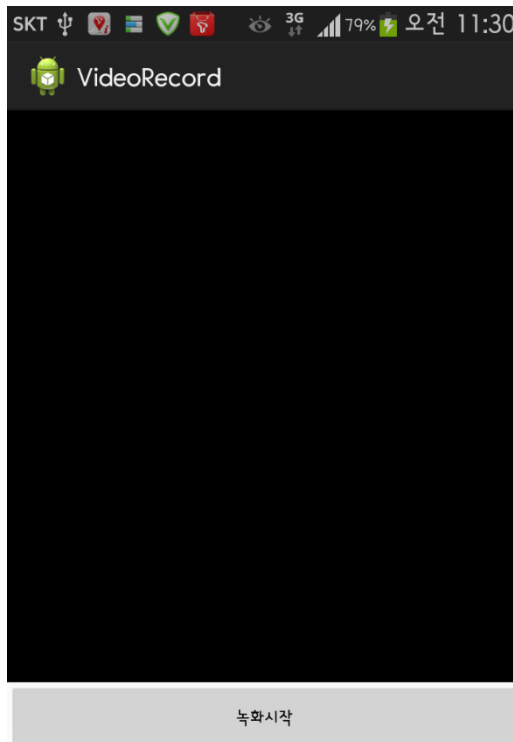


소프트웨어학부

비디오 녹화

42

- MediaRecorder 클래스: 비디오 녹화 담당
- SurfaceView : 캠코더의 화면을 표시



MediaPlayer 클래스

43

메소드	설명
<code>setVideoSource()</code>	이 메소드는 녹화에 사용되는 비디오 소스를 선택한다. 즉 Camera나 아니면 Default 중에서 선택한다. 만약 이 메소드가 호출되지 않으면 출력 파일은 비디오 트랙을 포함하지 않는다.
<code>setOutputFormat()</code>	이 메소드는 녹화 때 생성되는 출력 파일의 형식을 설정한다. 이 메소드는 <code>setAudioSource()/setVideoSource()</code> 메소드 이후에 호출해야 한다. 그러나 <code>prepare()</code> 메소드 전에 반드시 호출되어야 한다.
<code>setVideoEncode()</code>	이 메소드는 녹화 때 사용되는 비디오 엔코더를 설정한다. 만약 이 메소드가 호출되지 않으면 출력 파일은 비디오 트랙을 포함하지 않는다. 이 메소드는 <code>setOutputFormat()</code> 후에 그러나 <code>prepare()</code> 전에 호출되어야 한다.
<code>setOutputFile()</code>	이 메소드는 출력 파일의 경로를 설정한다. 이 메소드는 <code>setOutputFormat()</code> 후에 그러나 <code>prepare()</code> 전에 호출되어야 한다.
<code>setPreviewDisplay()</code>	녹화되는 비디오의 프리뷰를 보이기 위한 서피스를 설정한다. 이 메소드는 원하는 프리뷰 디스플레이를 설정하기 위하여 <code>prepare()</code> 전에 호출되어야 한다.
<code>prepare()</code>	비디오를 캡처하고 엔코딩하기 위하여 레코더를 준비한다. 이 메소드는 원하는 오디오나 비디오 소스나 엔코더, 파일 형식 등을 설정한 후에 호출되어야 한다. 그러나 <code>start()</code> 전에는 반드시 호출되어야 한다.
<code>start()</code>	비디오를 캡처하고 엔코딩하여서 파일에 저장한다. <code>setOutputFile()</code> 로 지정된 파일에 저장한다. <code>prepare()</code> 후에 이 메소드를 호출한다.

사용자 인터페이스 작성

44

```
<?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" >
```

```
    <SurfaceView  
        android:id="@+id/videoview"  
        android:layout_width="match_parent"  
        android:layout_height="400dp" />
```

```
    <Button  
        android:id="@+id/mybutton"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="녹화시작"  
        android:textSize="12dp" />
```

```
</LinearLayout>
```



코드 작성

45

```
public class MainActivity extends AppCompatActivity implements SurfaceHolder.Callback {

    Button myButton;
    MediaRecorder mediaRecorder;
    SurfaceHolder surfaceHolder;
    boolean is_recording;

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

        is_recording = false;

        mediaRecorder = new MediaRecorder();
        initMediaRecorder();

        setContentView(R.layout.main);

        SurfaceView myVideoView = (SurfaceView) findViewById(R.id.videoview);
        surfaceHolder = myVideoView.getHolder();
        surfaceHolder.addCallback(this);

        myButton = (Button) findViewById(R.id.mybutton);
        myButton.setOnClickListener(myButtonOnClickListener);
    }
}
```

코드 작성

46

```
private Button.OnClickListener myButtonOnClickListener = new Button.OnClickListener() {

    public void onClick(View arg0) {
        if (is_recording) {
            mediaRecorder.stop();
            mediaRecorder.release();
            finish();
        } else {
            mediaRecorder.start();
            is_recording = true;
            myButton.setText("녹화중지");
        }
    }
};

public void surfaceChanged(SurfaceHolder arg0, int arg1, int arg2, int arg3) {
}

public void surfaceCreated(SurfaceHolder arg0) {
    prepareMediaRecorder();
}

public void surfaceDestroyed(SurfaceHolder arg0) {
}
```

코드 작성

47

```
private void initMediaRecorder() {  
    mediaRecorder.setAudioSource(MediaRecorder.AudioSource.DEFAULT);  
    mediaRecorder.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    CamcorderProfile camcorderProfile_HQ = CamcorderProfile  
        .get(CamcorderProfile.QUALITY_LOW);  
    mediaRecorder.setProfile(camcorderProfile_HQ);  
    mediaRecorder.setOutputFile("/sdcard/video.mp4");  
    mediaRecorder.setMaxDuration(60000); // 최대 시간을 60초로 한정한다.  
    mediaRecorder.setMaxFileSize(5000000); // 최대 파일 크기를 5MB로 한정한다.  
}
```

```
private void prepareMediaRecorder() {  
    mediaRecorder.setPreviewDisplay(surfaceHolder.getSurface());  
    try {  
        mediaRecorder.prepare();  
    } catch (IllegalStateException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

실행 결과

48

