



30장. 멀티미디어 활용

30.1 오디오, 비디오 재생

30.1.1. 오디오 재생

- AAC LC, HE-AACv1, HE-AACv2 , AAC ELD, AMR-NB, AMR-WB, FLAC, MIDI, MP3, Opus, PCM/WAVE, Vorbis로 대부분 음원 파일 지원
- MediaPlayer를 이용하는 방법과 SoundPool을 이용
- MediaPlayer 이용
- MediaPlayer는 음원뿐 아니라 영상 재생에도 사용

```
mediaPlayer=MediaPlayer.create(this, R.raw.sound);  
mediaPlayer.start();
```

```
mediaPlayer = new MediaPlayer();  
mediaPlayer.setDataSource(this, Uri.parse(url));
```

```
mediaPlayer.setDataSource(filePath);
```



30.1 오디오, 비디오 재생

음원을 로딩하려면 `prepare()`와 `prepareAsync()` 두 함수를 사용

MediaPlayer에 등록하는 이벤트

- `OnCompletionListener`: 미디어 재생 완료 시 이벤트
- `OnErrorListener`: 재생 에러 발생
- `OnPreparedListener`: 재생 가능한 상태로 로딩된 상태
- `OnBufferingUpdateListener`: 버퍼링 이벤트
- `OnVideoSizeChangedListener`: 비디오 크기 변경 이벤트

MediaPlayer 함수

`start()`: 재생 시작

- `stop()`: 정지
- `prepare()`: 준비
- `pause()`: 일시 정지
- `release()`: 메모리 해제



30.1 오디오, 비디오 재생

- seekTo(): 재생 위치 지정
- getCurrentPosition(): 재생 위치
- getDuration(): 재생 시간
- getVideoHeight(): 영상 높이값
- getVideoWidth(): 영상 너비값
- setLooping(): 반복설정
- setVolume(): 볼륨 설정

SoundPool 이용

효과음을 여러 개 준비하고 상황에 따라 동시에 재생

```
SoundPool soundPool=null;

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    soundPool = new SoundPool.Builder()
        .setMaxStreams(10)
        .build();
} else {
    soundPool = new SoundPool(10, AudioManager.STREAM_MUSIC, 1);
}
```



30.1 오디오, 비디오 재생

SoundPool에서 load() 함수를 이용하여 로딩
재생 가능한 상태의 이벤트를 위해 OnLoadCompleteListener를 제공

```
soundPool.load(this, R.raw.sound, 1);

soundPool.setOnLoadCompleteListener(new SoundPool.OnLoadCompleteListener() {
    @Override
    public void onLoadComplete(SoundPool soundPool, int sampleId, int status) {
        soundPool.play(sampleId, 1f, 1f, 0, 0, 1f);
    }
});
```

음원 재생

play(int soundID, float leftVolume, float rightVolume, int priority, int loop, float rate)



30.1 오디오, 비디오 재생

30.1.2. 비디오 재생

- 지원하는 영상 파일 포맷은 H.263, H.264 AVC, H.265 HEVC, MPEG-4 SP, VP8, VP9을 지원
- MediaPlayer, VideoView를 이용

MediaPlayer

영상은 SurfaceView 필요

```
<SurfaceView
    android:id="@+id/lab1_surface"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"/>
```



30.1 오디오, 비디오 재생

```
public class MainActivity extends AppCompatActivity implements SurfaceHolder.Callback, View.OnClickListener {
    private MediaPlayer mediaPlayer;
    private SurfaceHolder holder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        holder = surfaceView.getHolder();
        holder.addCallback(this);
    }
    @Override
    public void onClick(View v) {          /.....
        mediaPlayer = MediaPlayer.create(this, R.raw.video);
        mediaPlayer.setDisplay(holder);
        mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        mediaPlayer.start();
    }
    public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {
    }
    public void surfaceCreated(SurfaceHolder holder) {
    }
    public void surfaceDestroyed(SurfaceHolder holder) {
    }
}
```



30.1 오디오, 비디오 재생

- onVideoSizeChanged() 함수의 매개변수로 영상을 출력하기 위한 크기 값을 획득

```
public void onVideoSizeChanged(MediaPlayer mp, int width, int height) {  
    holder.setFixedSize(width, height);  
    mediaPlayer.start();  
}
```

- VideoView
- SurfaceView를 준비하지 않아도 되며, 기본 컨트롤바를 제공

```
<VideoView  
    android:id="@+id/lab1_video"  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"/>
```

```
MediaController mediaController = new MediaController(this);
```

```
Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.video);  
videoView.setVideoURI(uri);  
videoView.setMediaController(mediaController);  
videoView.requestFocus();  
videoView.start();
```

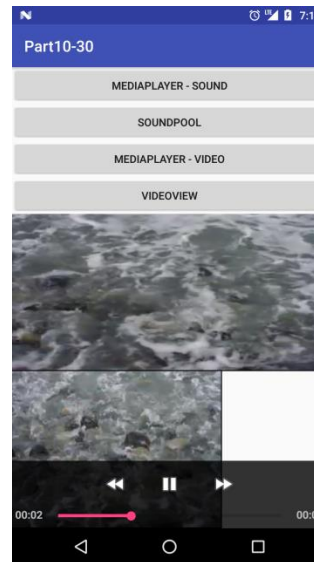


Step by Step 30-1 – 음원 및 영상 출력

음원 및 영상 출력을 테스트

버튼 클릭시 다양한 방법으로 음원 및 영상을 출력하는 테스트
음원 및 영상 파일은 리소스에 등록하여 테스트

1. Module 생성
2. 파일 복사
3. MainActivity.java 작성
4. 실행



30.2 음성, 영상 녹화

30.2.1. 음성 녹음

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

- 음성 녹음이나 영상 녹화 기능을 제공하는 클래스는 `MediaRecorder`

```
mediaRecorder = new MediaRecorder();
mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
mediaRecorder.setOutputFile(file.getAbsolutePath());
mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
```

- `prepare()`와 `start()` 함수를 호출하면 녹음이 시작

```
mediaRecorder.prepare();
mediaRecorder.start();
```

- 녹음을 종료하고 싶다면 `stop()` 함수를 호출

```
mediaRecorder.stop();
mediaRecorder.release();
mediaRecorder = null;
```



30.2 음성, 영상 녹화

30.2.2. 영상 녹화

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

- MediaRecorder가 카메라로부터 넘어오는 영상을 녹화

```
mediaRecorder = new MediaRecorder();  
camera.unlock();  
  
mediaRecorder.setCamera(camera);  
mediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);  
mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);  
  
mediaRecorder.setProfile(CamcorderProfile  
    .get(CamcorderProfile.QUALITY_HIGH));  
mediaRecorder.setOutputFile(file.getAbsolutePath());mediaRecorder.setOrientationHint(result);  
mediaRecorder.prepare();  
mediaRecorder.start();
```



Step by Step 30-2 – 영상 녹화

영상 녹화를 테스트

1. Activity 생성
2. 파일 복사
3. styles.xml 작성
4. AndroidManifest.xml 작업
5. Lab30_2Activity.java 작성
6. Lab30_2Activity.java 실행

