

Week07.
비디오 재생



개발환경 구축 절차

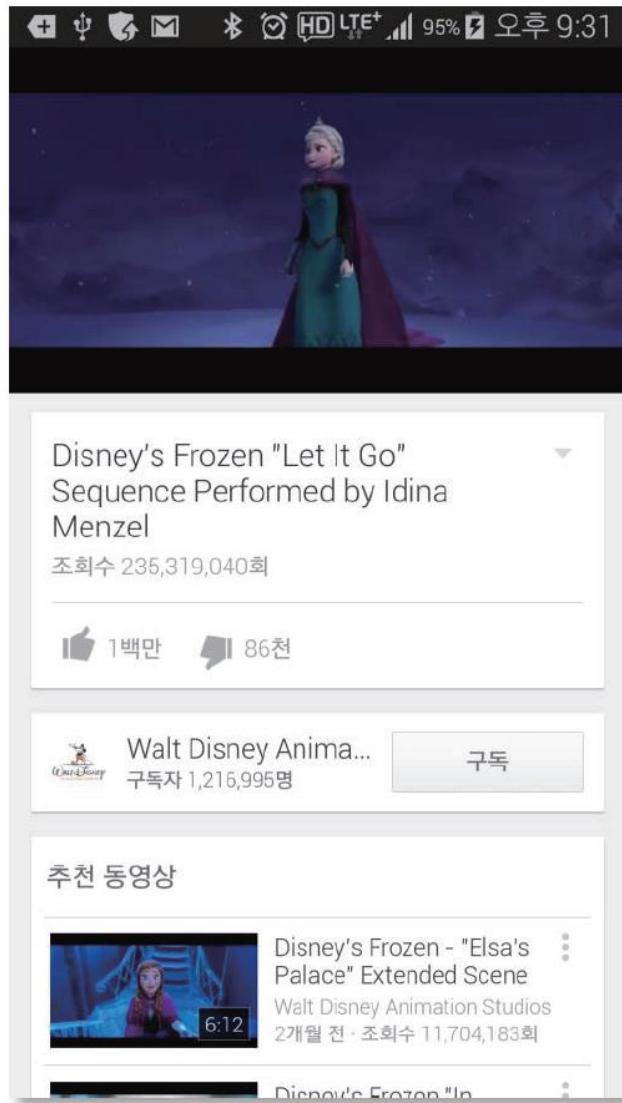
2

주 차	수업 내용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	이미지의 출력
5	이벤트 처리와 액티비티 간 이동
6	오디오 재생
7	비디오 재생
8	중간고사
9	애니메이션
10	사물인터넷과 센서 – 터치 센서, 모션 센서
11	사물인터넷과 센서 – 위치 센서, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활용
14	구글 맵과 위치 추적
15	기말 고사

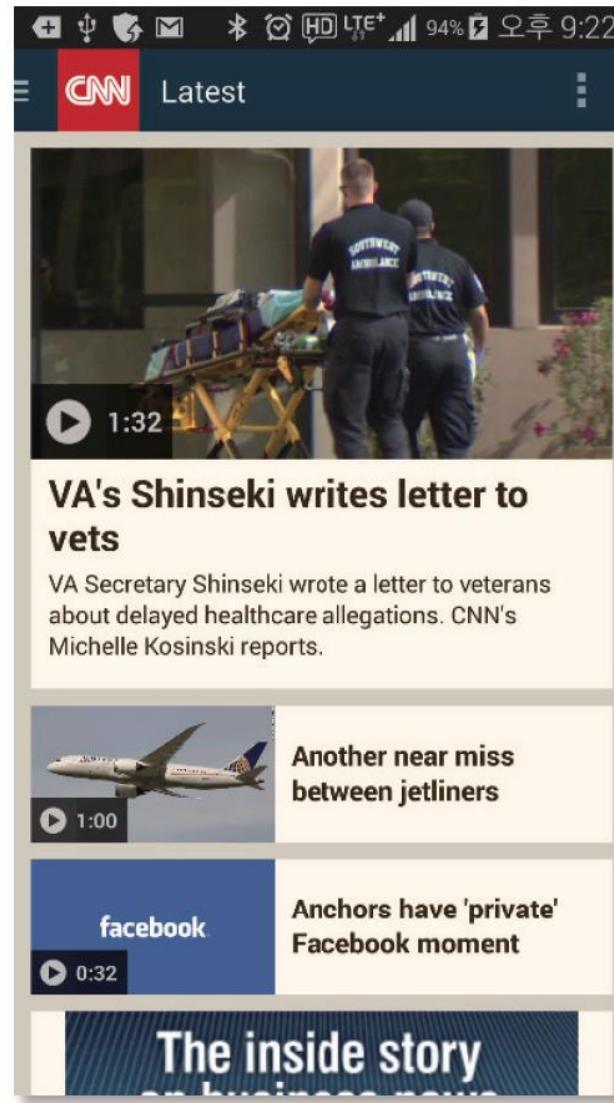


<http://github.com/hopypark>

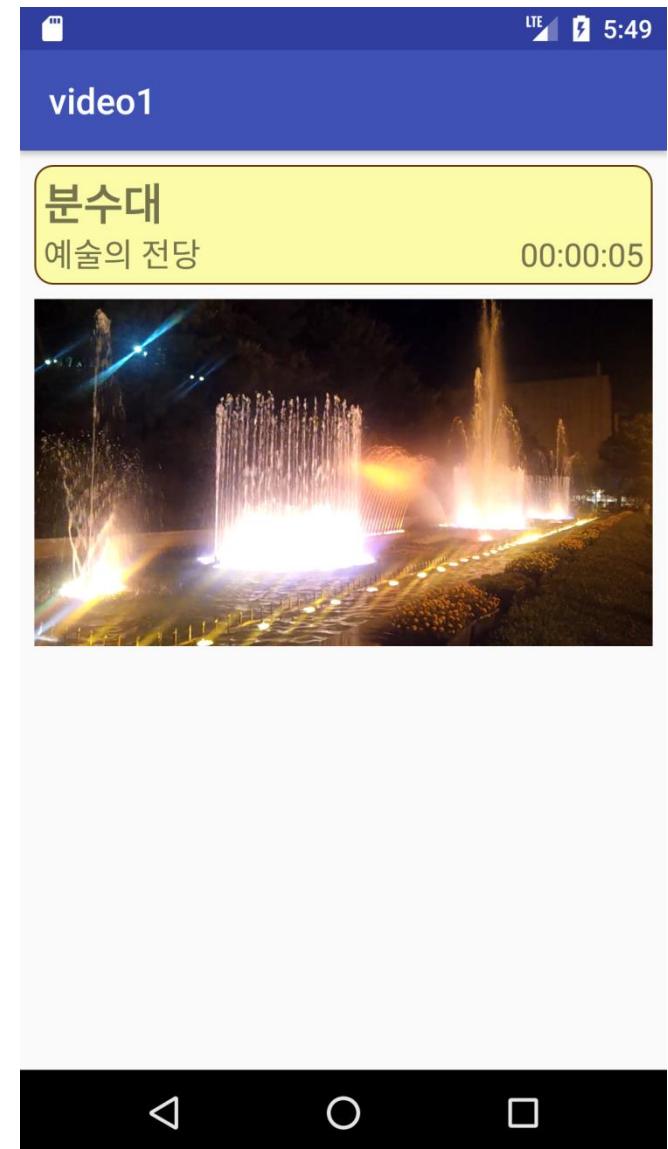
비디오 재생 앱의 예



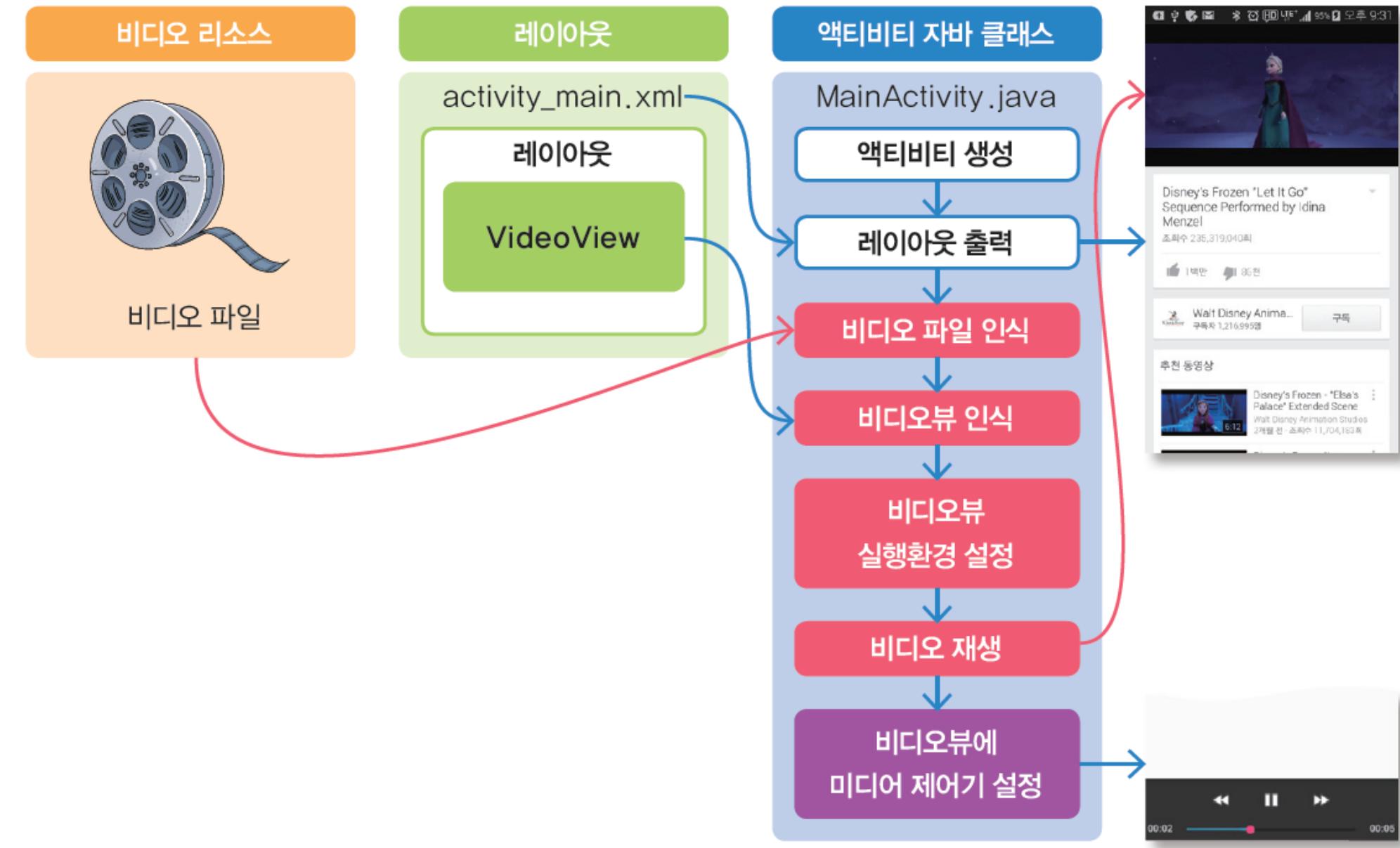
(a) YouTube 동영상



(b) CNN Video

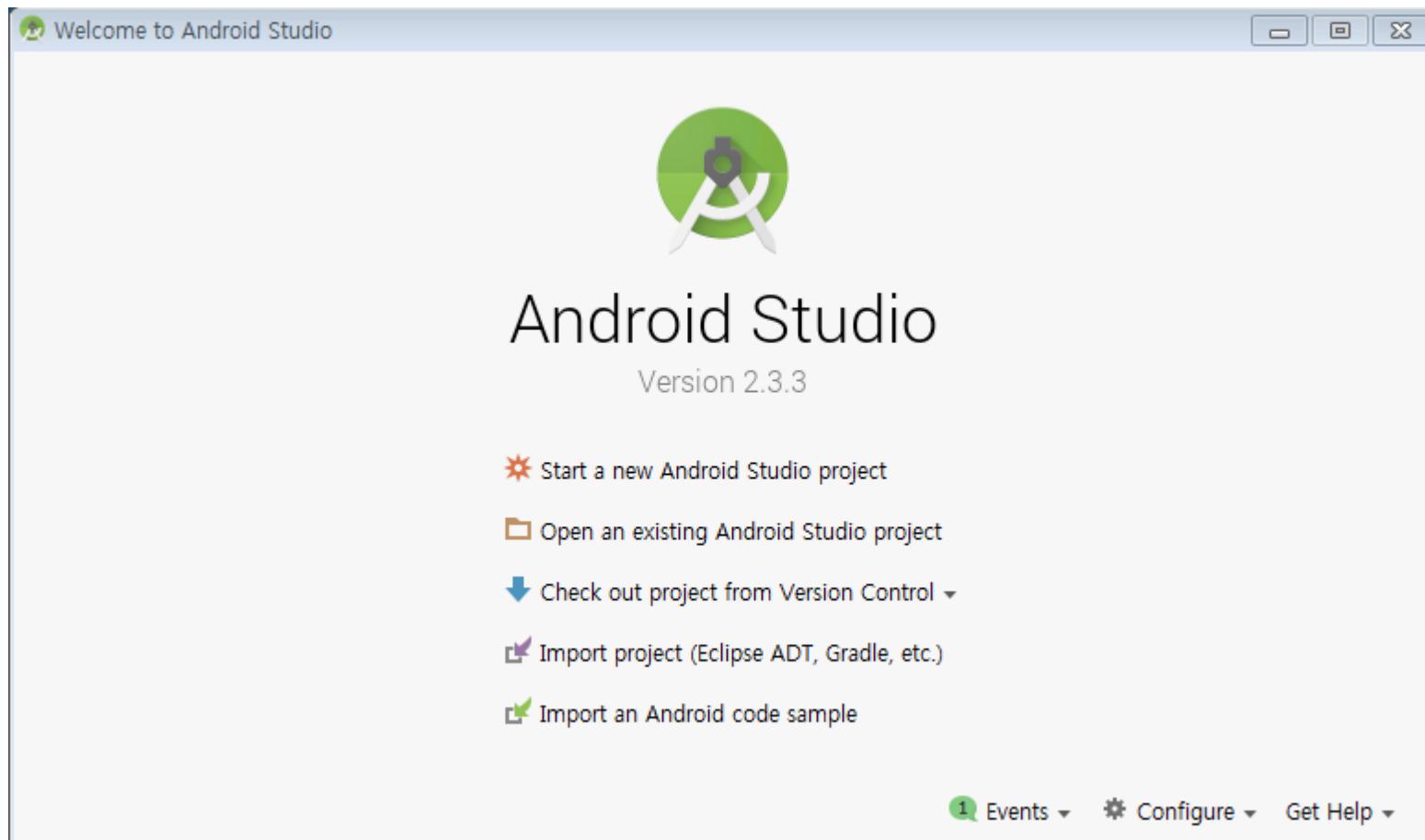


오디오 재생 원리



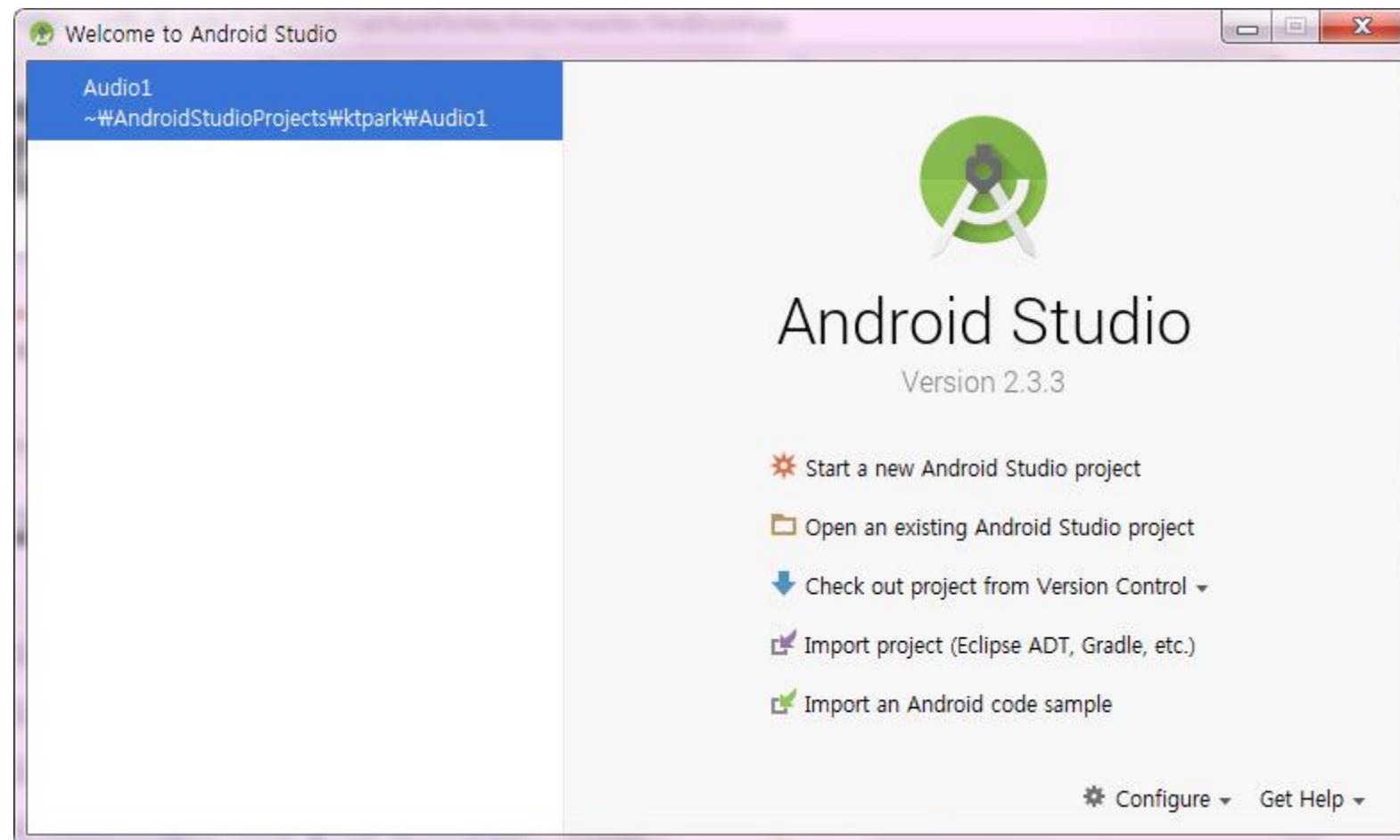
Start a new Android Studio project-type1

6

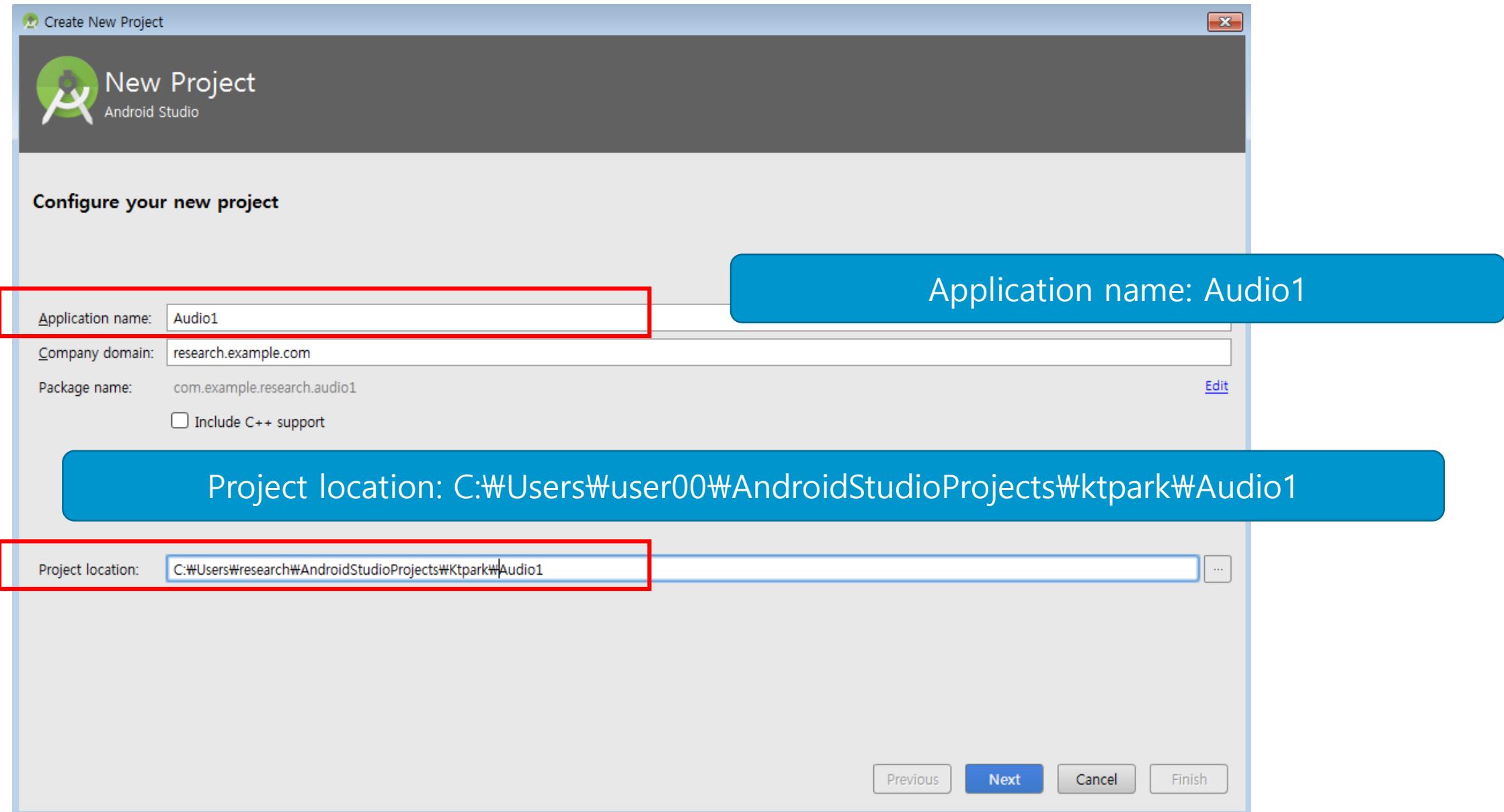


Start a new Android Studio project-type2

7



Configure your new project



Configure your new project

9

Application name:

Company domain:

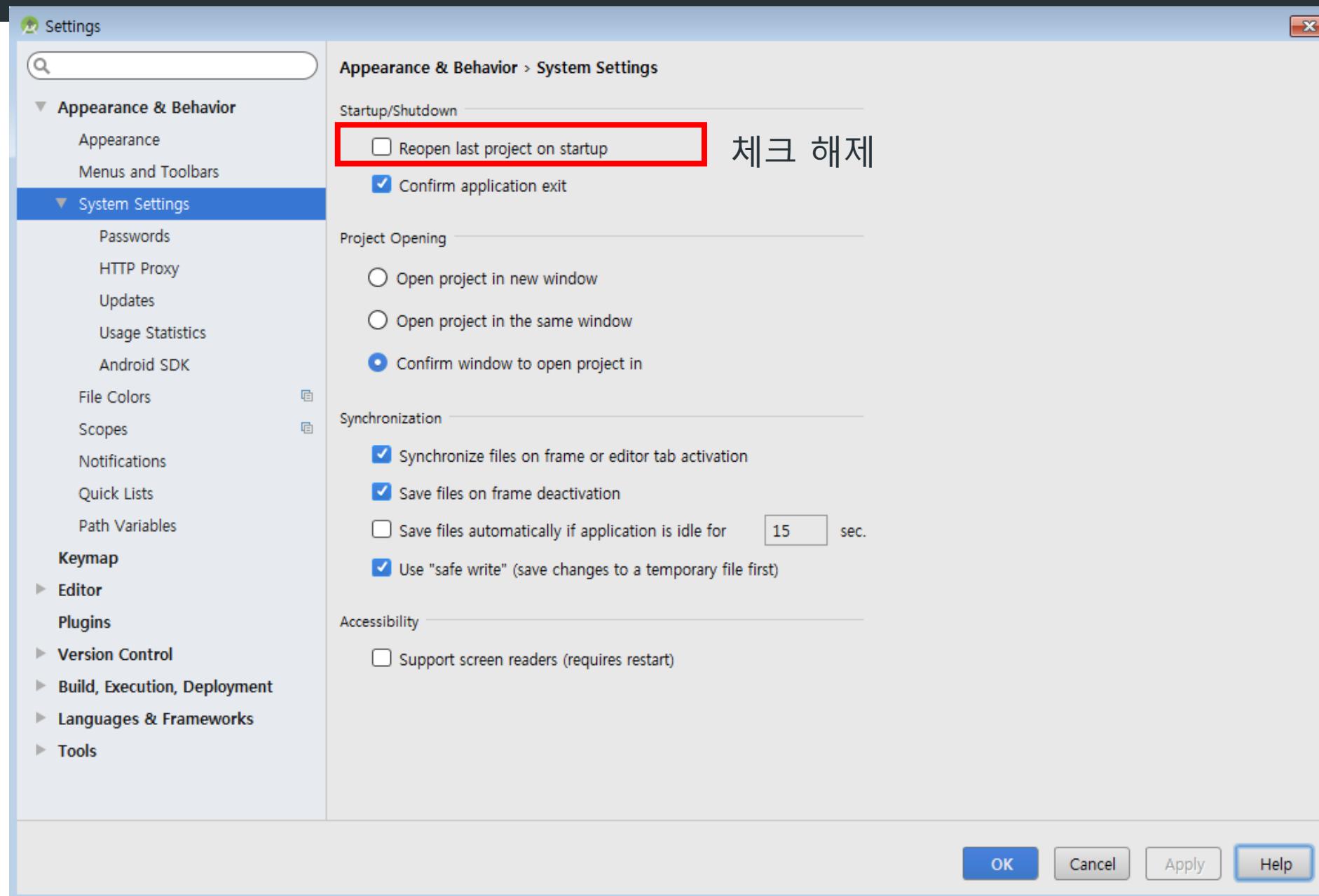
Package name: com.example.research.audio1 [Edit](#)

Include C++ support

Project location: [...](#)

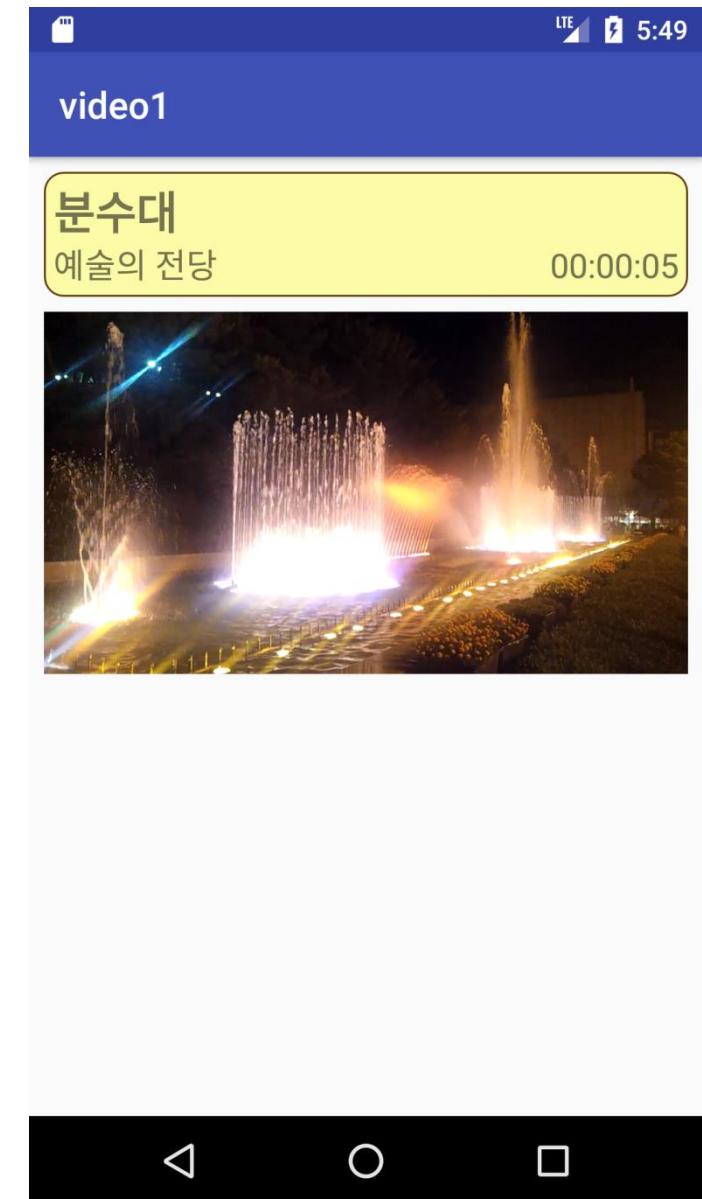
초기 화면 변경-유저가 프로젝트 선택

10



Step 0. 프로젝트 개요

11



Step 1. 프로젝트 생성

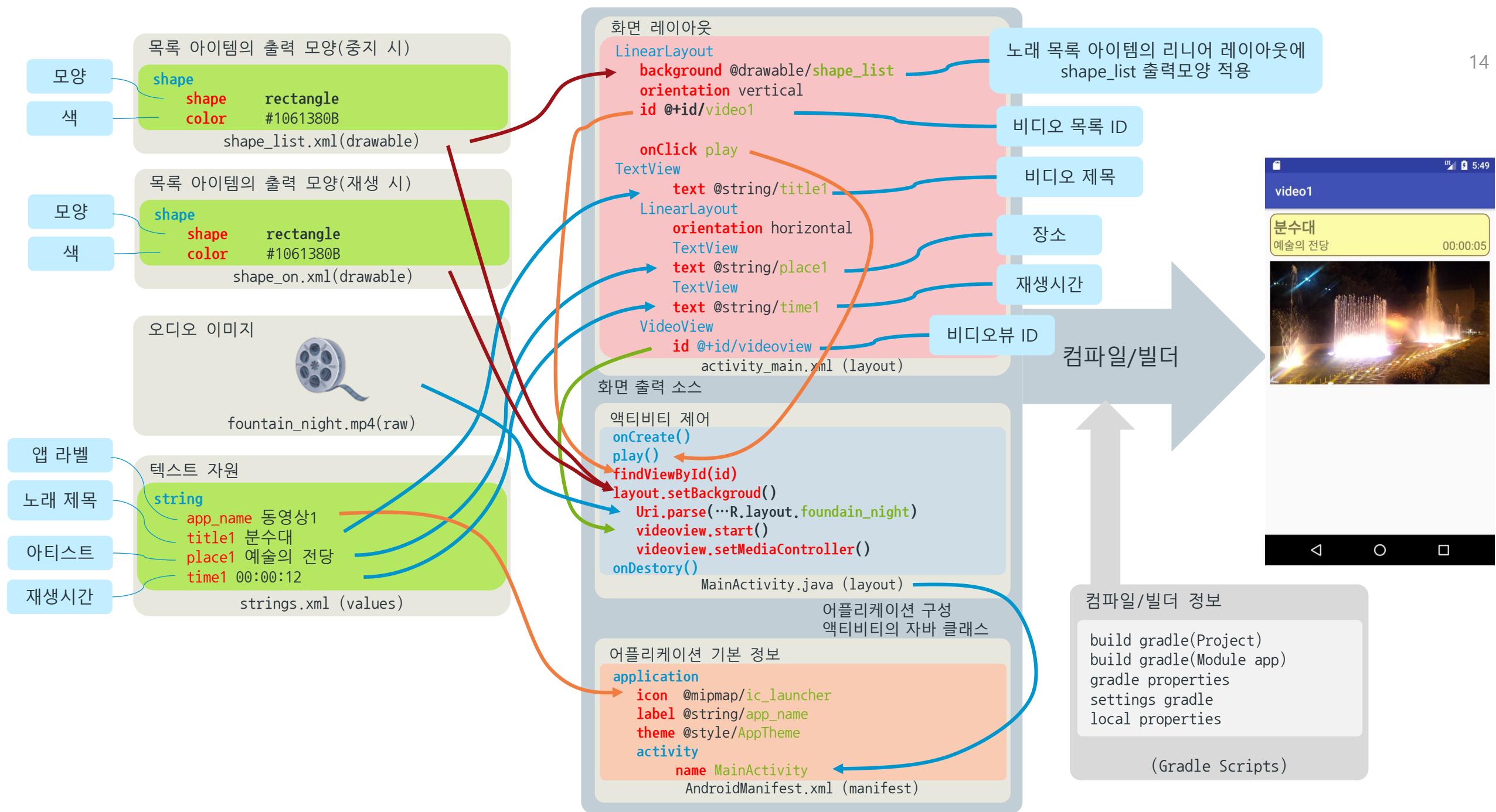
12

절차	내용
①프로젝트 시작	메뉴에서 ‘File → New Project’ 클릭
②프로젝트 구성	Application Name: Video1
	Company Domain: 사용자계정.example.com(디폴트 사용)
	Project location: ~\user00\AndroidStudioProject\ktpark\Video1
③제품형태	Phone and Tablet(사용할 안드로이드 버전 지정: Android 7.0 Nougat)
④액티비티 유형	Empty Activity
⑤파일 옵션	Activity Name: MainActivity (디폴트 사용)
	Layout Name: activity_main (디폴트 사용)

Step 2. 파일 편집

13

모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.사용자계정.video1	MainActivity.java	<ul style="list-style-type: none">비디오 목록 출력과 비디오 자동 재생
	drawable	shape_list.xml	<ul style="list-style-type: none">목록 아이템에 대한 출력 스타일 설계 (테두리, 패딩, 모서리) – 중지 시
		shape_on.xml	<ul style="list-style-type: none">목록 아이템에 대한 출력 스타일 설계 (테두리, 패딩, 모서리) - 재생시
	layout	activity_main.xml	<ul style="list-style-type: none">비디오 목록의 화면 배치목록 아이템에 출력 모양 적용 (shape_list.xml)
res	mipmap	ic_launcher.png	
	raw	fountain_night.mp4	<ul style="list-style-type: none">비디오 파일
	values	colors.xml	
		dimens.xml	<ul style="list-style-type: none">화면의 구성자원 크기(여백, 글자크기 등)
		strings.xml	<ul style="list-style-type: none">어플리케이션 라벨비디오에 대한 제목, 제작자, 재생시간에 대한 텍스트 리소스 정의
		styles.xml	



Step 2.1 이미지 파일 복사

15

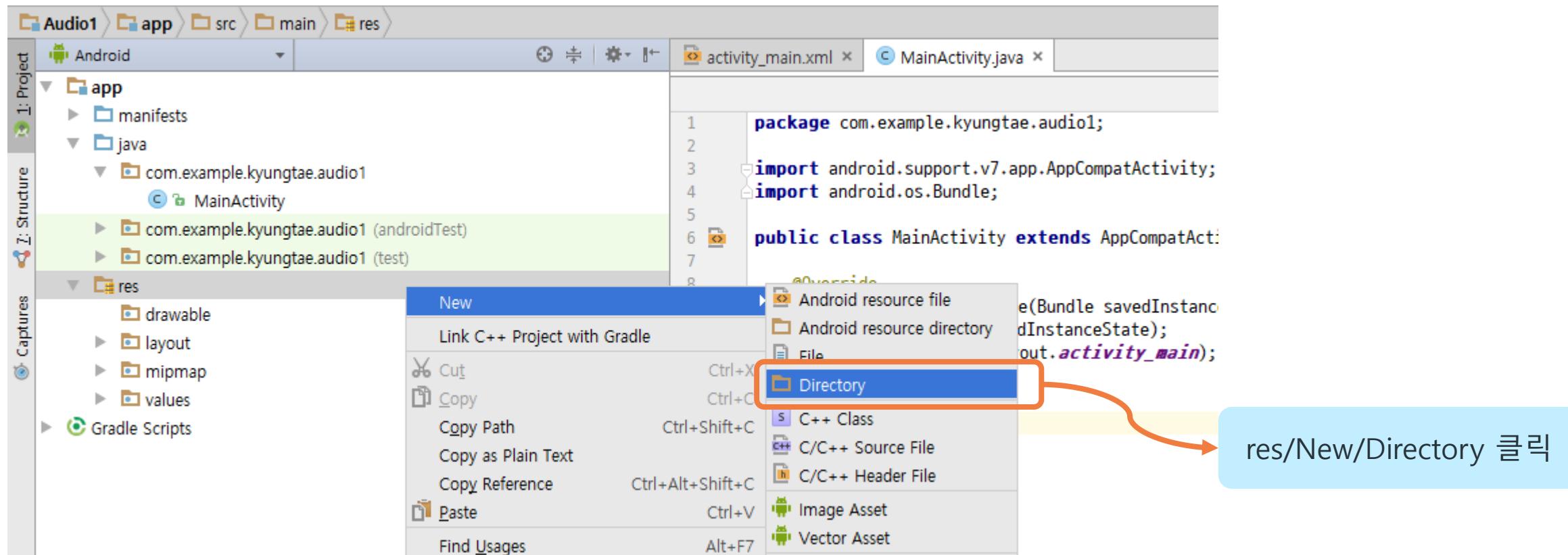
- res 폴더에 있는 **raw** 폴더에 fountain_night.mp4 파일 저장

모듈	폴더	소스 파일	내용
res	raw	fountain_night.mp4	동영상 파일

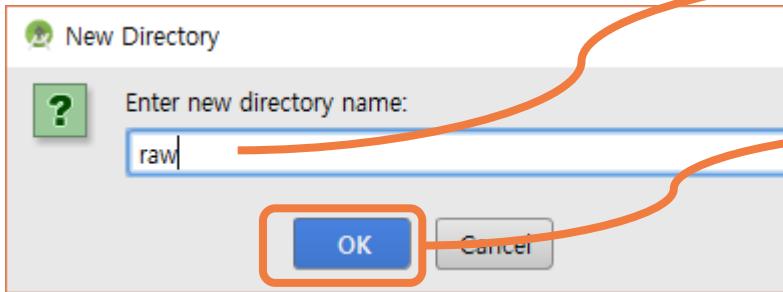
res/raw 폴더에
비디오 파일 올리기

drawable/raw 폴더에 비디오 파일 추가하기

- app → res → New → Directory 클릭



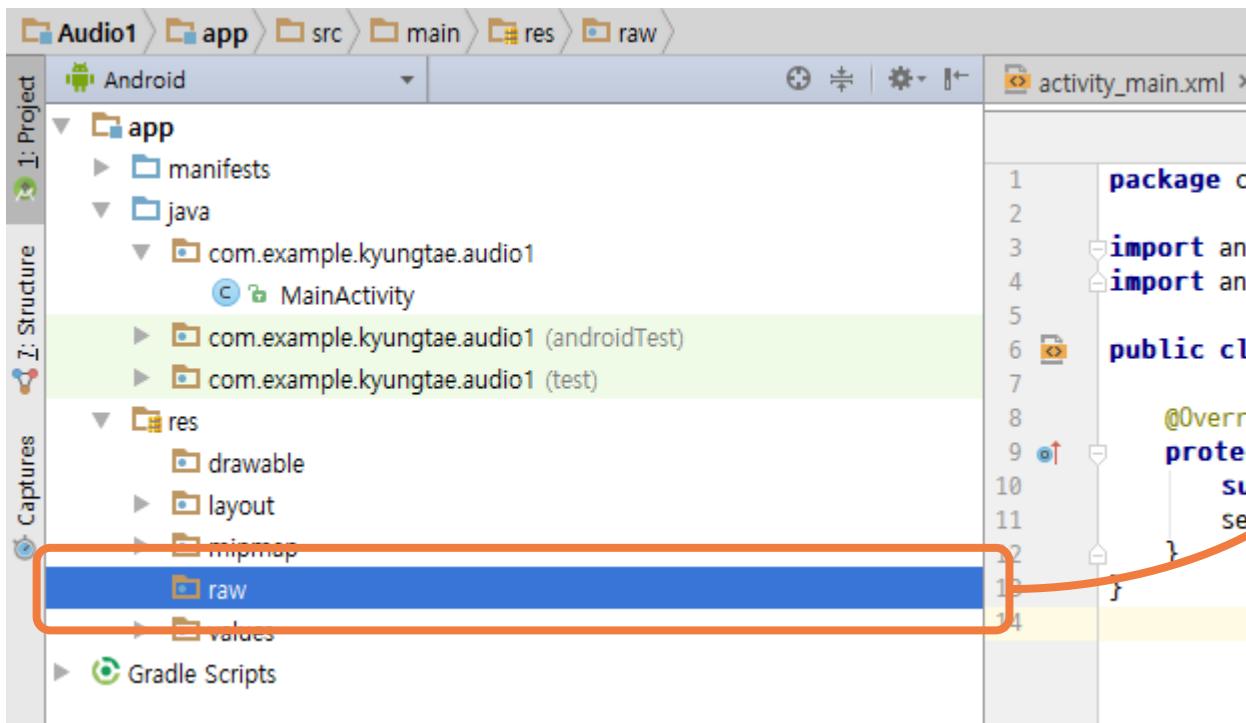
• 폴더 이름 작성



폴더 이름: raw

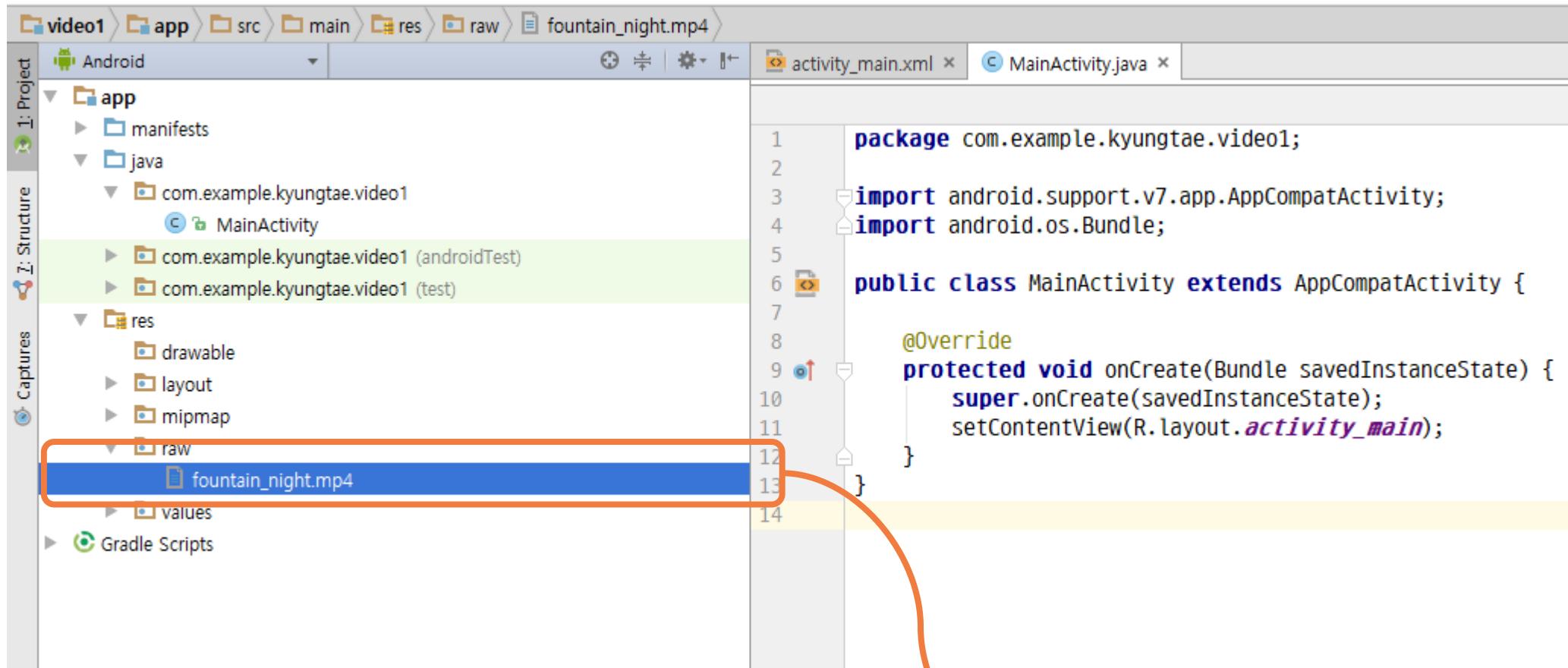
OK 클릭

• 실행 결과



raw 폴더 생성됨

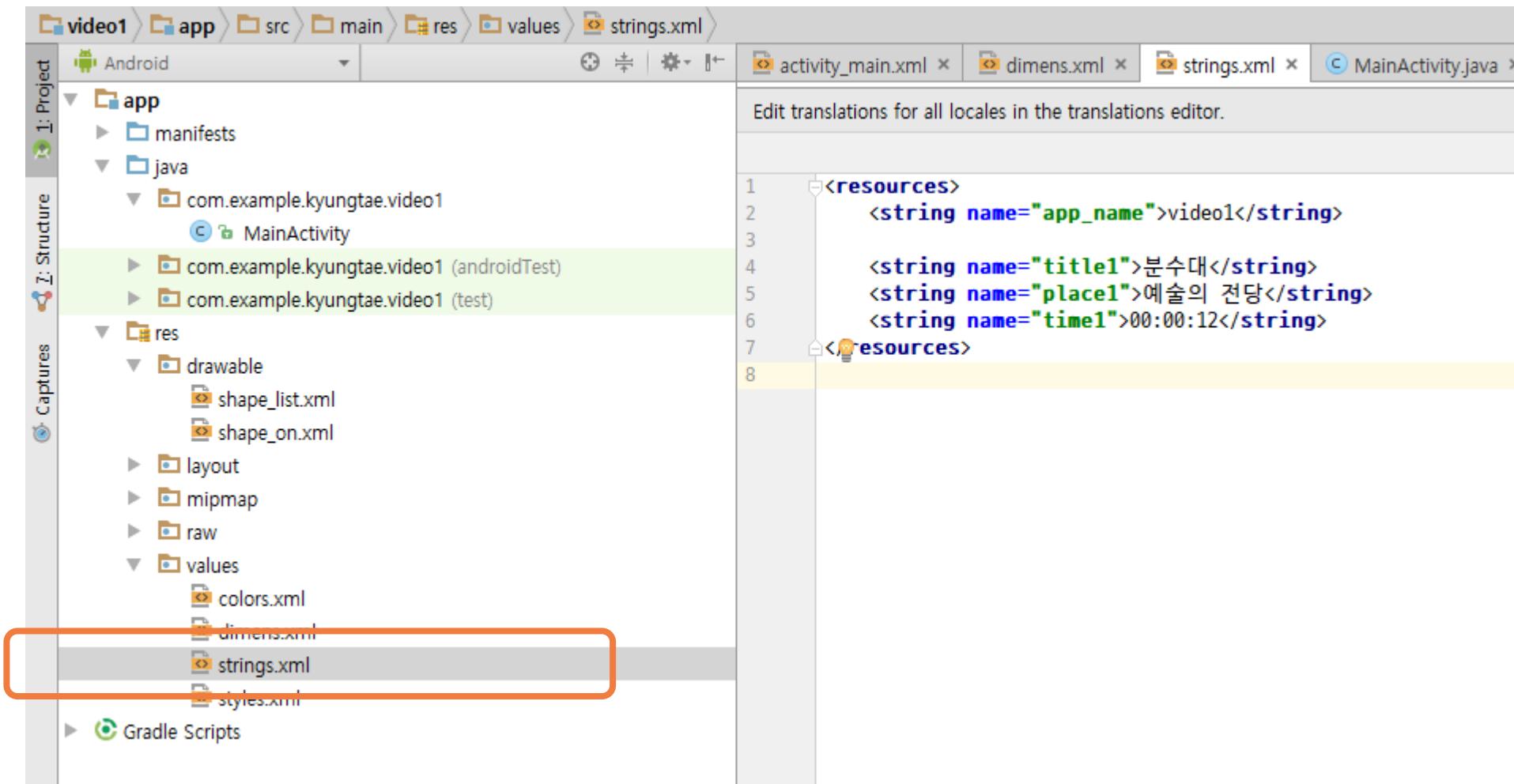
- 비디오 파일(fountain_night.mp4) 복사하기



fountain_night.mp4 복여넣기

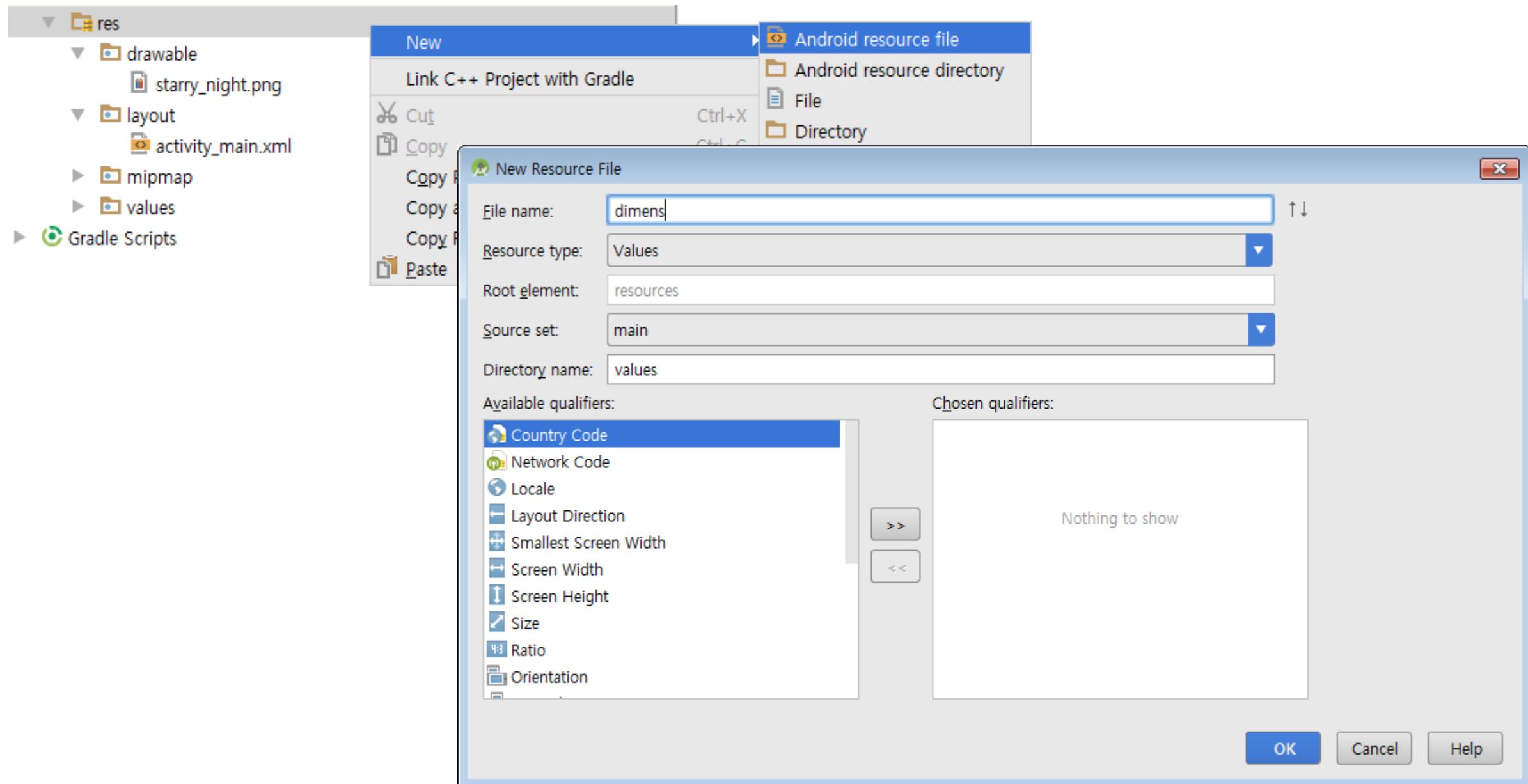
Step 2.2 텍스트 자원의 편집

- strings.xml



Step 2.2 화면 구성 자원(Dimens.xml)

21



- dimens.xml

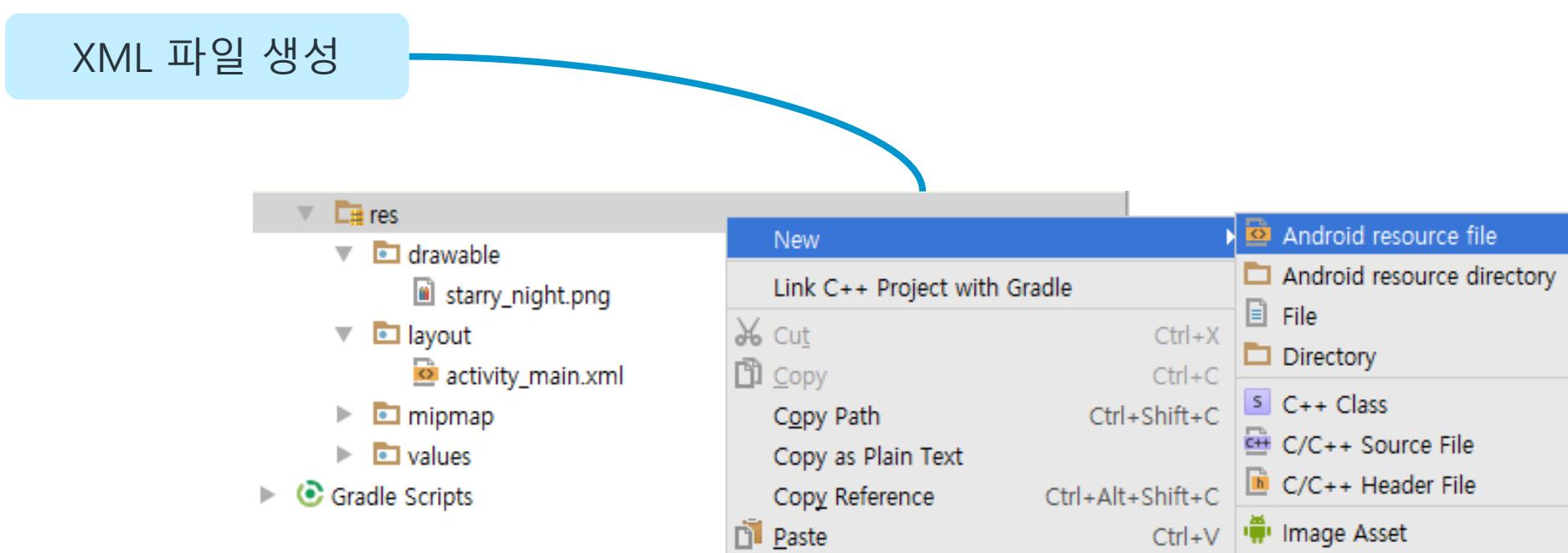
The screenshot shows the Android Studio interface with the project 'video1' open. The 'values' folder under 'res' contains four XML files: colors.xml, dimens.xml (which is selected and highlighted with a red box), strings.xml, and styles.xml. The 'dimens.xml' file is displayed in the main editor area. It contains the following code:

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
    <dimen name="activity_horizontal_margin">8dp</dimen>
    <dimen name="activity_vertical_margin">8dp</dimen>
    <dimen name="padding_horizontal">8sp</dimen>
    <dimen name="padding_vertical">8sp</dimen>
</resources>
```

Step 2.3 Drawable Resource 추가 및 편집

23

- **shape_list.xml** 생성(res/drawable 폴더)
 - drawable resource를 이용한 그림 출력



- Set New Resource File

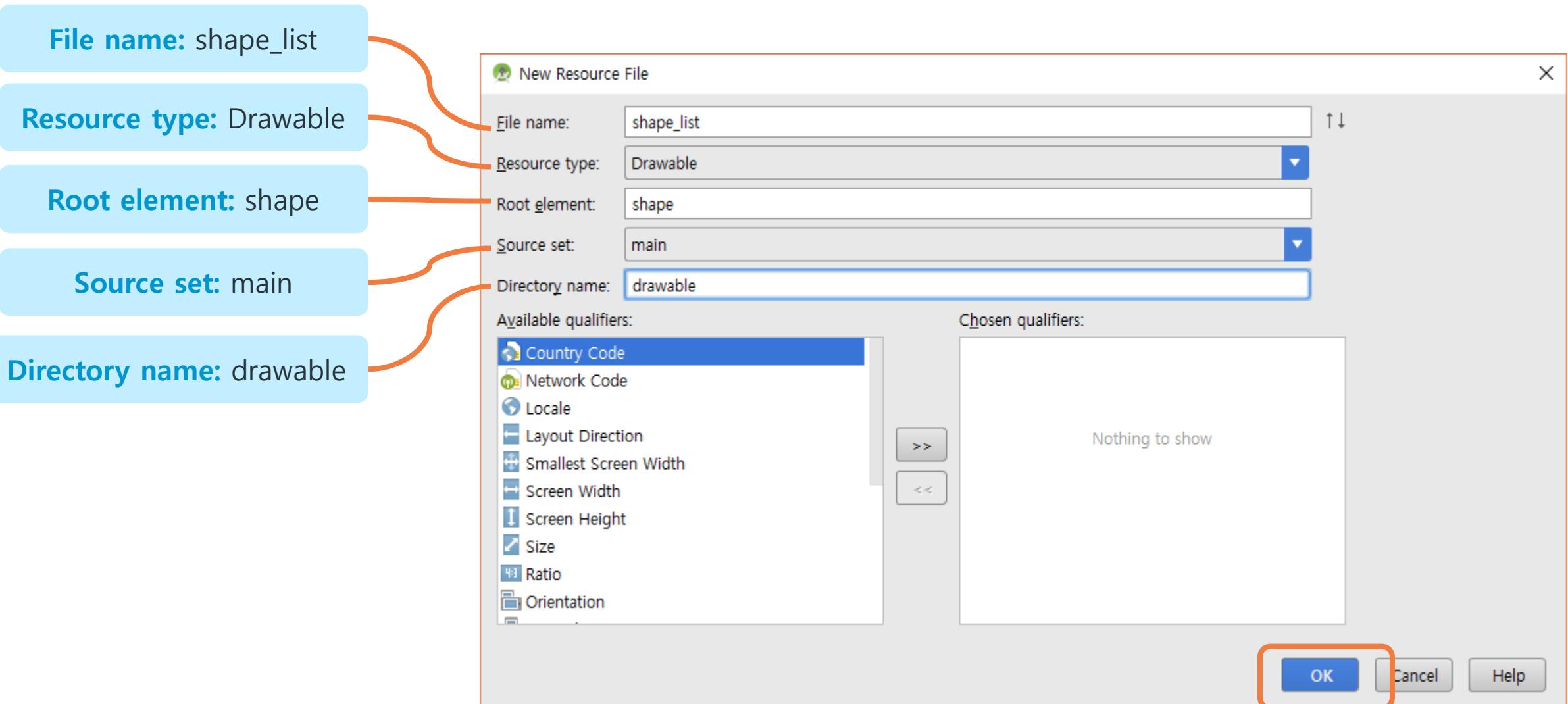
File name: shape_list

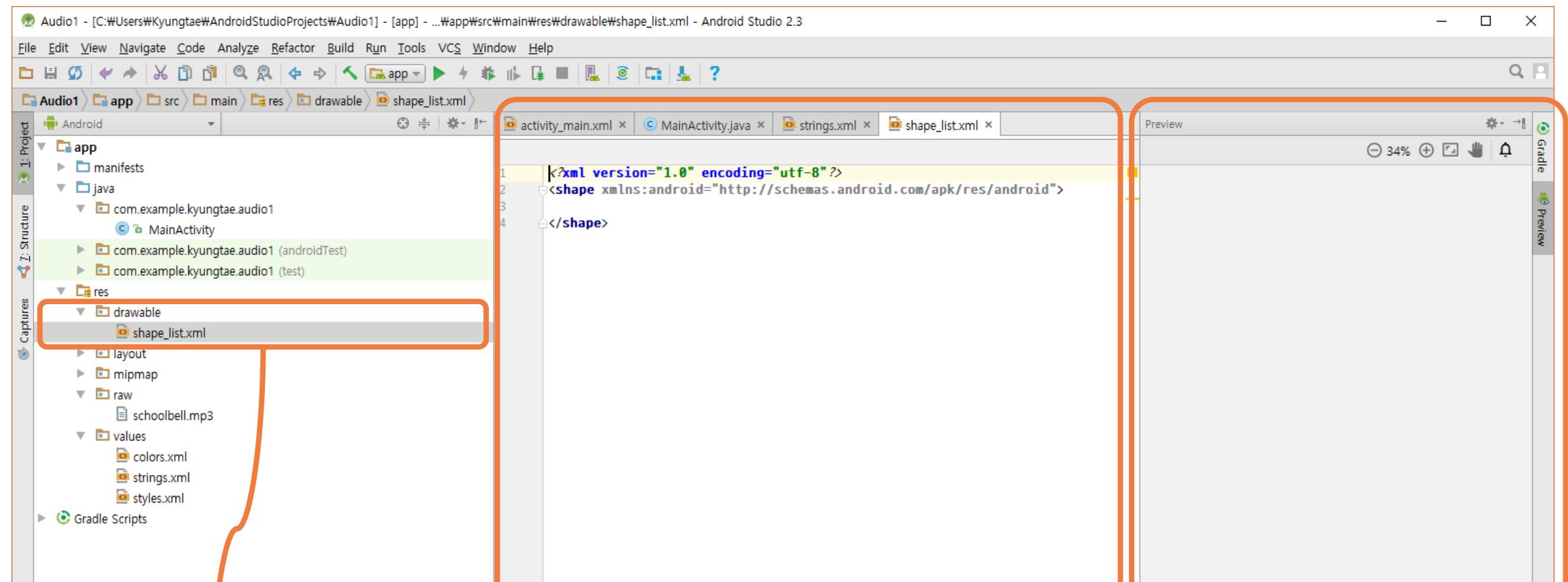
Resource type: Drawable

Root element: shape

Source set: main

Directory name: drawable





shape_list.xml 파일

shape_list.xml 파일의
텍스트 코딩 영역

shape_list 파일에 의한
shape 미리보기 영역

• shape_list.xml 소스(동영상 중지 시)

The screenshot shows the Android Studio code editor with the `shape_list.xml` file open. The code defines a shape with various properties:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#2261380B"/>
    <stroke android:width="1dp" android:color="#61380B"/>
    <padding android:top="5dp"
        android:bottom="5dp"
        android:right="5dp"
        android:left="5dp">
    </padding>
    <corners android:radius="10dp"/>
</shape>
```

Annotations on the left side explain specific parts of the XML code:

- 출력모양을 사각형으로 지정 (Line 3: `android:shape="rectangle"`)
- 출력모양을 내부의 색 (Line 5: `<solid android:color="#2261380B"/>`)
- 출력모양을 테두리의 색 (Line 7: `<stroke android:width="1dp" android:color="#61380B"/>`)
- 내부 패딩 정보 (Line 12: `<padding android:top="5dp" android:bottom="5dp" android:right="5dp" android:left="5dp">`)
- 출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp) (Line 15: `<corners android:radius="10dp"/>`)

• shape_on.xml 소스(동영상 재생 시)

출력모양을 사각형으로 지정

출력모양을 내부의 색

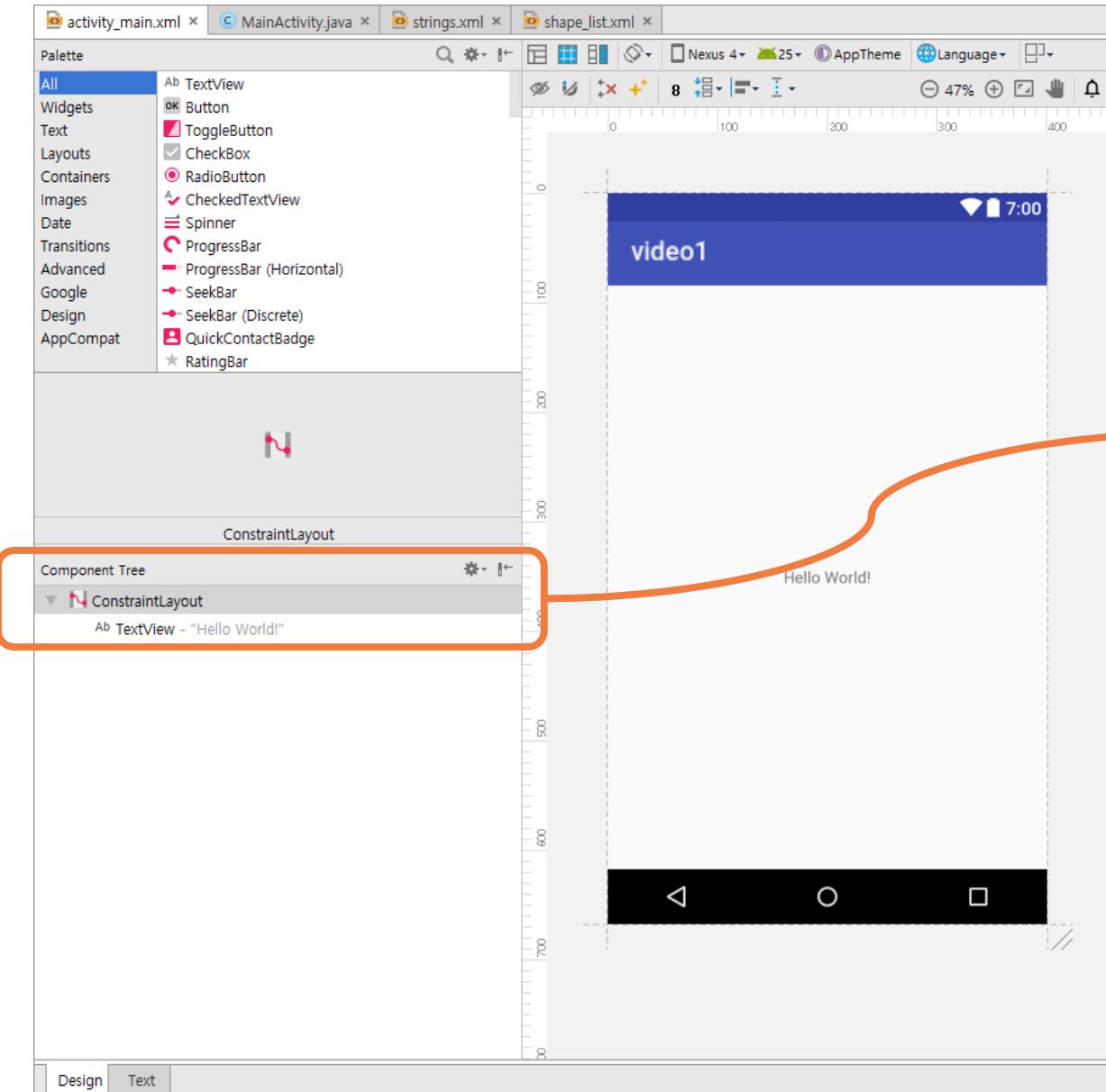
출력모양을 테두리의 색

내부 패딩 정보

출력모양 모서리를 둥근 모양
으로 지정(반지름은 5dp)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#ffff00"/>
    <stroke android:width="1dp" android:color="#61380B"/>
    <padding android:top="5dp"
        android:bottom="5dp"
        android:left="5dp"
        android:right="5dp">
    </padding>
    <corners android:radius="10dp"/>
</shape>
```

2.4 화면 설계



ConstraintLayout →
LinearLayout으로 변경

Text 에디터에서 수정

• Layout 변경 및 기본 TextView 삭제

The image shows two screenshots of the Android Studio code editor for the file `activity_main.xml`. In the top screenshot, the XML code uses a `ConstraintLayout` as the root container. A red box highlights the opening tag `<android.support.constraint.ConstraintLayout`. In the bottom screenshot, the XML code has been modified to use a `LinearLayout` as the root container. A red box highlights the opening tag `<LinearLayout`. A large orange arrow points from the `ConstraintLayout` section in the top screenshot to the `LinearLayout` section in the bottom screenshot. Another orange arrow points from the `<LinearLayout>` tag in the bottom screenshot to the right, with the Korean word "삭제" (Delete) written next to it.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.example.kyungtae.audio1.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

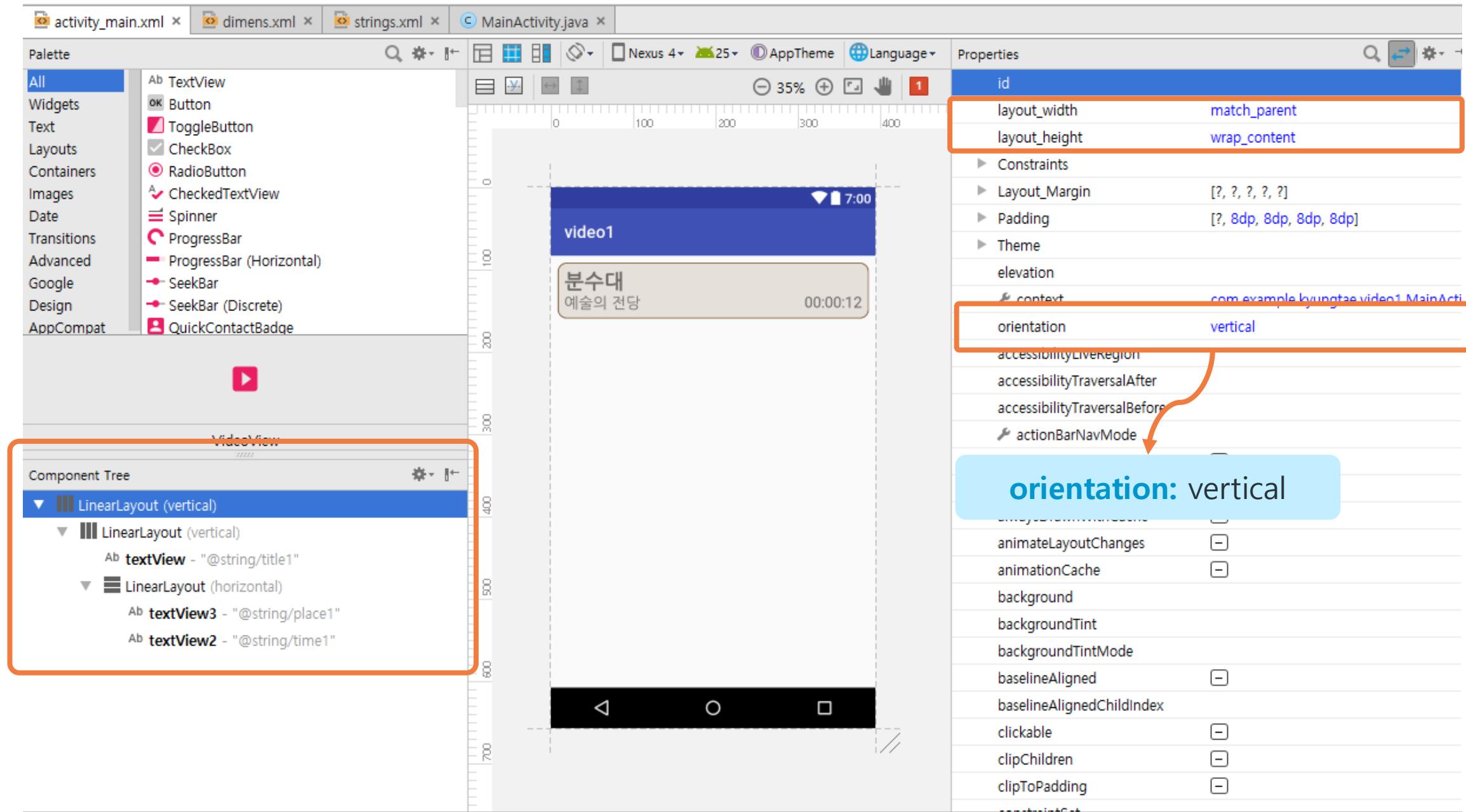
</android.support.constraint.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="com.example.kyungtae.audio1.MainActivity">

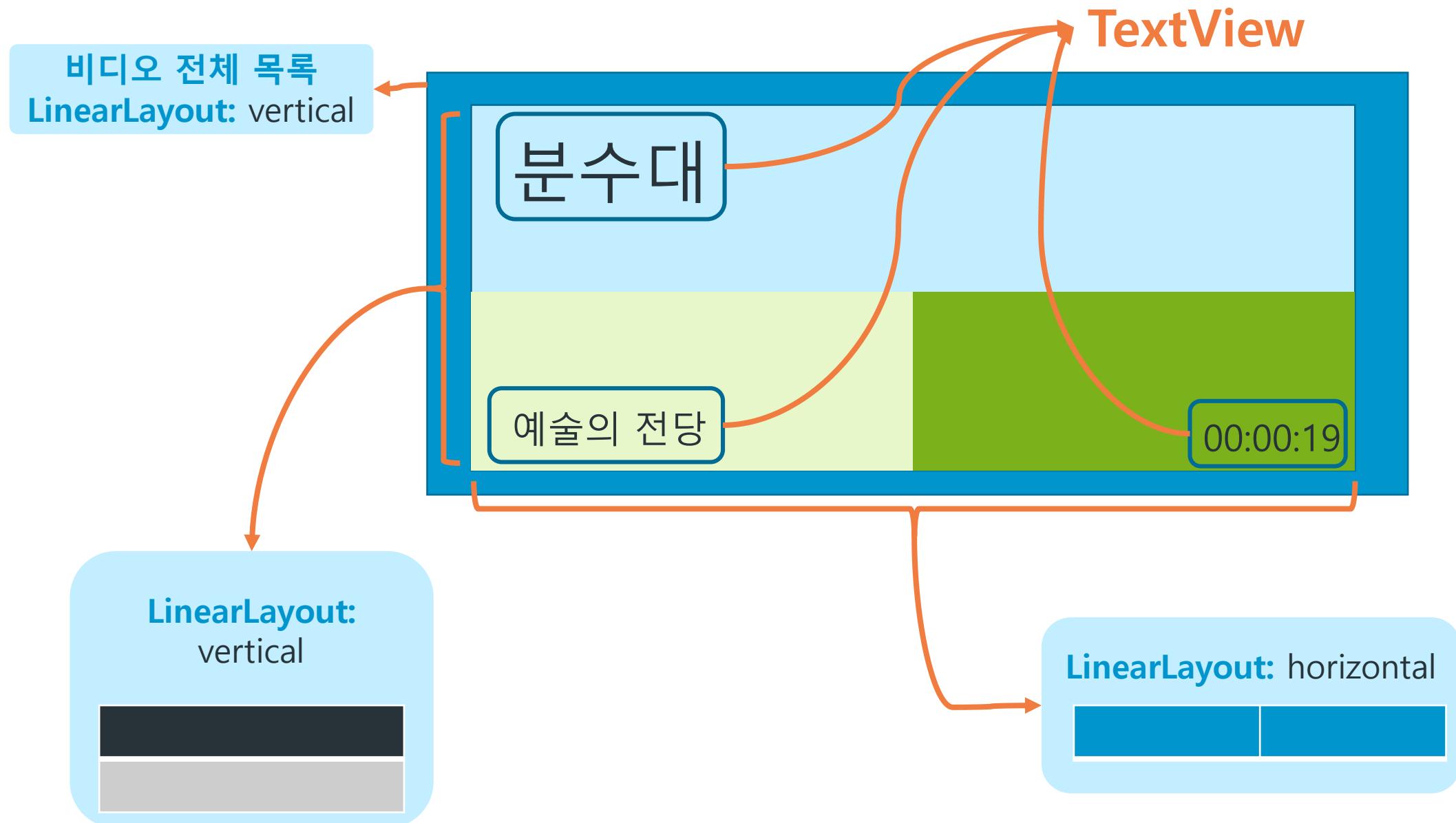
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</LinearLayout>
```

• 동영상 정보 표시 -



- 비디오 목록 표시를 위한 Layout 구조

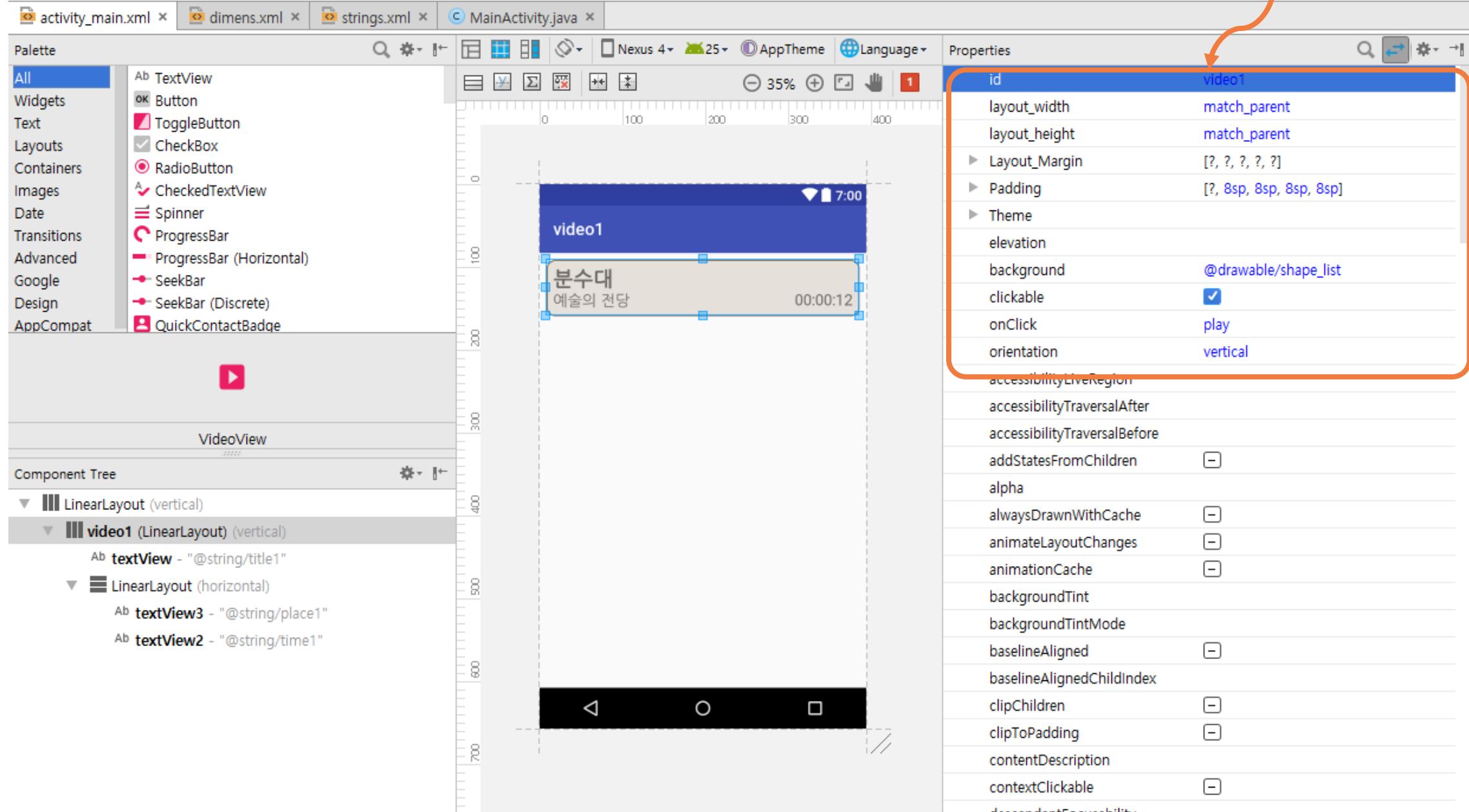


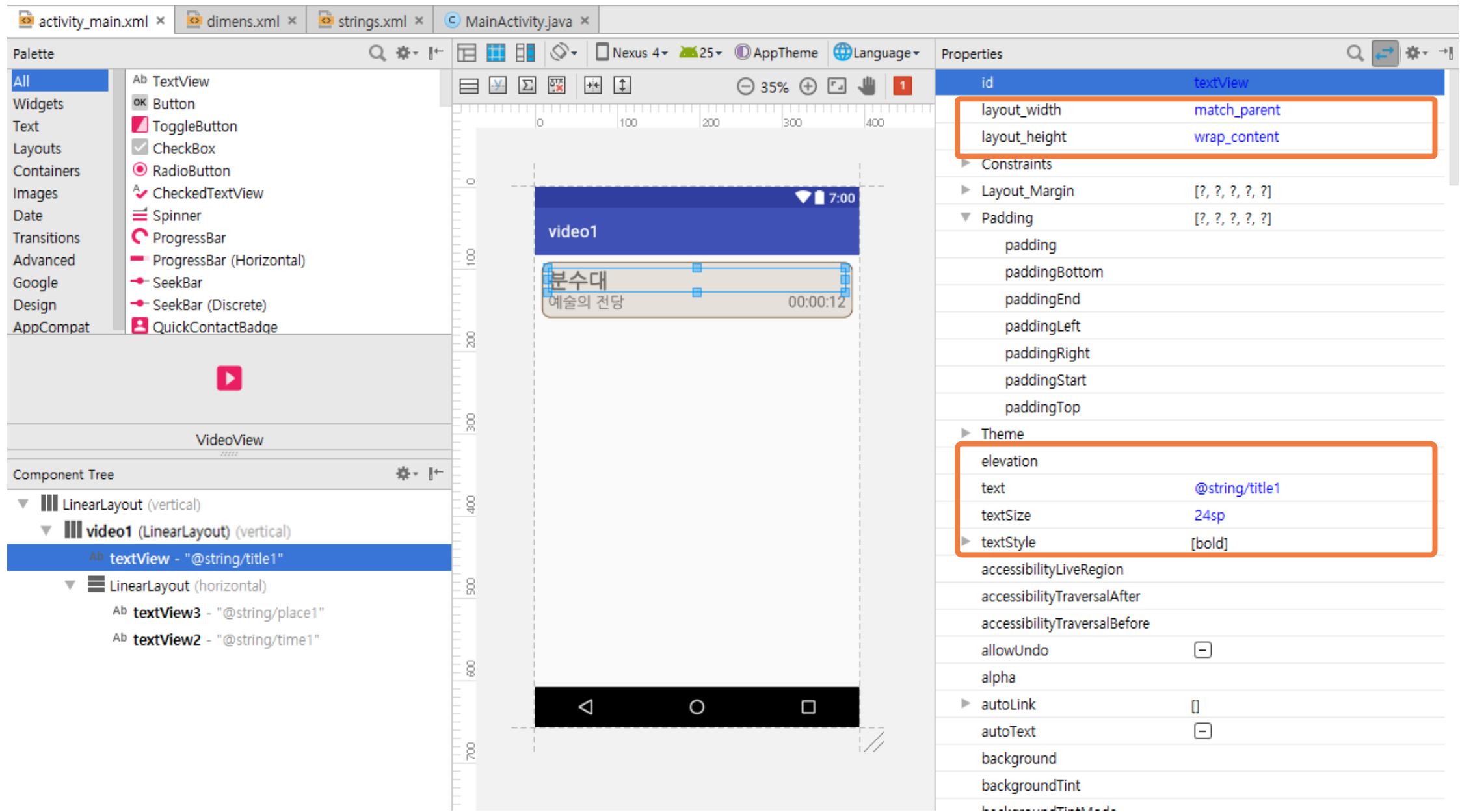
• padding 속성을 이용한 여백 설정

The screenshot shows the Android Studio interface with the following components:

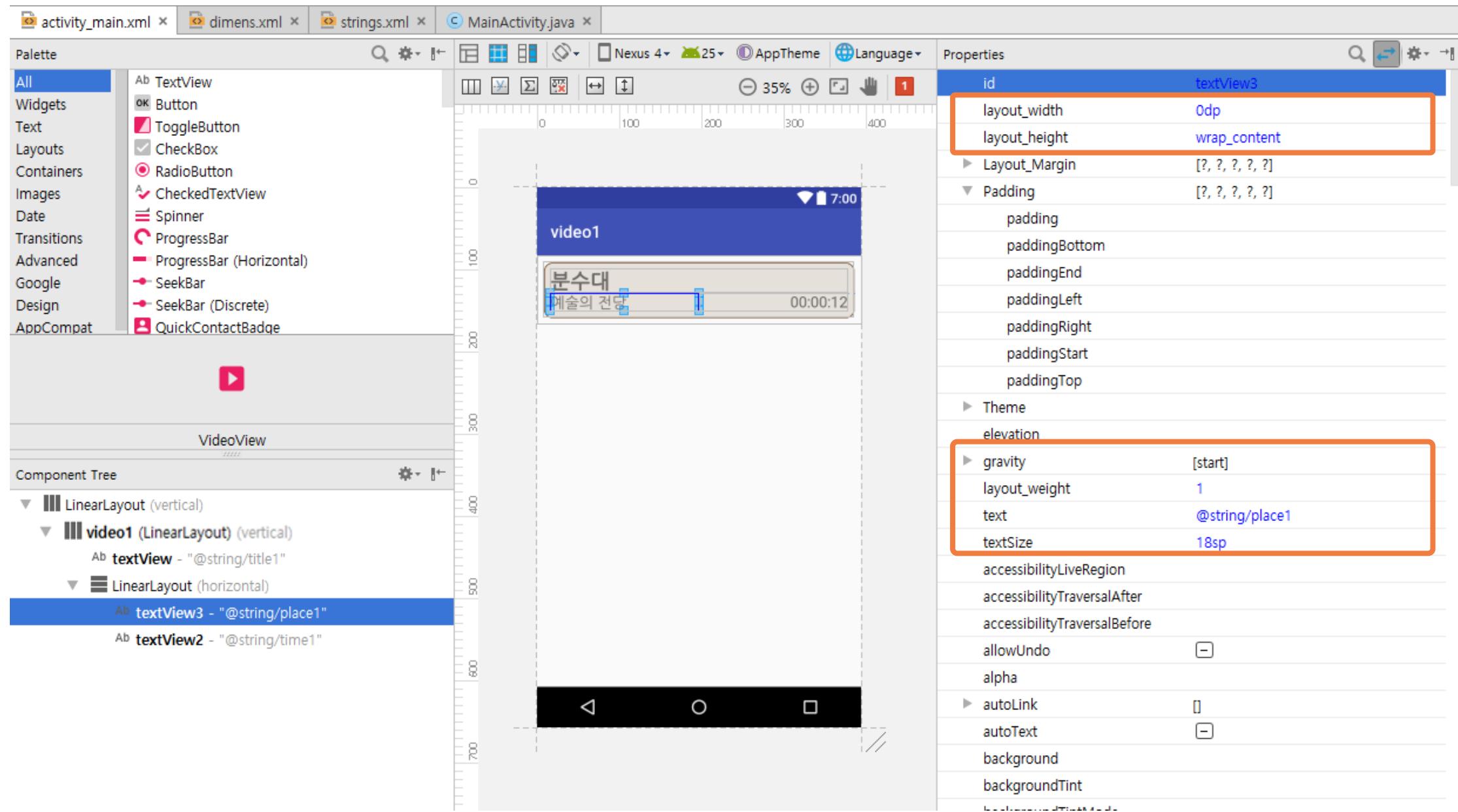
- Top Bar:** activity_main.xml, dimens.xml, strings.xml, MainActivity.java.
- Palette:** All, Widgets, Text, Layouts, Containers, Images, Date, Transitions, Advanced, Google, Design, AppCompat.
- Layout Editor:** Shows a video player interface with a title bar, a progress bar, and a bottom navigation bar. The progress bar displays "분수대" and "예술의 전당" with a seek bar at 00:00:12 and a duration of 7:00.
- Component Tree:** LinearLayout (vertical) containing a TextView and another LinearLayout (horizontal) containing three TextViews: textView3 (@string/place1), textView2 (@string/time1), and textView (@string/title1).
- Properties Panel:** Displays various properties for the selected view, including:
 - id:** video1
 - layout_width:** match_parent
 - layout_height:** wrap_content
 - Constraints:** (checkbox)
 - Layout_Margin:** [?, ?, ?, ?, ?]
 - Padding:** [?, 8dp, 8dp, 8dp, 8dp] (highlighted with an orange box)
 - padding:** (checkbox)
 - paddingBottom:** @dimen/activity_vertical_margin
 - paddingLeft:** @dimen/activity_horizontal_margin
 - paddingRight:** @dimen/activity_horizontal_margin
 - paddingTop:** @dimen/activity_vertical_margin
 - paddingEnd:** (checkbox)
 - paddingStart:** (checkbox)
- Theme:** (checkbox)
- padding:** (checkbox) (highlighted with a blue callout bubble)
- dimens 리소스 사용:** (text in the callout bubble)

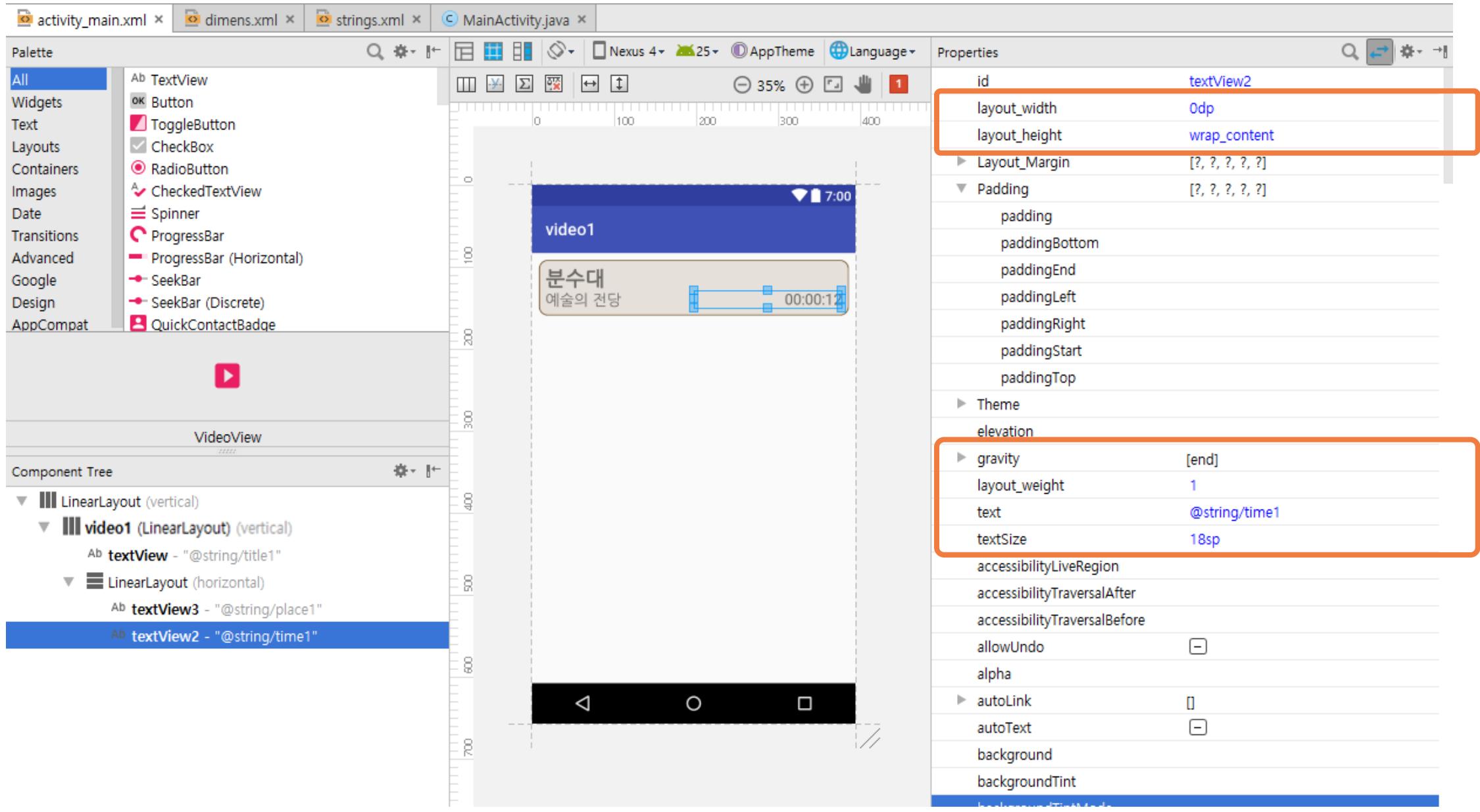
• 비디오 제목 레이아웃 속성 설정(클릭시 동영상 ...)



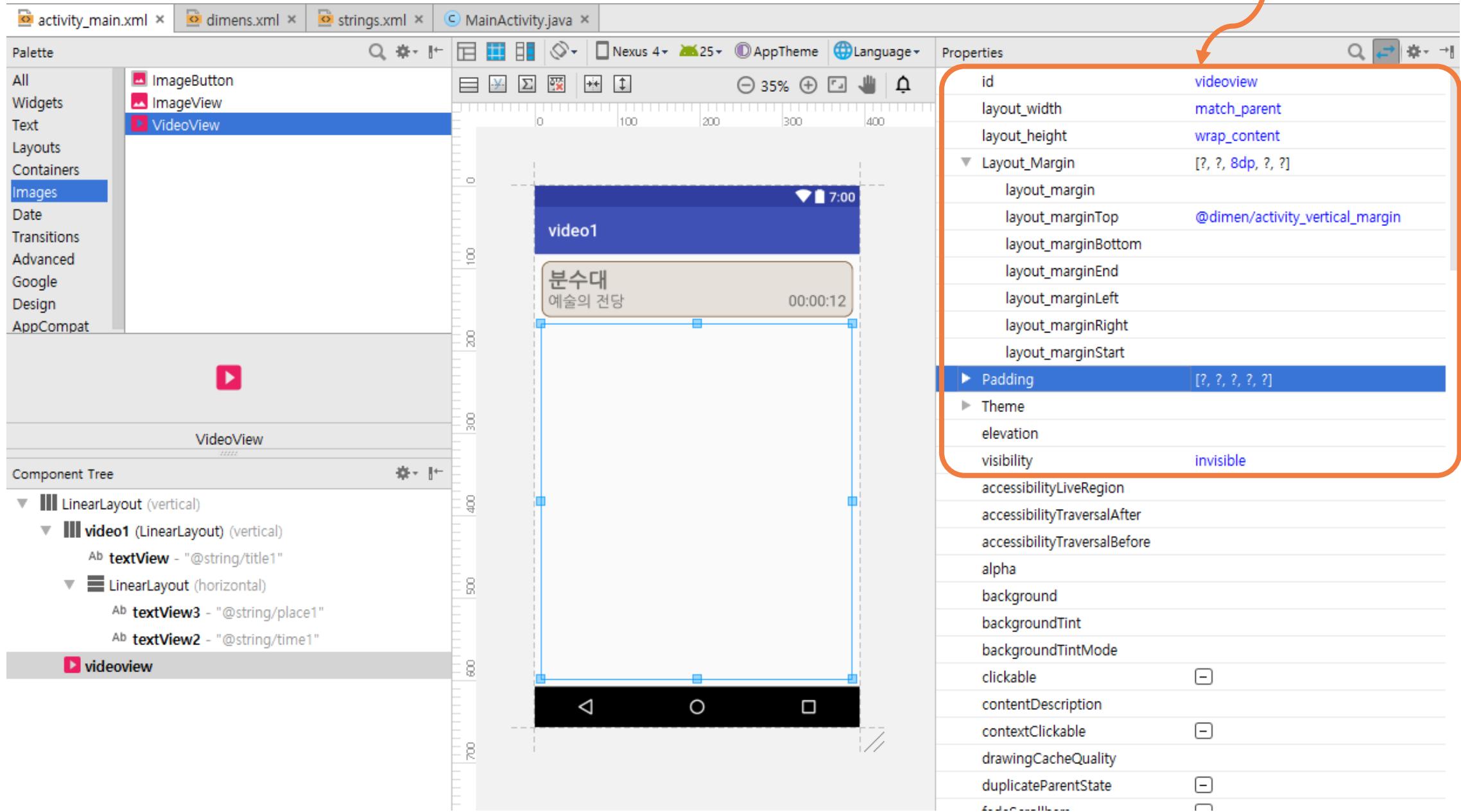


• 비디오 설명(place) Layout





• 비디오



2.5 Activity 제어(MainActivity.java)

39

- 비디오 리소스에 대한 VideoView를 생성

```
1 package com.example.kyungtae.video1;  
2  
3 import android.content.res.Resources;  
4 import android.graphics.drawable.Drawable;  
5  
6 import android.net.Uri;  
7 import android.support.v7.app.AppCompatActivity;  
8 import android.os.Bundle;  
9 import android.view.View;  
10 import android.widget.LinearLayout;  
11 import android.widget.VideoView;  
12 import android.widget.MediaController;  
13  
14 public class MainActivity extends AppCompatActivity {  
15     VideoView videoView = null; // 비디오뷰 객체 생성(null)  
16  
17     @Override  
18     protected void onCreate(Bundle savedInstanceState) {  
19         super.onCreate(savedInstanceState);  
20         setContentView(R.layout.activity_main);  
21         videoView = (VideoView) findViewById(R.id.videoview); // 비디오뷰 객체 초기화  
22     }  
23 }  
24
```

• 비디오 제목을 클릭했을 때 호출되는 메소드(play()) 추가

40

재생 중 일 때

정지 일 때

```
25 public void play(View v){  
26     int id = v.getId();  
27     LinearLayout layout = (LinearLayout) findViewById(id);  
28  
29     Resources res = getResources();  
30  
31     if (videoView.isPlaying()){  
32         videoView.pause();  
33         Drawable drawable = res.getDrawable(R.drawable.shape_list);  
34         layout.setBackground(drawable);  
35     }else{  
36  
37         Uri uri = Uri.parse("android.resource://com.example.kyungtae.video1/" + R.raw.fountain_night);  
38  
39         videoView.setVideoURI(uri);  
40         videoView.start();  
41         videoView.setVisibility(View.VISIBLE);  
42  
43         Drawable drawable = res.getDrawable(R.drawable.shape_on);  
44         layout.setBackground(drawable);  
45  
46         MediaController mc = new MediaController(this);  
47         videoView.setMediaController(mc);  
48  
49     }  
50 }  
51 }
```

미디어 플레이어 중지

새로운 Drawable 객체 인식해서 선택
한 비디오의 배경으로 설정

uri위치의 동영상재생

동영상 파일의 uri 인식

새로운 Drawable 객체 인식해서 선택
한 비디오의 배경으로 설정

미디어 제어기 생성

비디오뷰의 미디어 제어기로 설정

- 프로젝트 리소스를 얻기 위한 함수 추가

```
52
53     @Override
54     protected void onDestroy() {
55         videoView.pause();
56         super.onDestroy();
57     }
58
59
```

액티비티 종료

비디오 뷰 중지

클래스와 속성/메소드

- 클래스

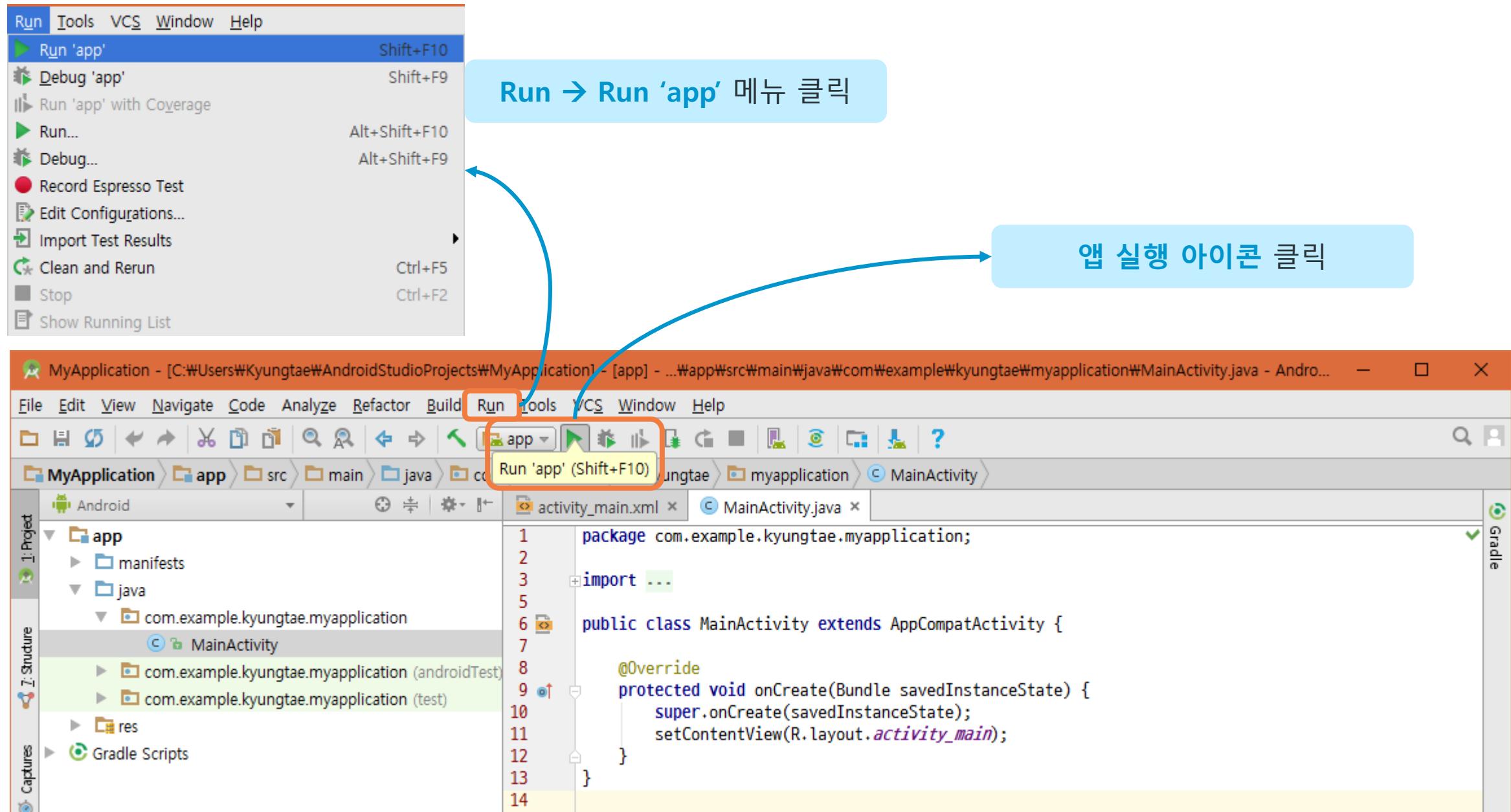
클래스	설명
MediaController	미디어 실행 제어를 포함하는 뷰

- 메소드

클래스	메소드	설명
VideoView	Boolean <code>isPlaying()</code>	비디오뷰의 실행 여부
	<code>void pause()</code>	비디오뷰의 중지
	<code>void setMediaController(MediaController controller)</code>	미디어 제어기를 설정함

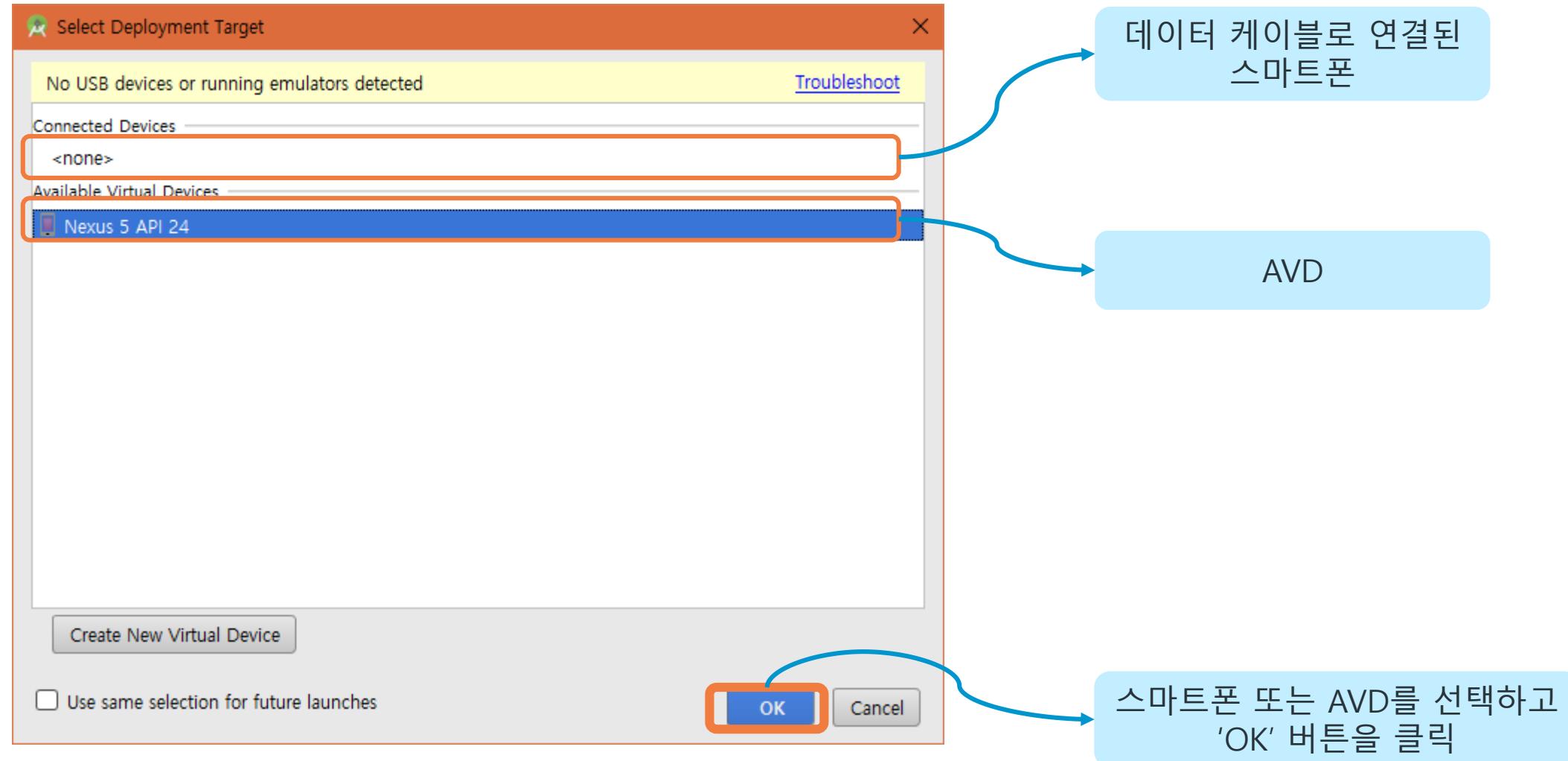
Step 3. 프로젝트 실행

43

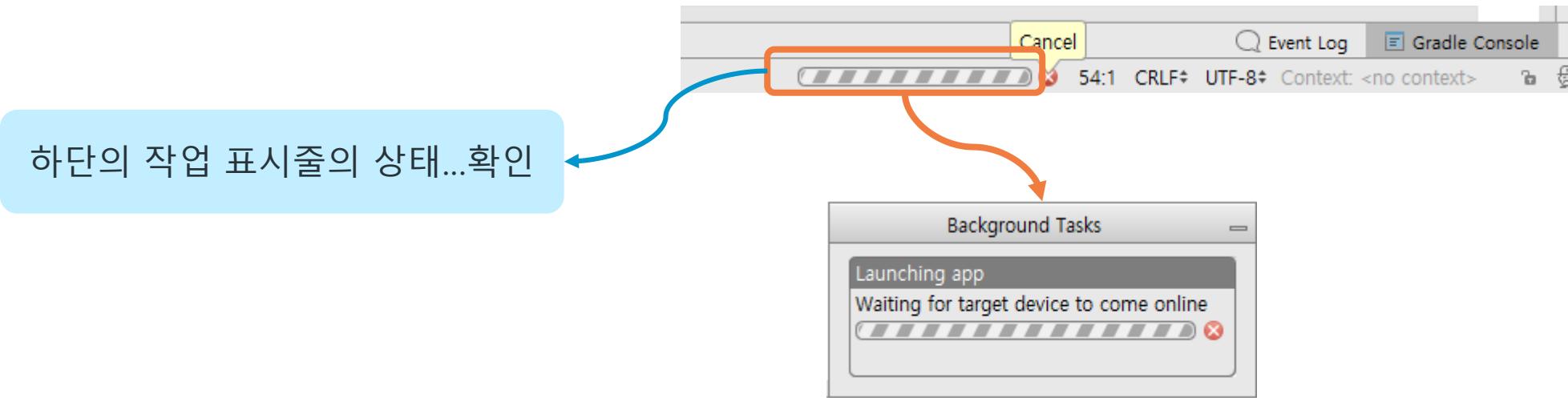


• AVD 장비 선택하기

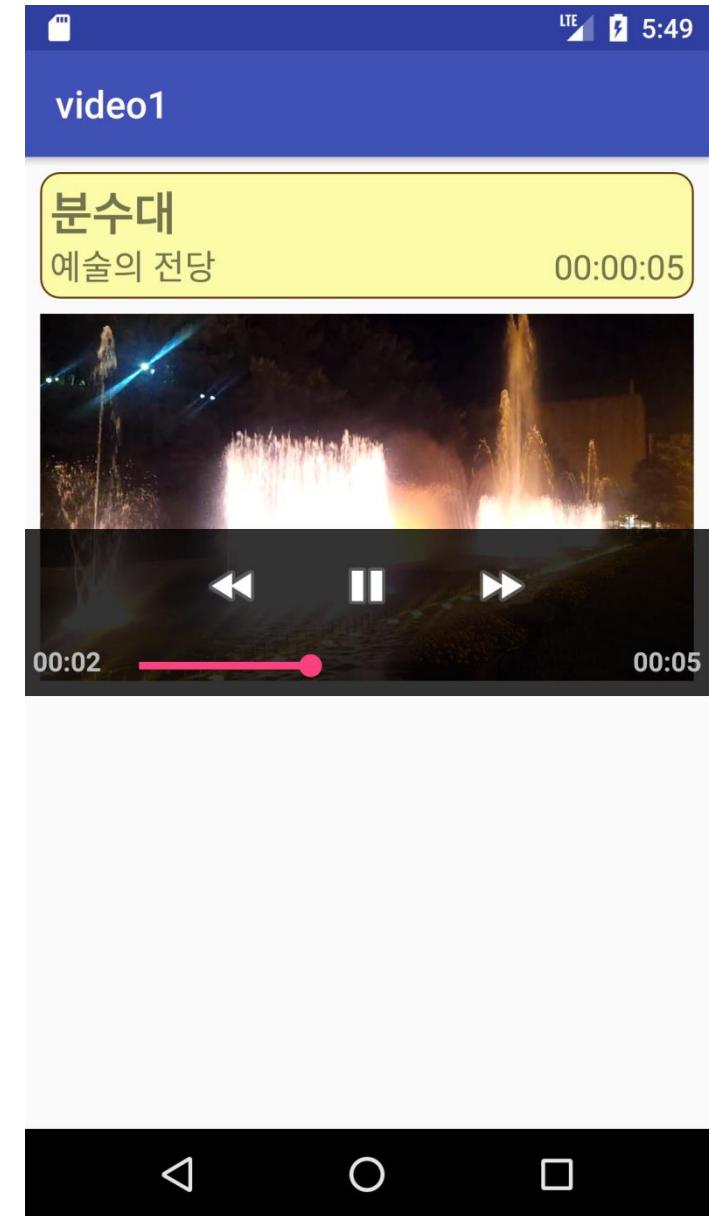
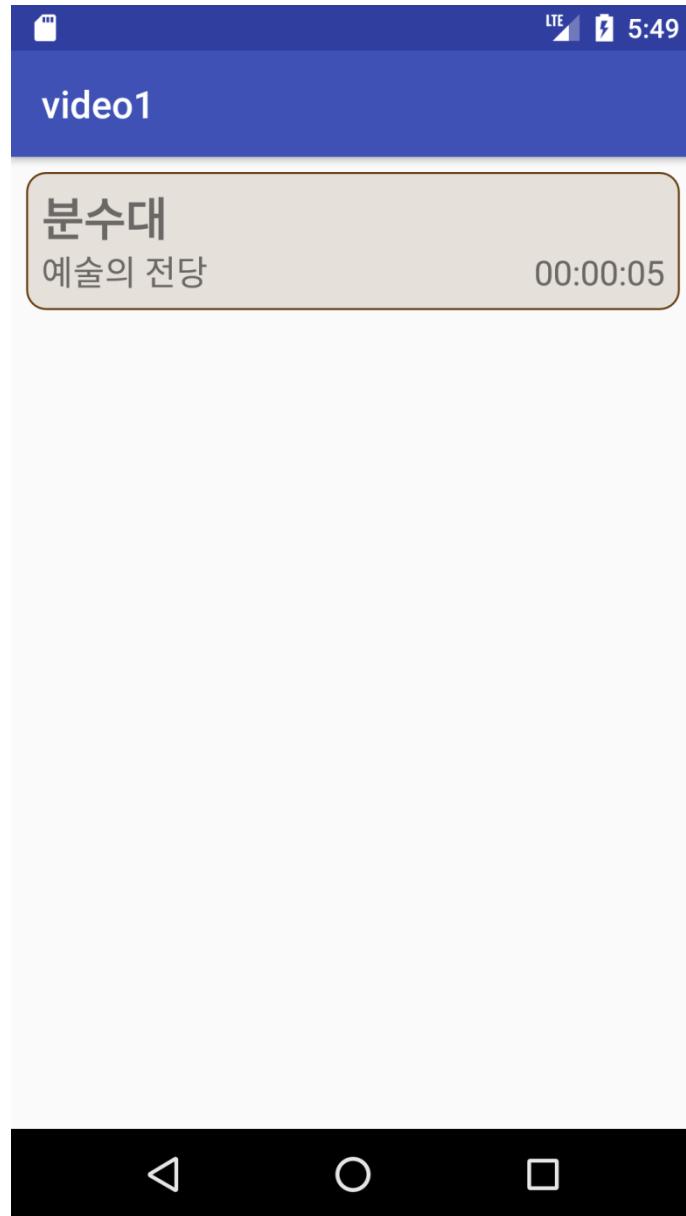
44



- 에뮬레이터는 실행되지만 앱이 실행되지 않을 경우



O outputs





question

&



answer

