

Week07.

비디오 재생

개발환경 구축 절차

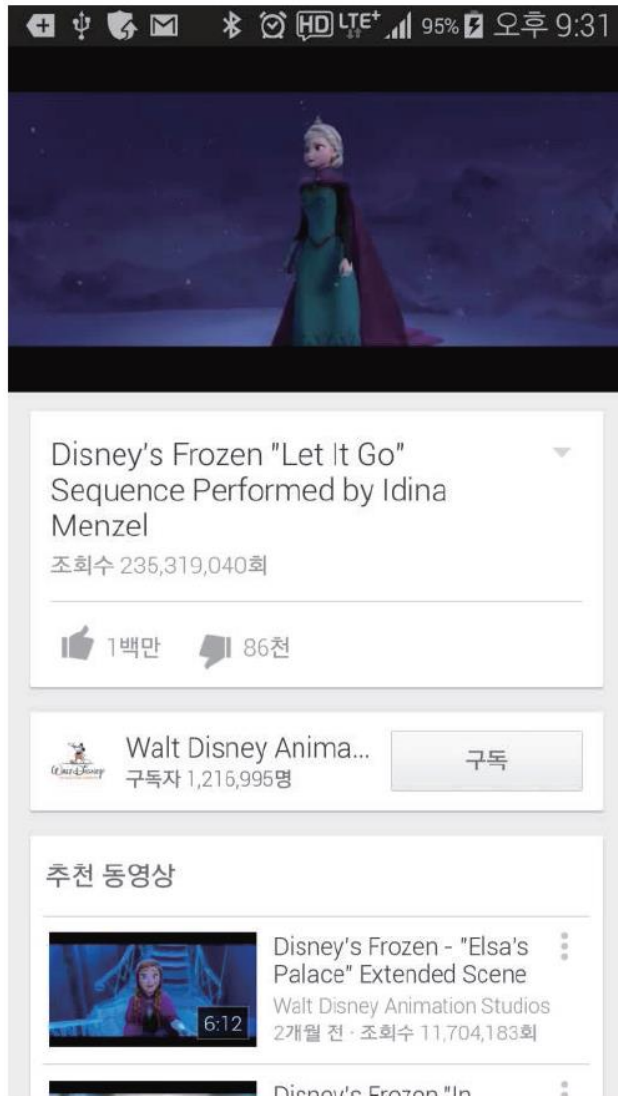
주 차	수 업 내 용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	이미지의 출력
5	이벤트 처리와 액티비티 간 이동
6	오디오 재생
7	비디오 재생
8	중간고사
9	애니메이션
10	사물인터넷과 센서 – 터치 센서, 모션 센서
11	사물인터넷과 센서 – 위치 센서, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활용
14	구글 맵과 위치 추적
15	기말 고사



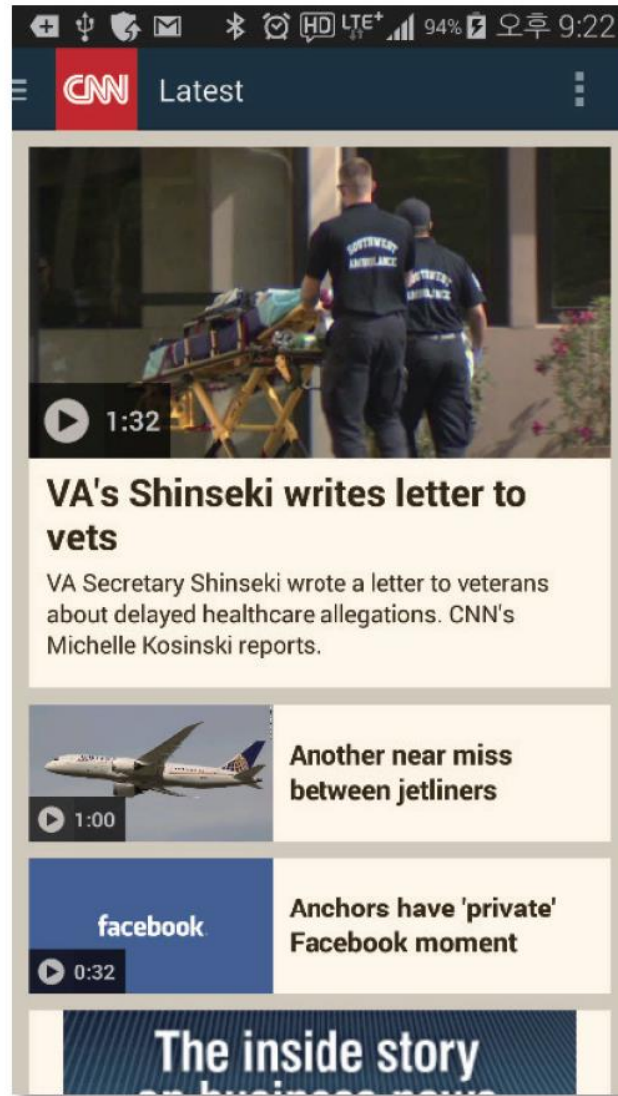
<http://github.com/hopypark>

비디오 재생 앱의 예

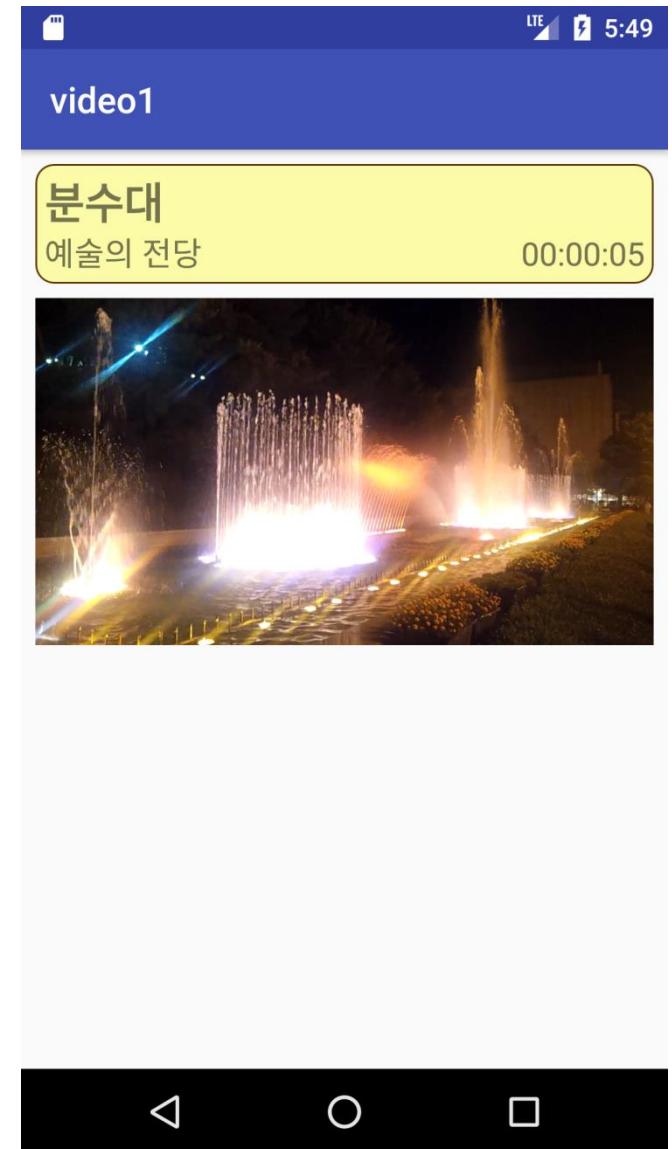
4



(a) YouTube 동영상

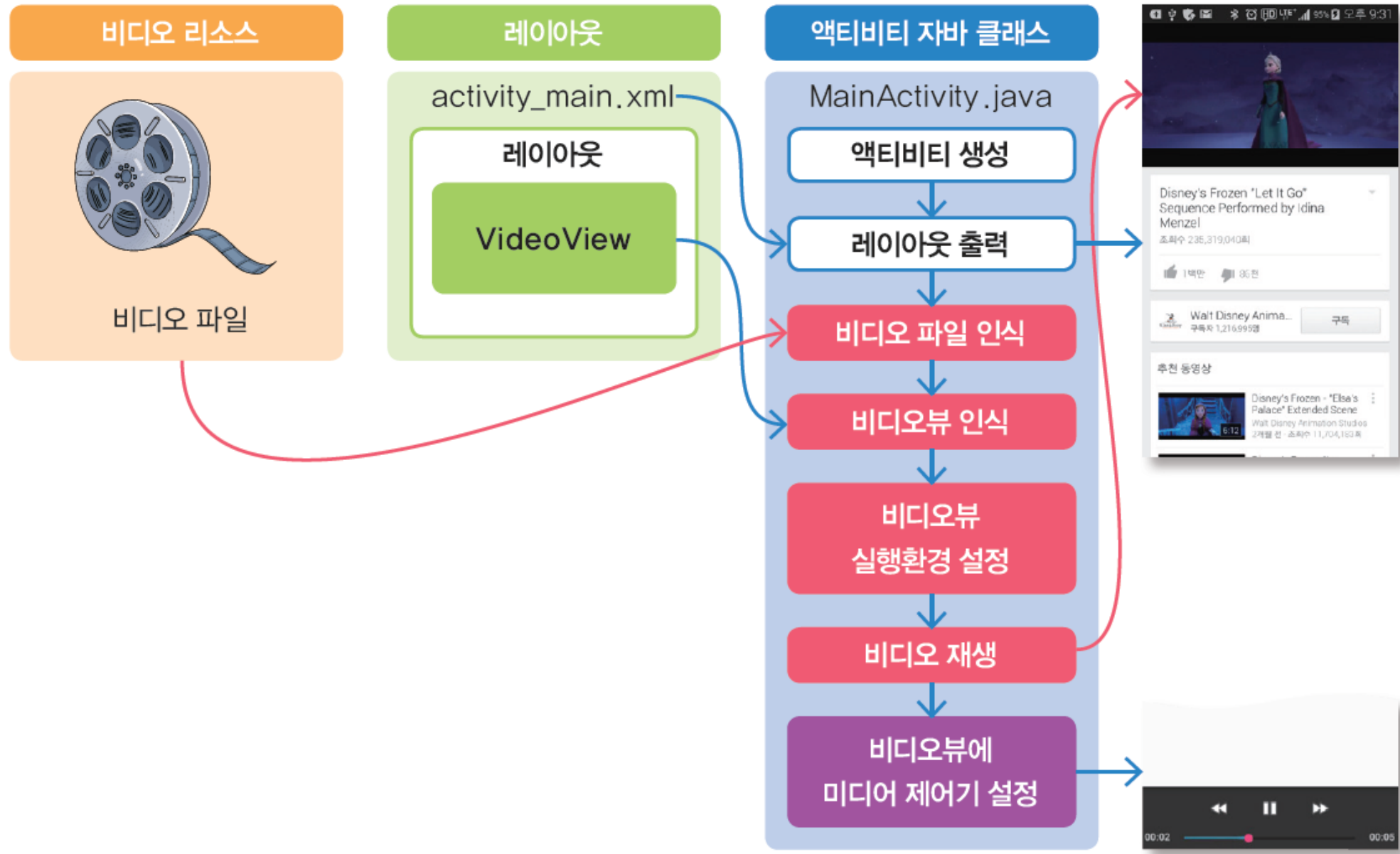


(b) CNN Video



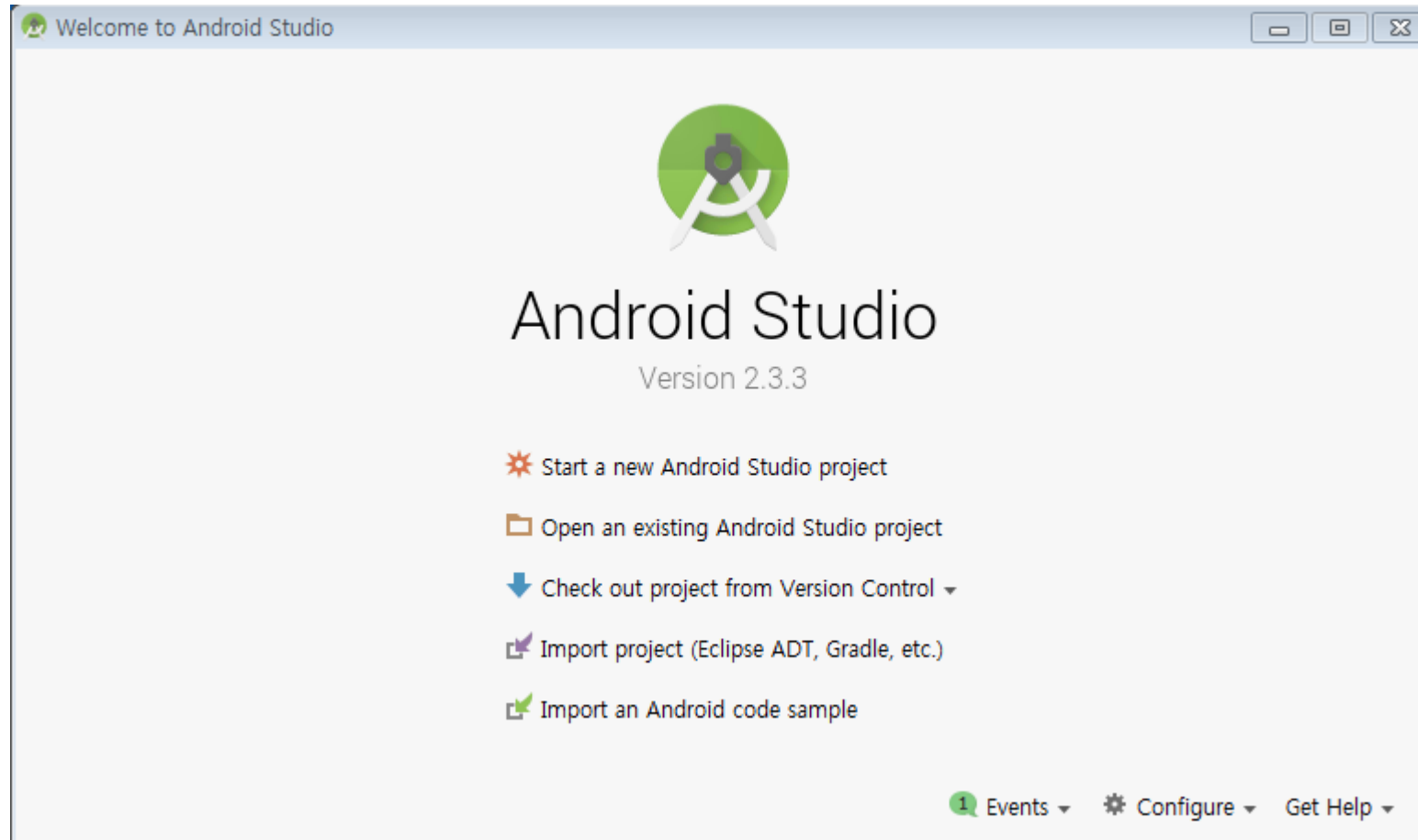
오디오 재생 원리

5



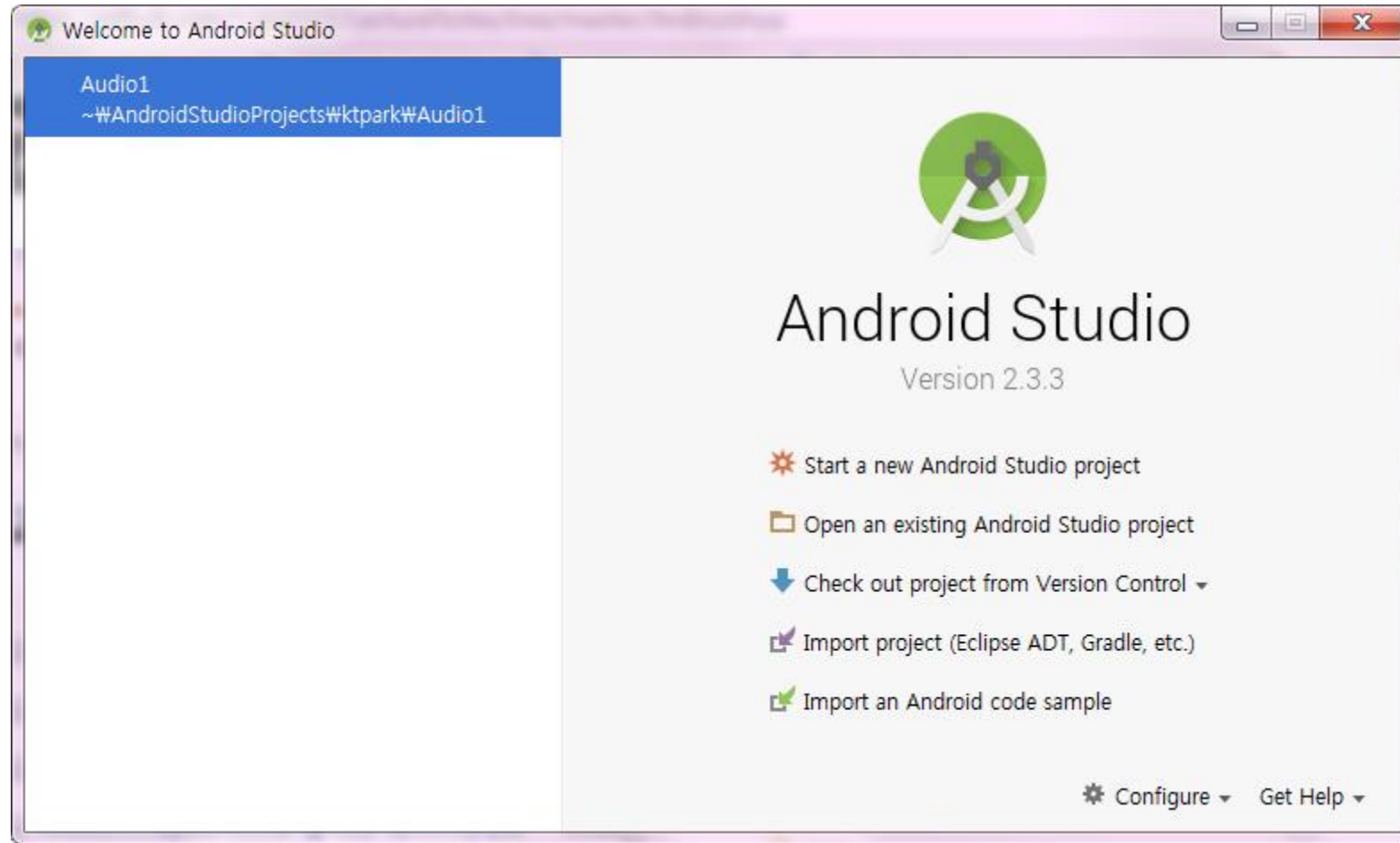
Start a new Android Studio project-type1

6



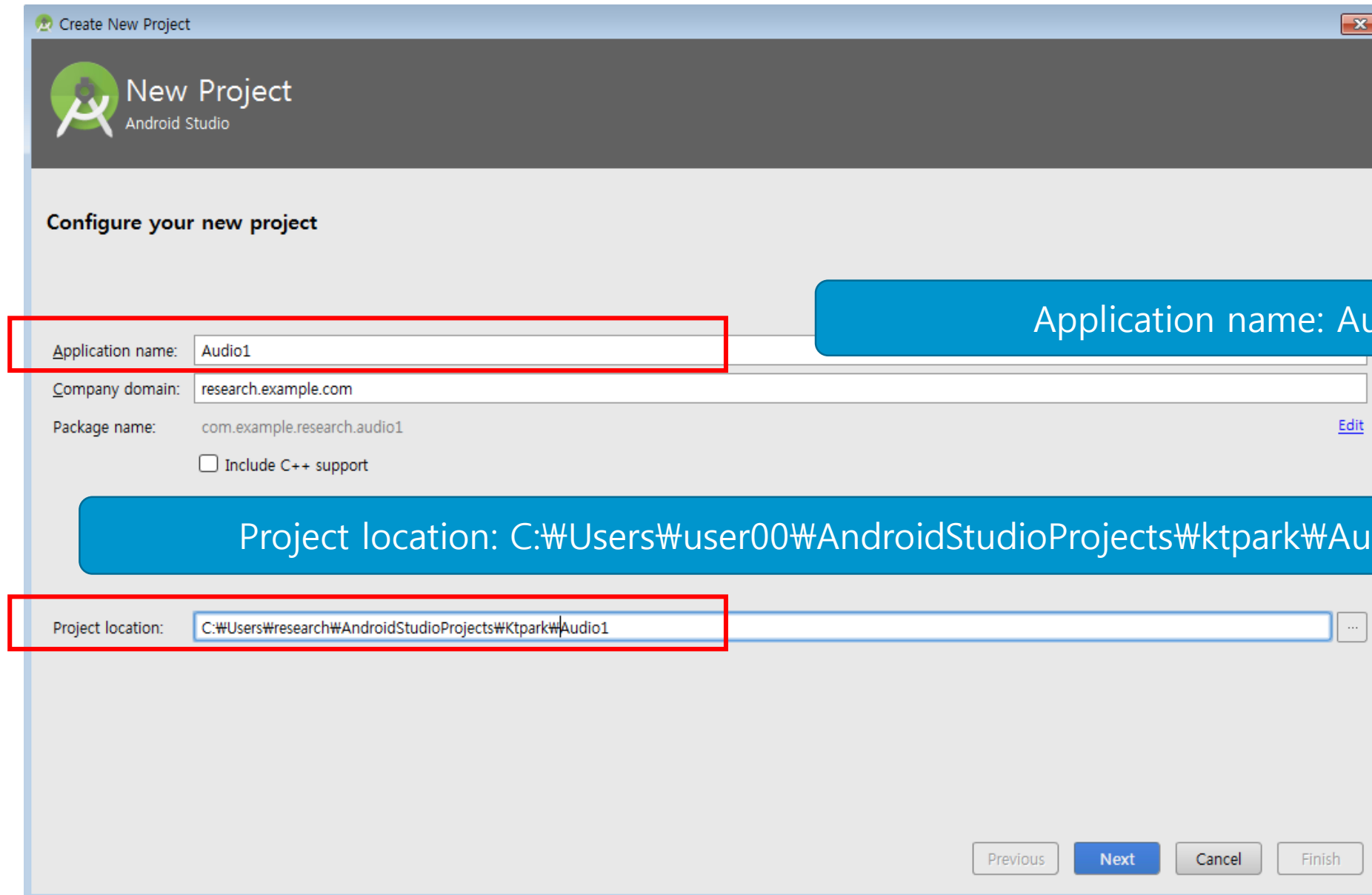
Start a new Android Studio project-type2

7



Configure your new project

8



Create New Project

New Project
Android Studio

Configure your new project

Application name: Audio1

Company domain: research.example.com

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

☐ Include C++ support

Project location: C:\Users\research\AndroidStudioProjects\Wktpark\Audio1

Previous Next Cancel Finish

Configure your new project

9

Application name:

Company domain:

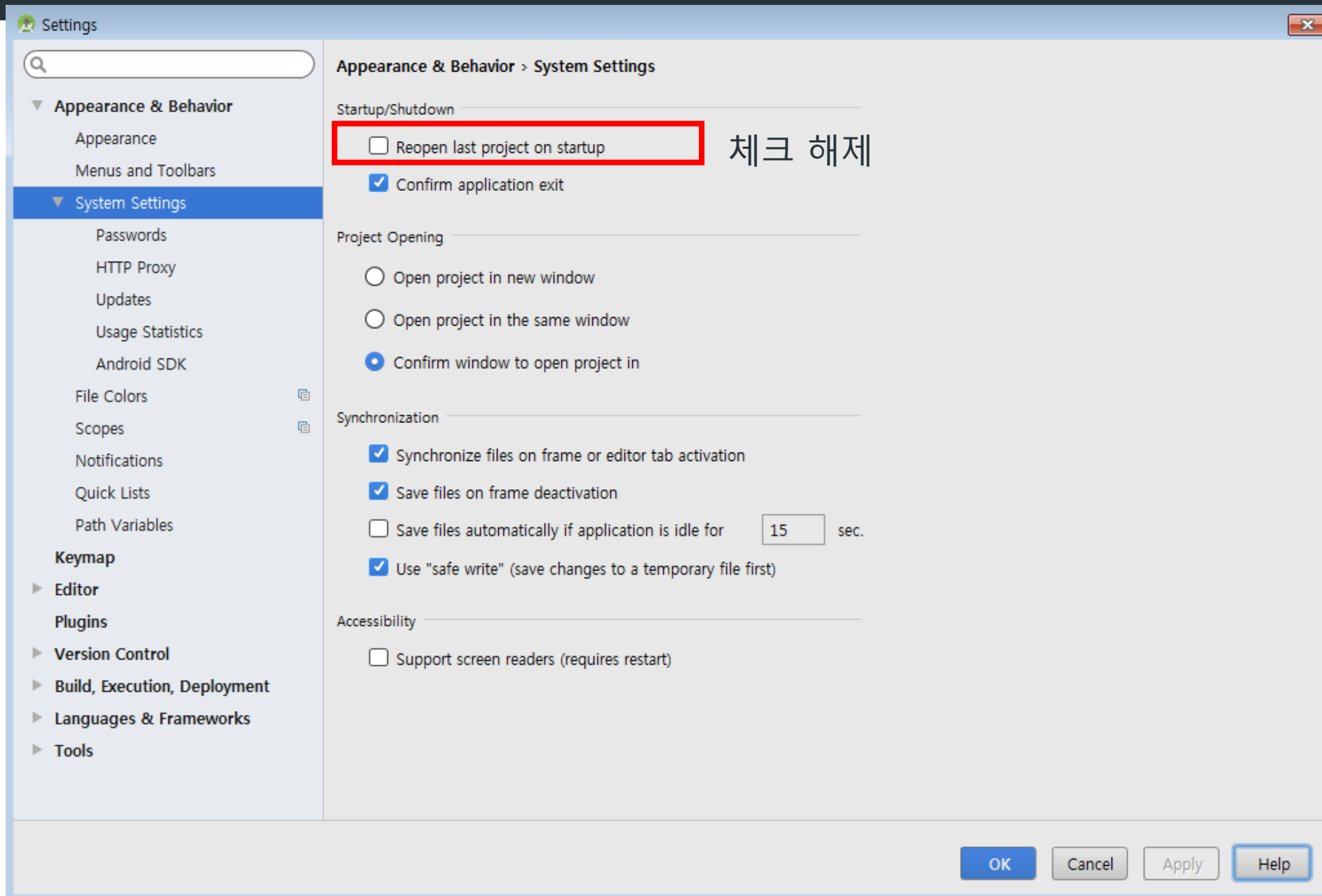
Package name: [Edit](#)

☐ Include C++ support

Project location:

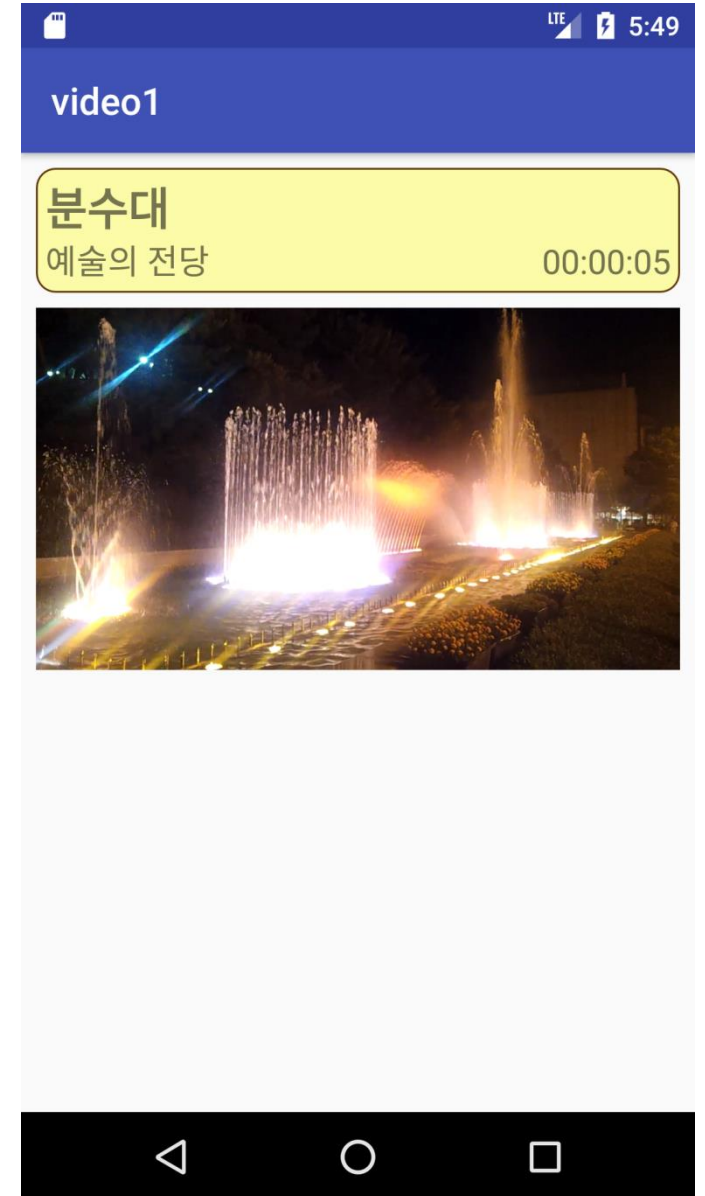
초기 화면 변경-유저가 프로젝트 선택

10



Step 0.프로젝트 개요

11



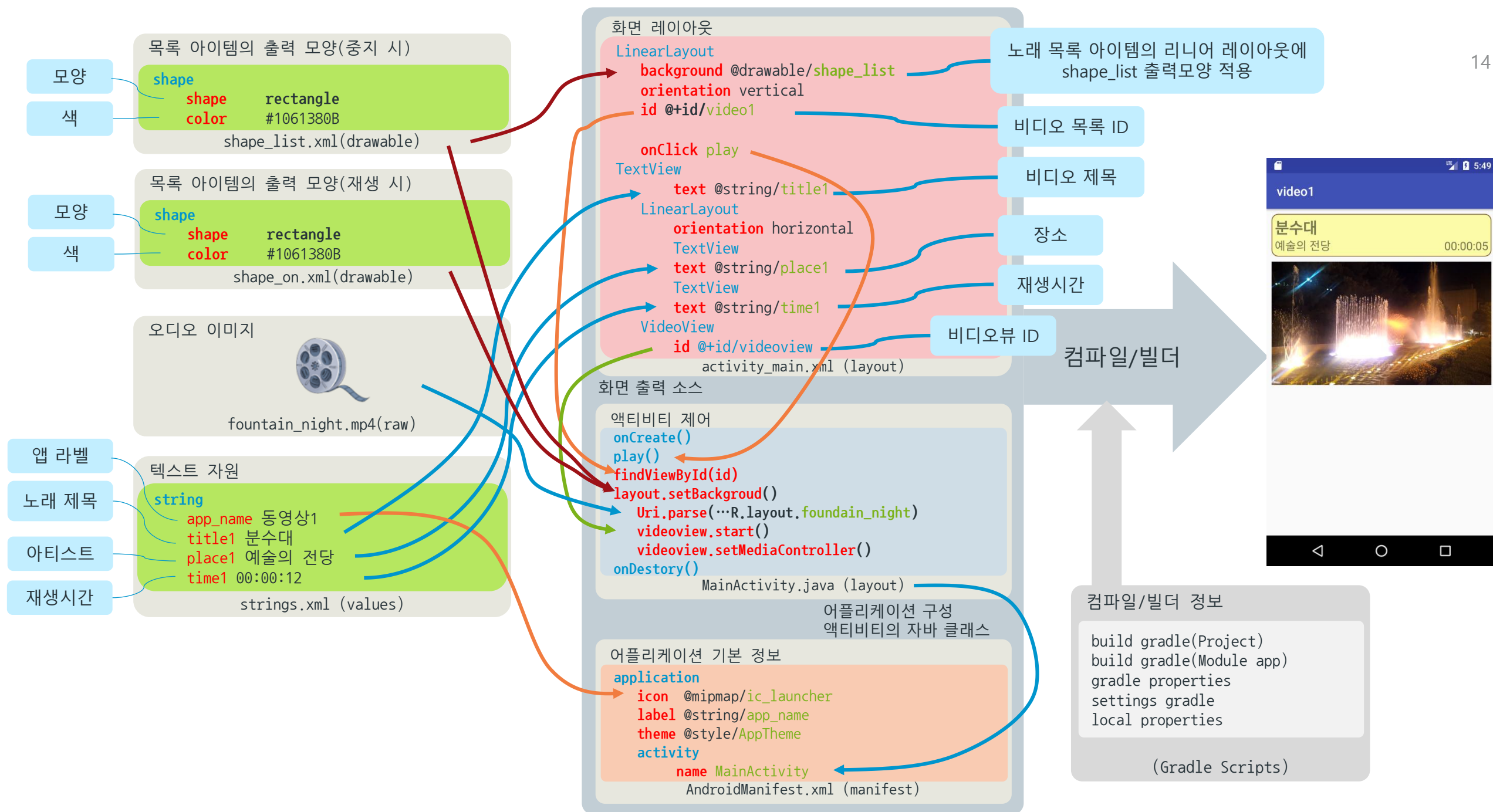
Step 1. 프로젝트 생성

12

절차	내 용
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
③제품 형태	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

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
res	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)
	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

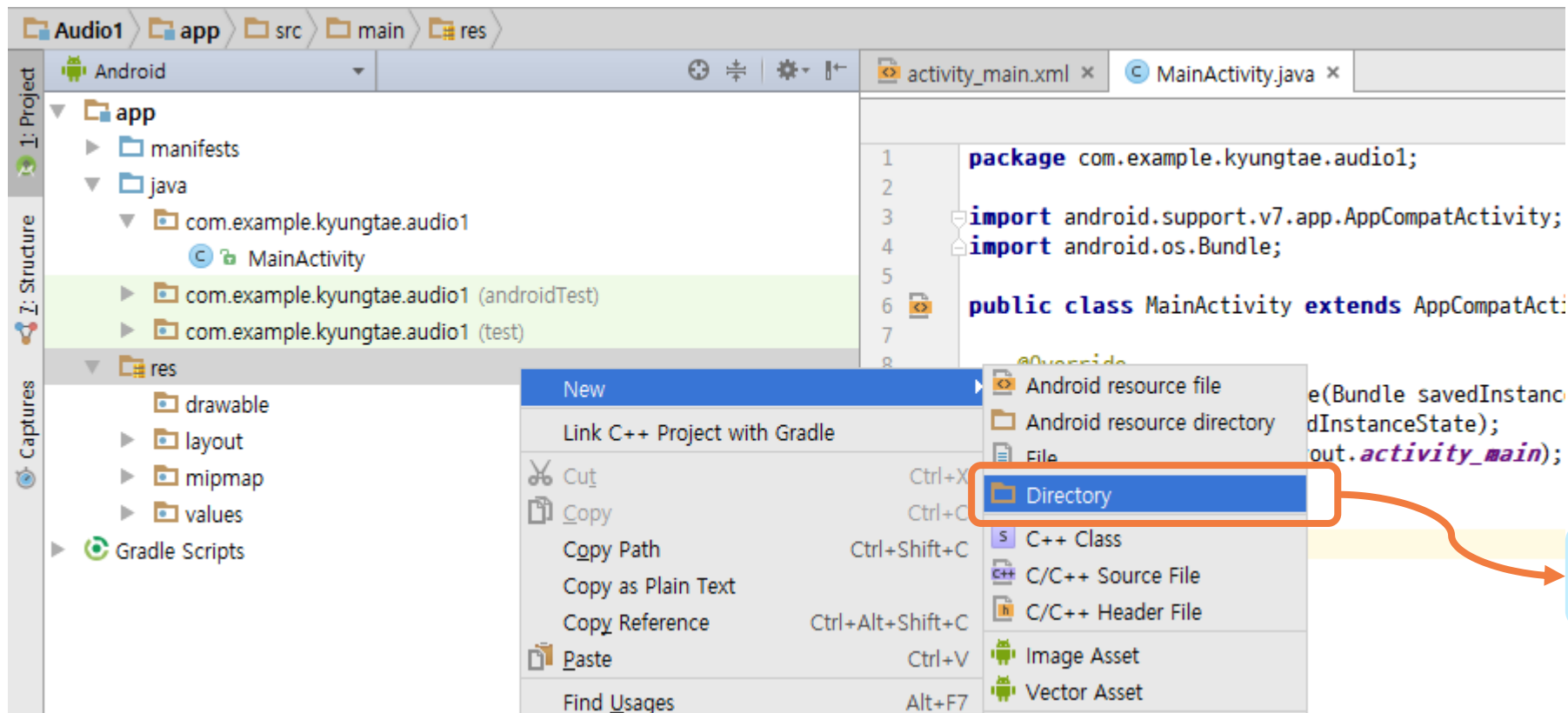
- res 폴더에 있는 **raw** 폴더에 fountain_night.mp4 파일 저장

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

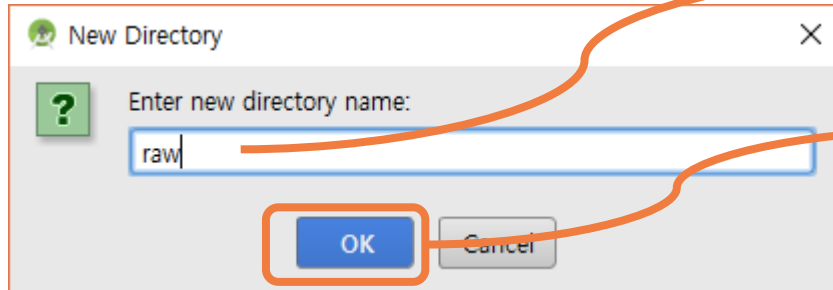
res/raw 폴더에
비디오 파일 올리기

drawable/raw 폴더에 비디오 파일 추가하기

- app→res→New→Directory 클릭



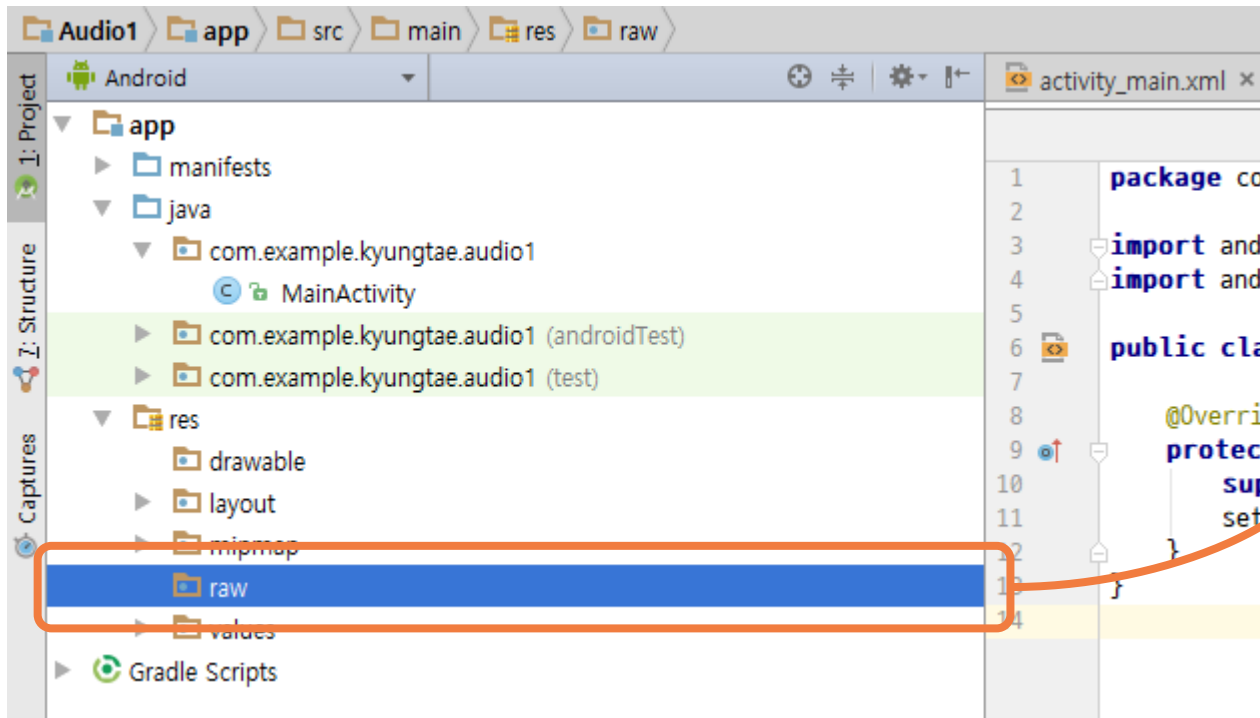
- 폴더 이름 작성



폴더 이름:raw

OK 클릭

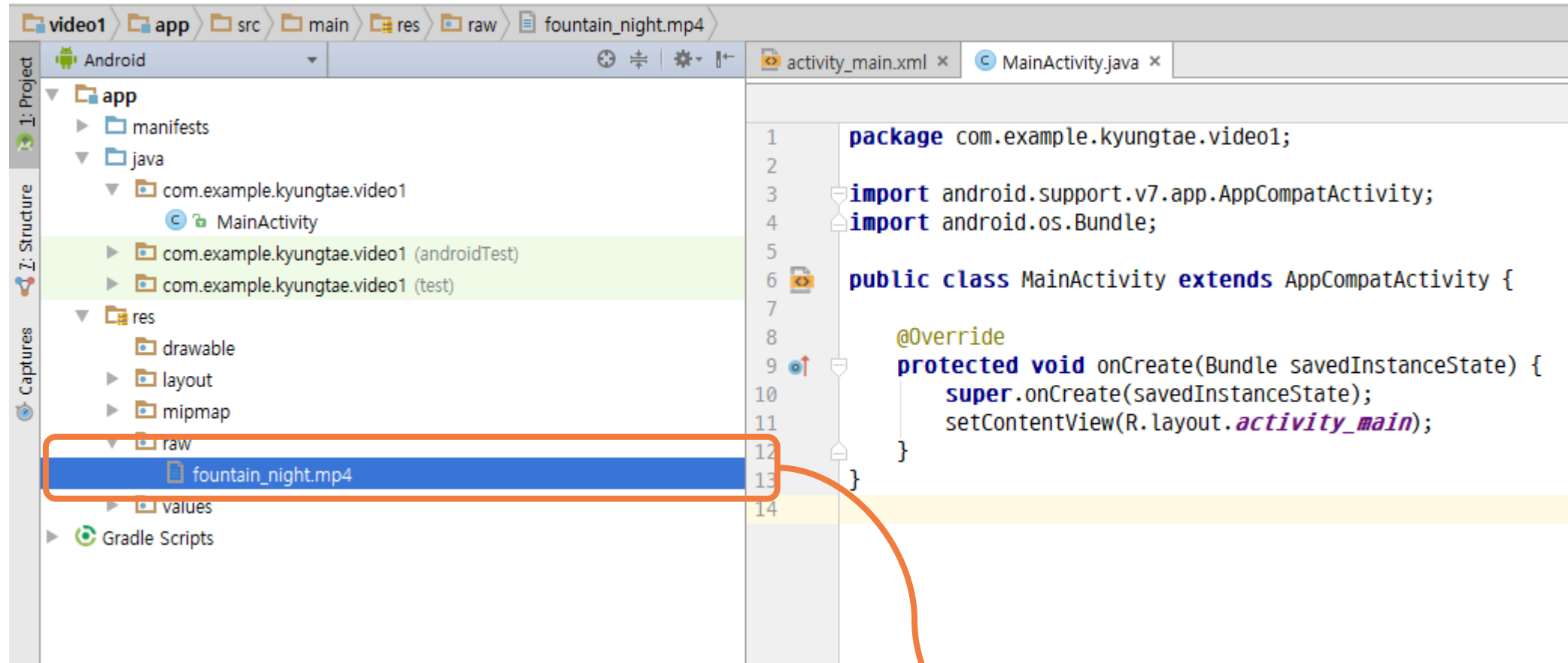
- 실행 결과



raw 폴더 생성됨

- 비디오 파일(fountain_night.mp4) 복사하기

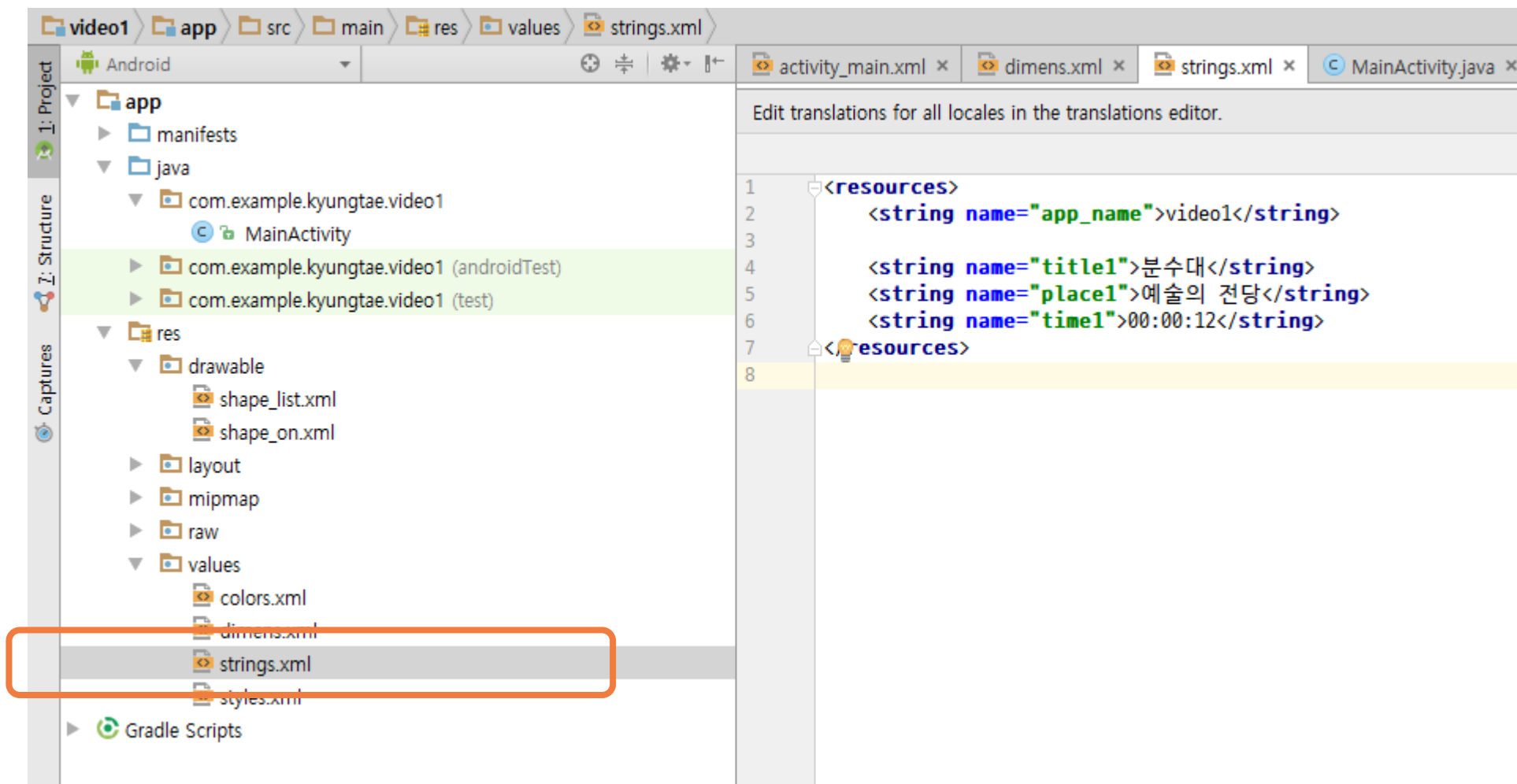
19



fountain_night.mp4 붙여넣기

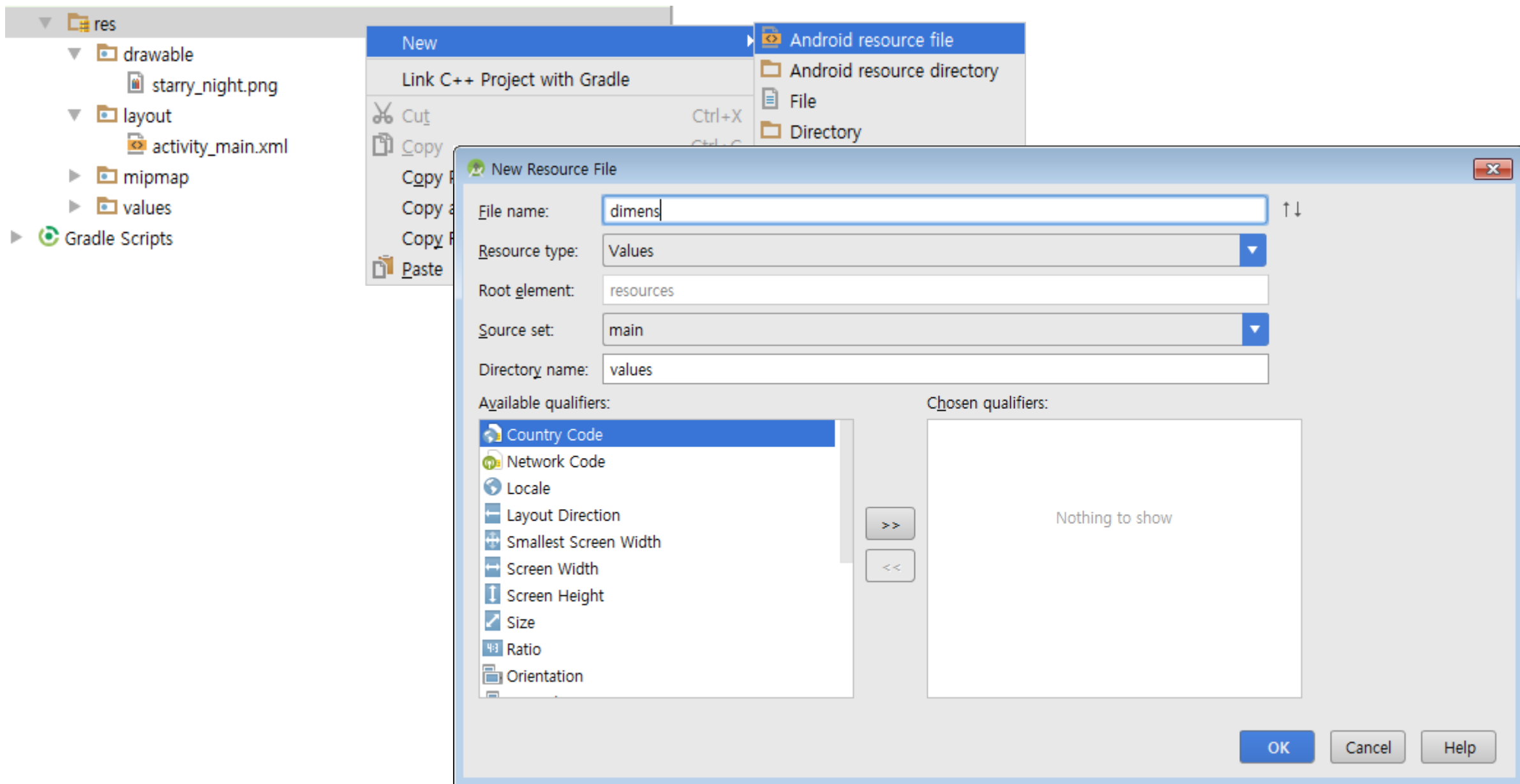
Step 2.2 텍스트 자원의 편집

- strings.xml



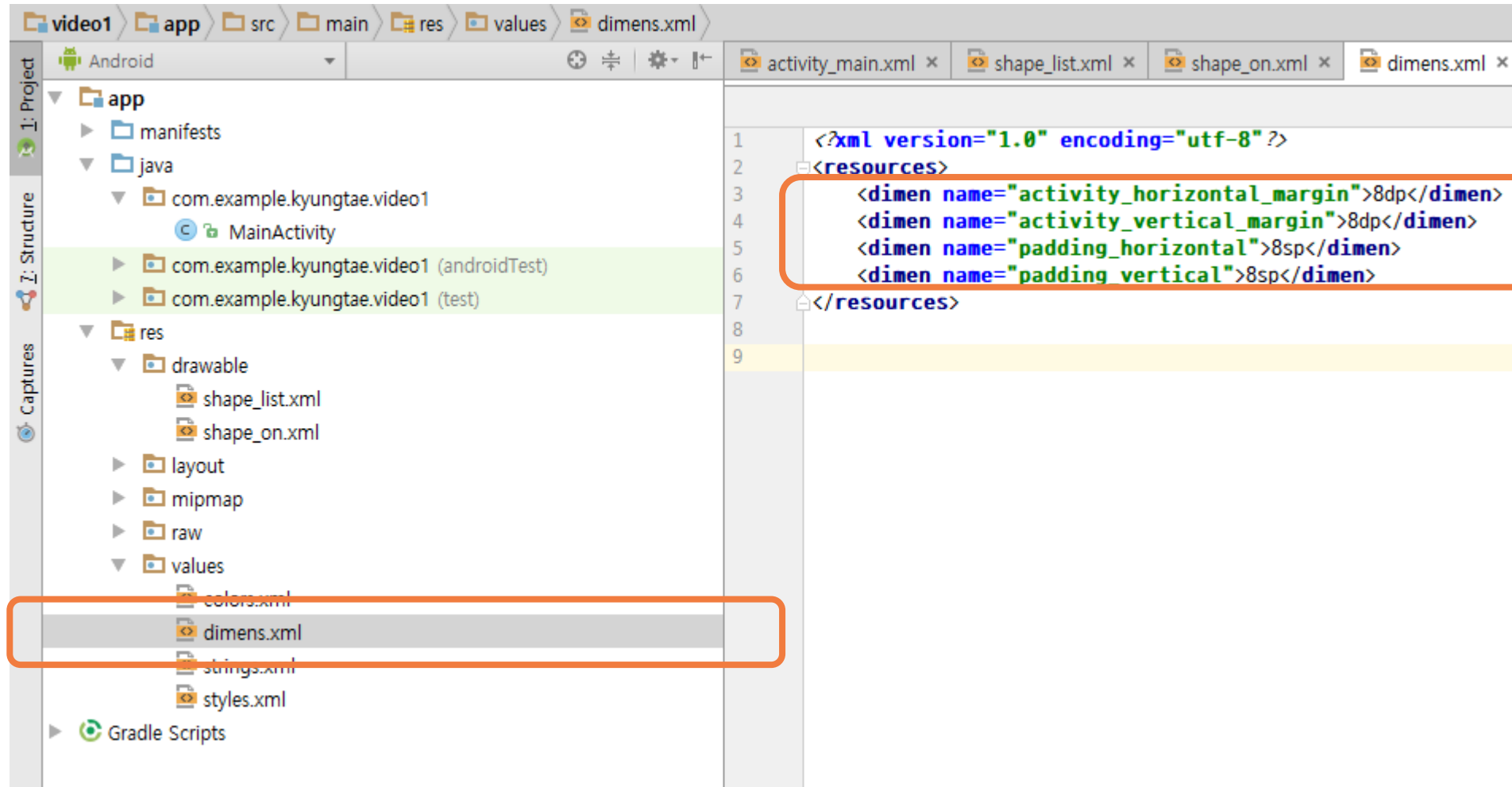
Step 2.2 화면 구성 자원(Dimens.xml)

21



- `dimens.xml`

22



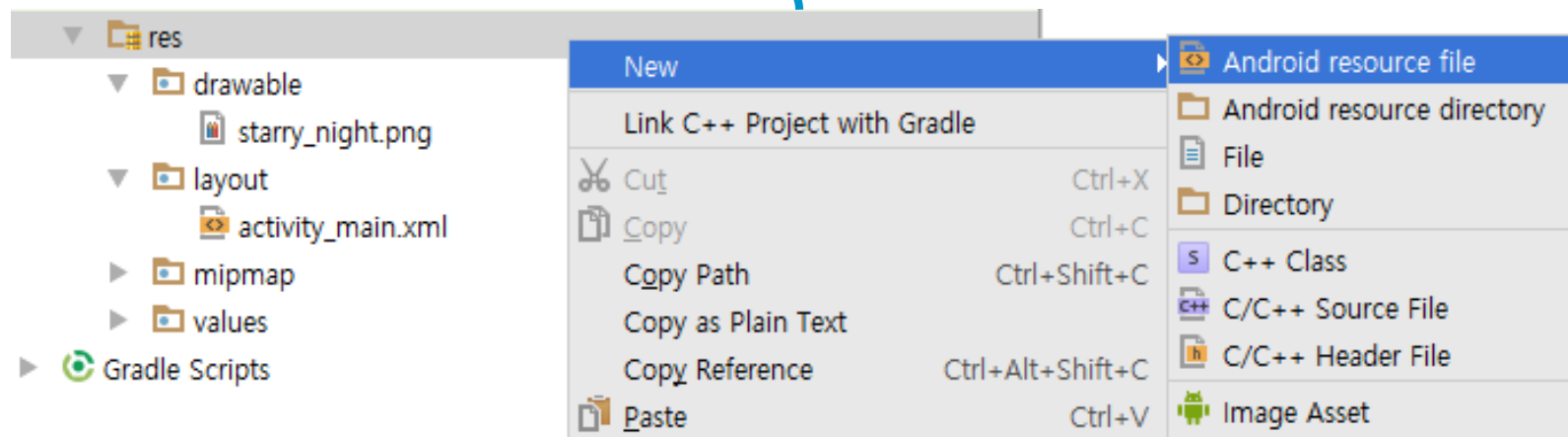
Step 2.3 Drawable Resource 추가 및 편집

23

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

Starry Night

XML 파일 생성



- Set New Resource File

File name: shape_list

Resource type: Drawable

Root element: shape

Source set: main

Directory name: drawable

New Resource File

File name: shape_list

Resource type: Drawable

Root element: shape

Source set: main

Directory name: drawable

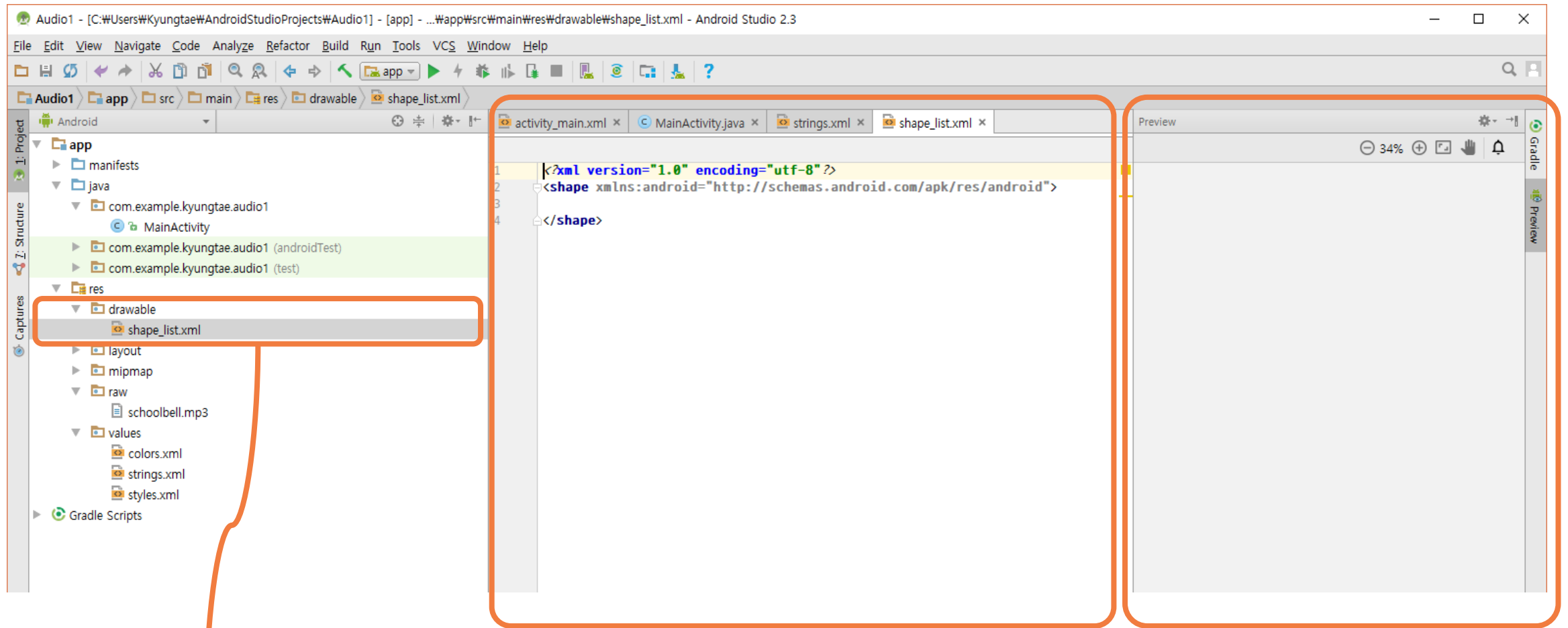
Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation

Chosen qualifiers:

Nothing to show

OK Cancel Help



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)

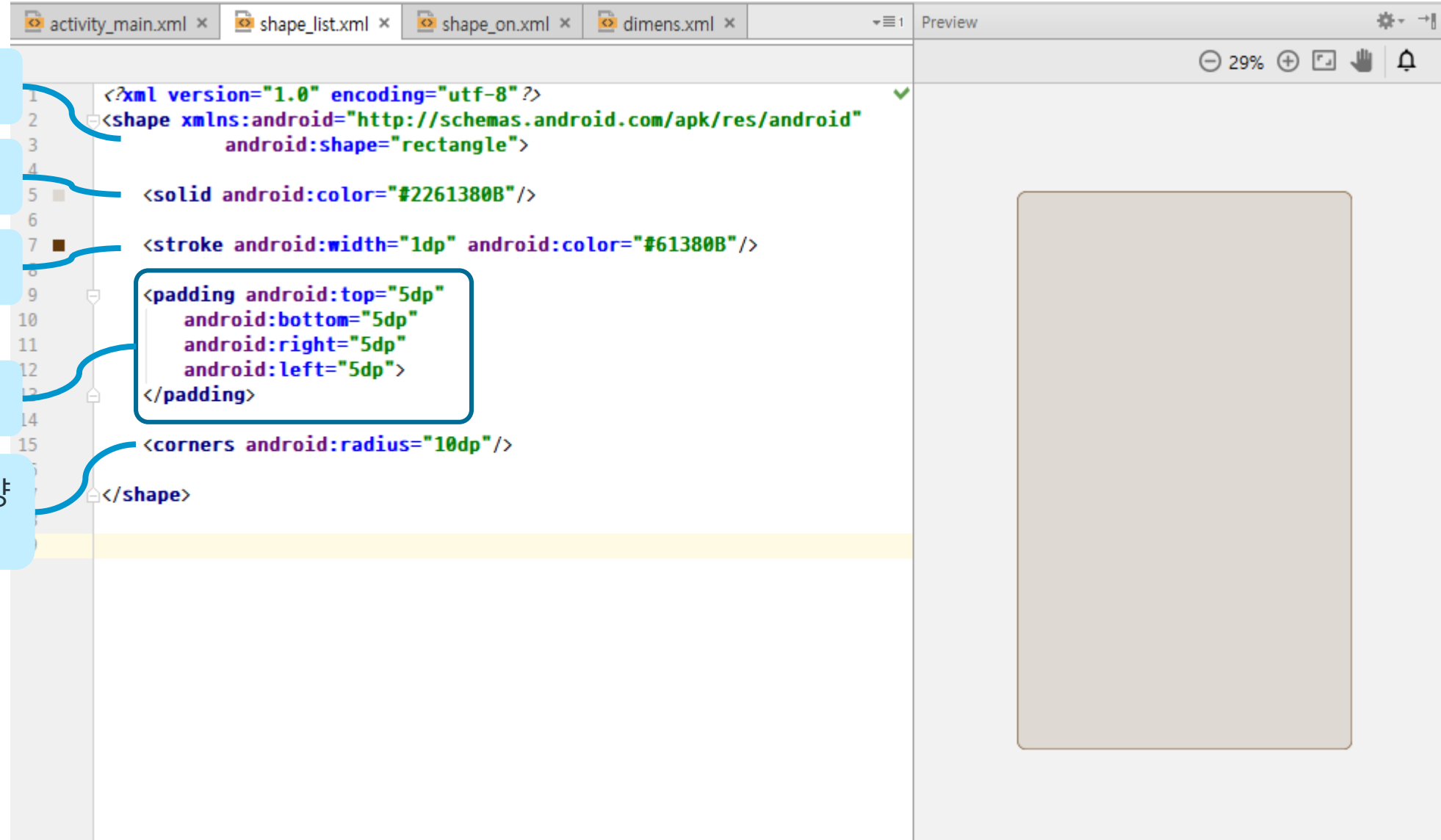
출력모양을 사각형으로 지정

출력모양을 내부의 색

출력모양을 테두리의 색

내부 패딩 정보

출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp)



• shape_on.xml 소스(동영상 재생 시)

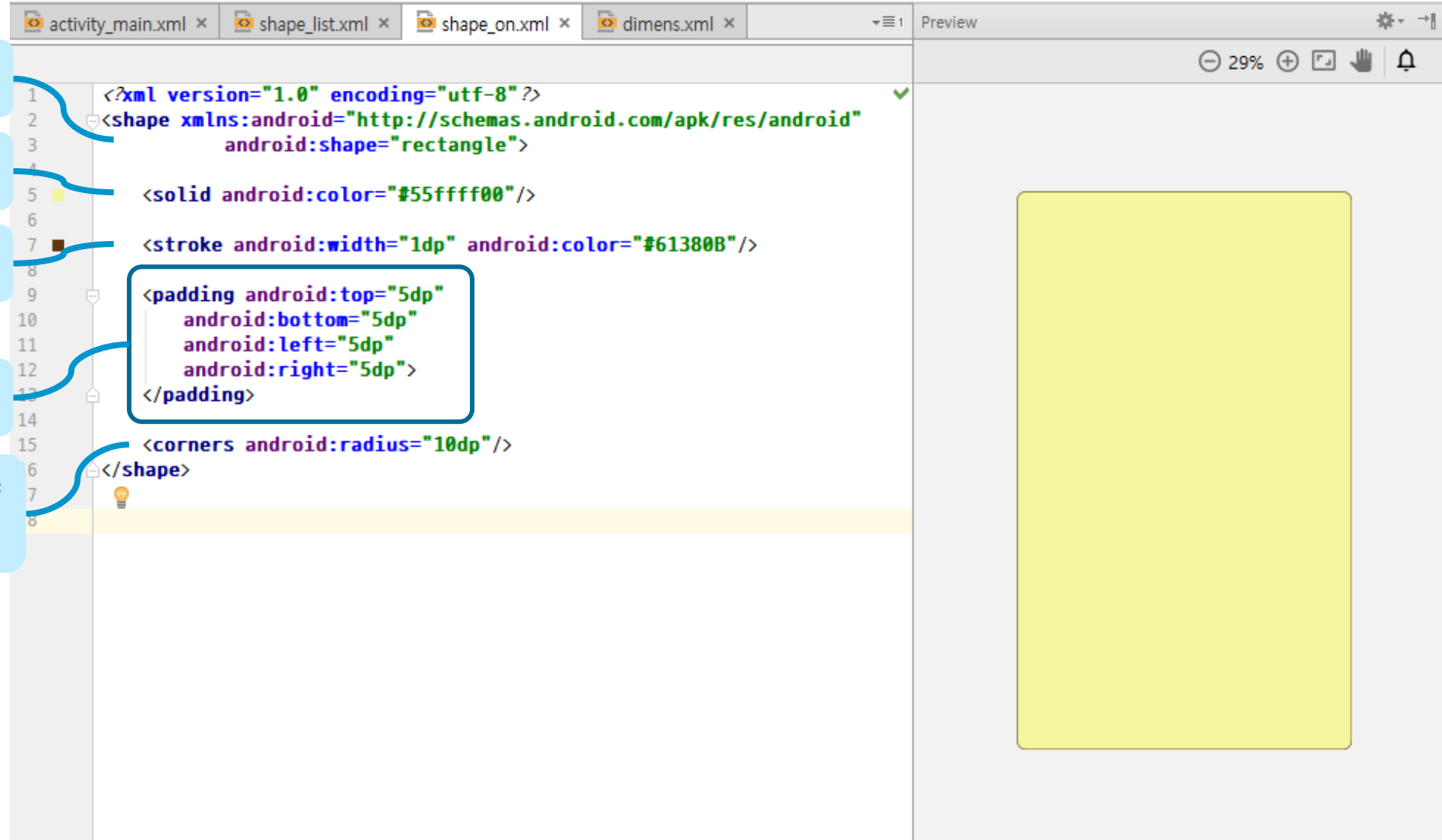
출력모양을 사각형으로 지정

출력모양을 내부의 색

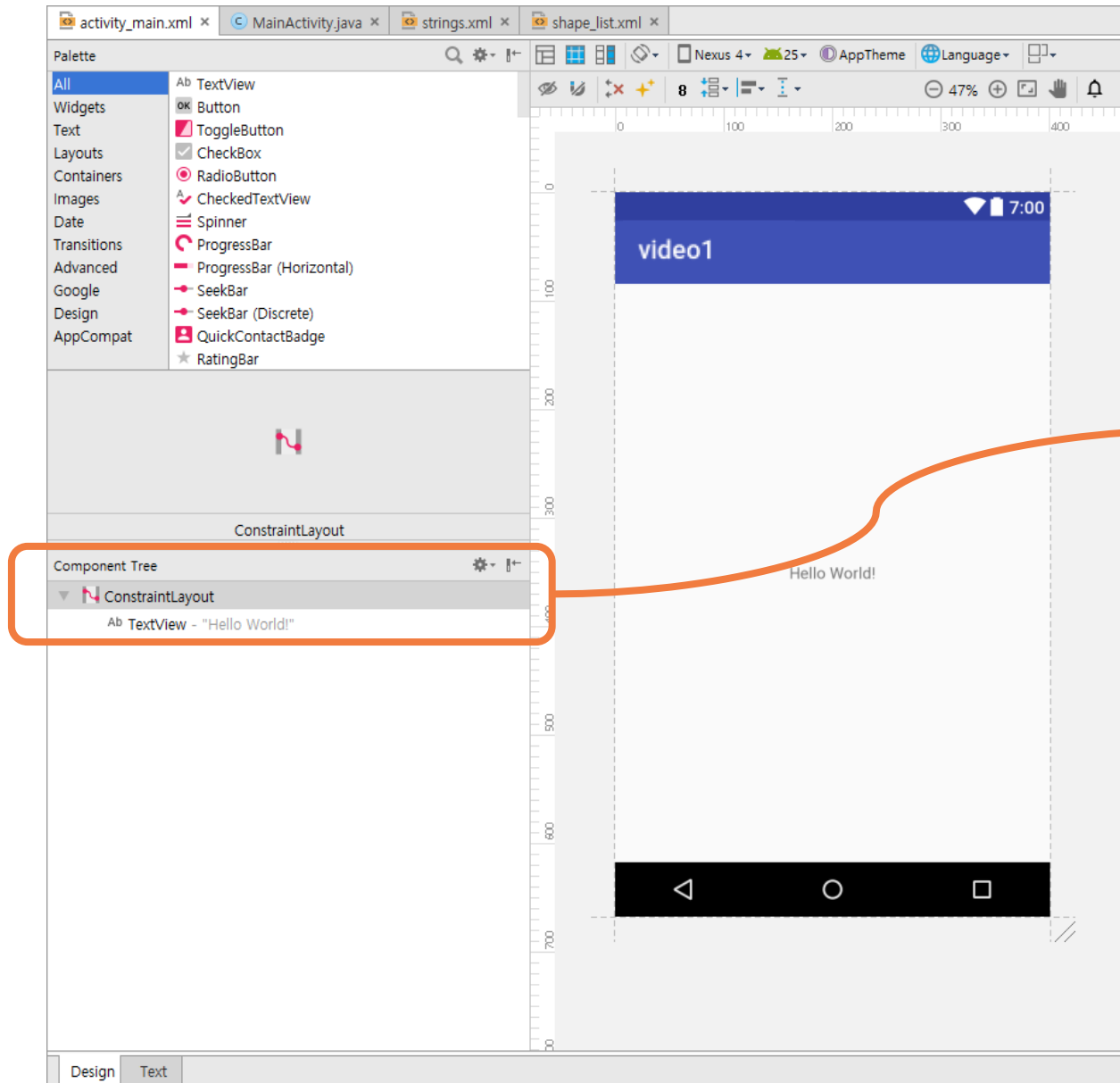
출력모양을 테두리의 색

내부 패딩 정보

출력모양 모서리를 둥근 모양
으로 지정(반지름은 5dp)



2.4 화면 설계



ConstraintLayout →
LinearLayout으로 변경

Text 에디터에서 수정

• Layout 변경 및 기본 TextView 삭제

29

```
activity_main.xml x MainActivity.java x strings.xml x shape_list.xml x
1 android.support.constraint.ConstraintLayout
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
4   xmlns:app="http://schemas.android.com/apk/res-auto"
5   xmlns:tools="http://schemas.android.com/tools"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   tools:context="com.example.kyungtae.audio1.MainActivity">
9
10   <TextView
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:text="Hello World!"
14     app:layout_constraintBottom_toBottomOf="parent"
15     app:layout_constraintLeft_toLeftOf="parent"
16     app:layout_constraintRight_toRightOf="parent"
17     app:layout_constraintTop_toTopOf="parent" />
18 </android.support.constraint.ConstraintLayout>
19
```

```
activity_main.xml x MainActivity.java x strings.xml x shape_list.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context="com.example.kyungtae.audio1.MainActivity">
8
9   <TextView
10     android:layout_width="wrap_content"
11     android:layout_height="wrap_content"
12     android:text="Hello World!"
13     app:layout_constraintBottom_toBottomOf="parent"
14     app:layout_constraintLeft_toLeftOf="parent"
15     app:layout_constraintRight_toRightOf="parent"
16     app:layout_constraintTop_toTopOf="parent" />
17
18 </LinearLayout>
19
```

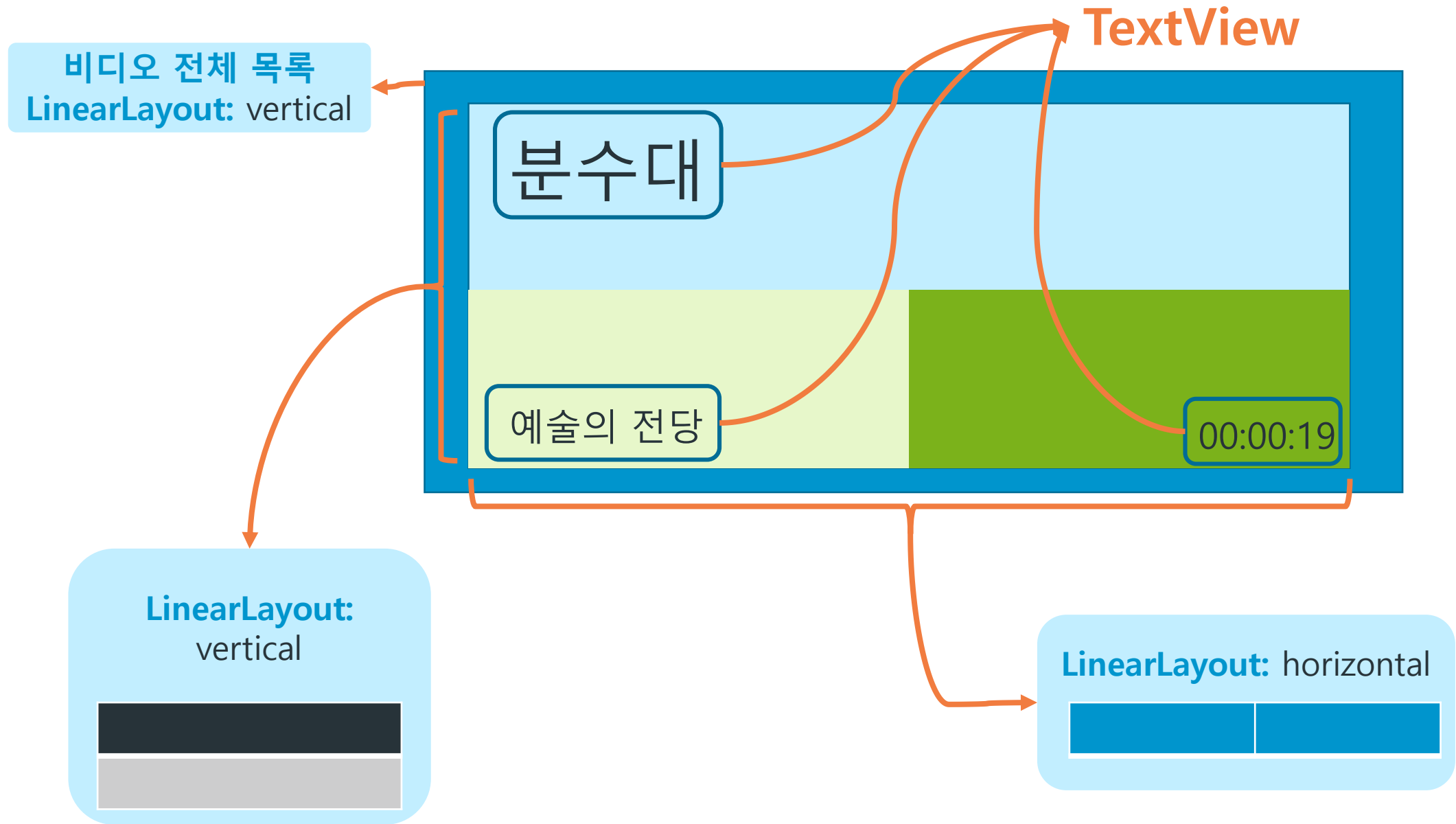
삭제

• 동영상 정보 표시 -

The screenshot displays the Android Studio IDE with the following components:

- Palette:** Lists various UI widgets such as TextView, Button, ToggleButton, CheckBox, RadioButton, CheckedTextView, Spinner, ProgressBar, SeekBar, and QuickContactBadge.
- Component Tree:** Shows the hierarchy of the layout:
 - LinearLayout (vertical)
 - LinearLayout (vertical)
 - TextView - "@string/title1"
 - LinearLayout (horizontal)
 - TextView3 - "@string/place1"
 - TextView2 - "@string/time1"
- Properties Panel:** Displays the properties for the selected video player view. Key properties highlighted include:
 - `layout_width`: match_parent
 - `layout_height`: wrap_content
 - `orientation`: vertical

- 비디오 목록 표시를 위한 Layout 구조



• padding 속성을 이용한 여백 설정

The screenshot displays the Android Studio IDE with the following components:

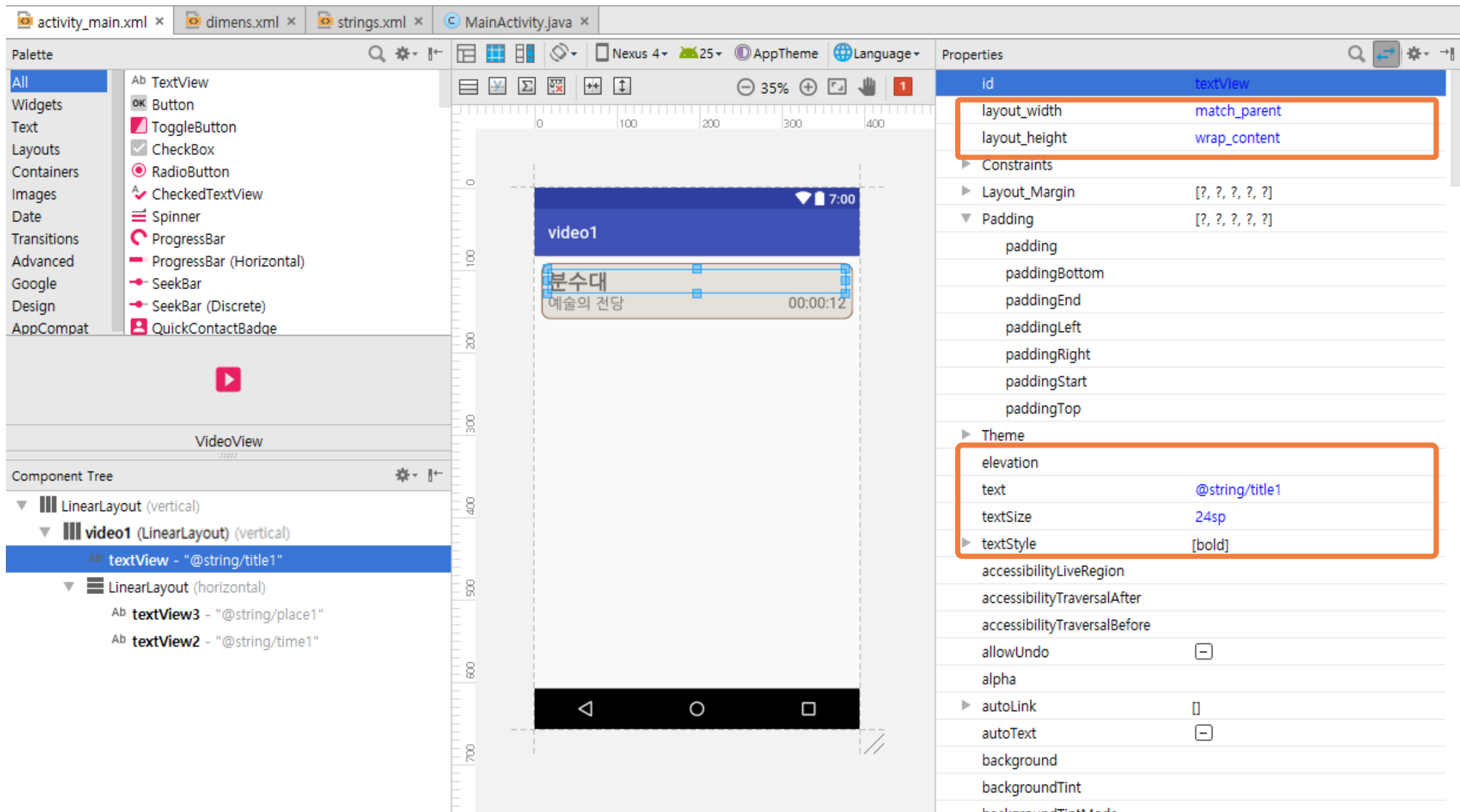
- Palette:** Lists various widgets including TextView, Button, ToggleButton, CheckBox, RadioButton, CheckedTextView, Spinner, ProgressBar, SeekBar, and QuickContactBadge.
- Component Tree:** Shows a vertical LinearLayout containing another vertical LinearLayout, which in turn contains a horizontal LinearLayout with three TextViews: title1, place1, and time1.
- Design View:** A visual representation of the video player interface on a Nexus 4 device. It features a blue header labeled 'video1', a video player area with the text '분수대 예술의 전당' and a progress bar, and a black navigation bar at the bottom.
- Properties Panel:** Displays the properties for the selected video player view. The 'Padding' property is highlighted with an orange box and set to [?, 8dp, 8dp, 8dp, 8dp]. A blue callout box points to the 'padding' property with the text 'padding: dimens 리소스사용'.

• 비디오 제목 레이아웃 속성 설정(클릭시 동영상 id: video1)

The screenshot displays the Android Studio IDE with the following components:

- Palettes:** The left palette shows various widgets like TextView, Button, ToggleButton, etc. The right palette shows the properties for the selected 'video1' view.
- Component Tree:** The bottom-left pane shows the hierarchy: `LinearLayout (vertical)` containing `video1 (LinearLayout) (vertical)`, which contains `textView - "@string/title1"`, `LinearLayout (horizontal)` containing `textView3 - "@string/place1"` and `textView2 - "@string/time1"`.
- Design View:** The central canvas shows a video player layout on a Nexus 4 device. The video player has a blue header bar with a play button icon and a title bar with the text '분수대 예술의 전당' and a duration '00:00:12'. The video player is labeled 'video1'.
- Properties Panel:** The right pane shows the properties for the 'video1' view. The 'id' property is highlighted with a blue box and an orange arrow pointing to the text 'id: video1' in the header. The 'clickable' property is checked, and the 'onClick' property is set to 'play'.

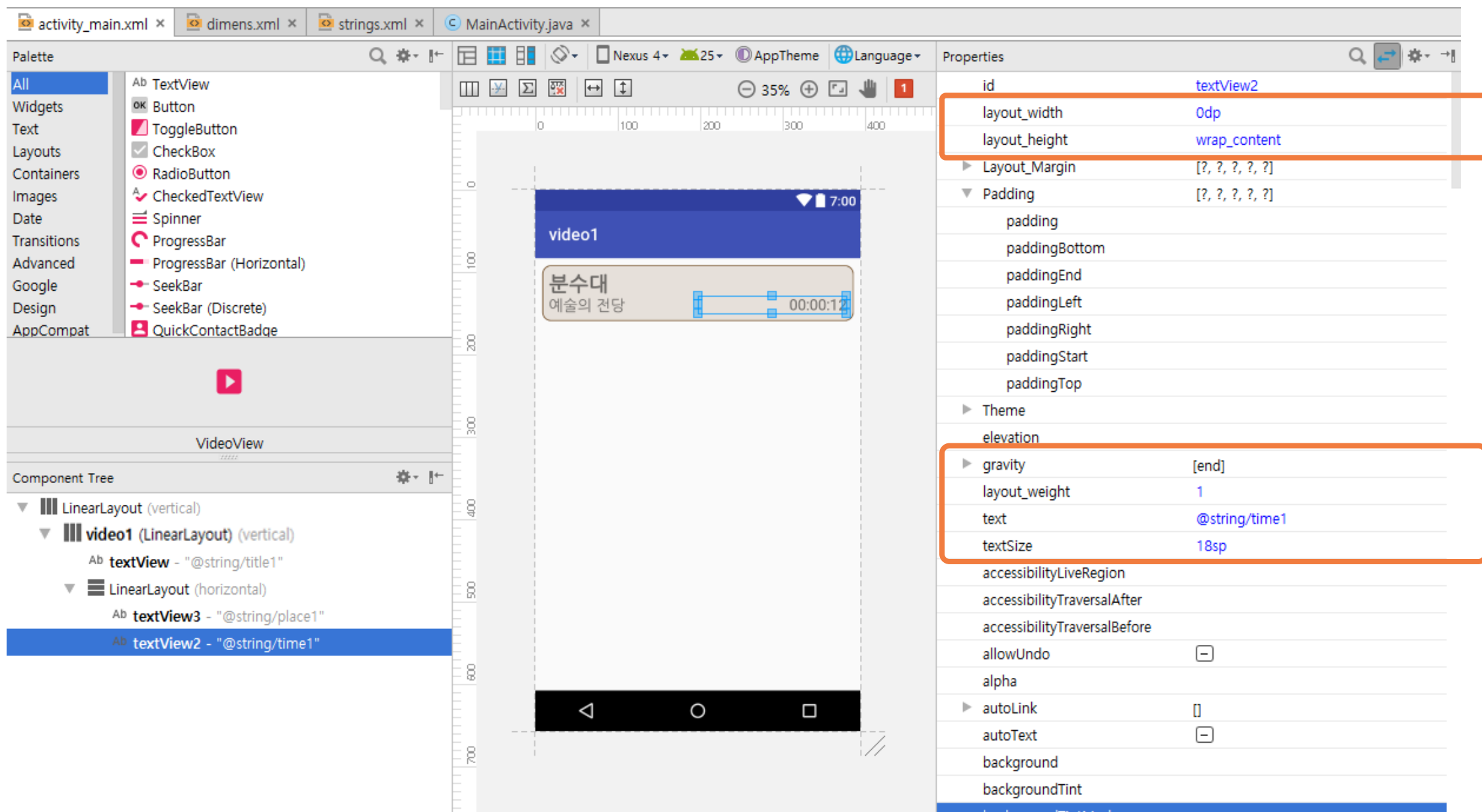
Property	Value
id	video1
layout_width	match_parent
layout_height	match_parent
Layout_Margin	[?, ?, ?, ?]
Padding	[?, 8sp, 8sp, 8sp, 8sp]
Theme	
elevation	
background	@drawable/shape_list
clickable	<input checked="" type="checkbox"/>
onClick	play
orientation	vertical
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
addStatesFromChildren	<input type="checkbox"/>
alpha	
alwaysDrawnWithCache	<input type="checkbox"/>
animateLayoutChanges	<input type="checkbox"/>
animationCache	<input type="checkbox"/>
backgroundTint	
backgroundTintMode	
baselineAligned	<input type="checkbox"/>
baselineAlignedChildIndex	
clipChildren	<input type="checkbox"/>
clipToPadding	<input type="checkbox"/>
contentDescription	
contextClickable	<input type="checkbox"/>



• 비디오 설명(place) Layout

The screenshot displays the Android Studio IDE with the following components:

- Palettes:** A list of widgets on the left, including TextView, Button, ToggleButton, CheckBox, RadioButton, CheckedTextView, Spinner, ProgressBar, SeekBar, and QuickContactBadge.
- Component Tree:** A hierarchical view of the layout structure at the bottom left, showing a vertical LinearLayout containing a video1 (LinearLayout) which includes a TextView (textView) and another vertical LinearLayout containing a TextView3 (textView3) and a TextView2 (textView2).
- Design View:** A central visual representation of the layout on a Nexus 4 device. It shows a video player interface with a blue header labeled 'video1', a video player control bar with Korean text '분수대 예술의 전당' and a duration of '00:00:12', and a black navigation bar at the bottom.
- Properties Panel:** A panel on the right showing the attributes for the selected 'textView3' widget. Two sections are highlighted with orange boxes:
 - Layout Attributes:** layout_width is set to '0dp' and layout_height is set to 'wrap_content'.
 - Text Attributes:** gravity is set to '[start]', layout_weight is set to '1', text is set to '@string/place1', and textSize is set to '18sp'.



• 비디오

id: videoview

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

- PaLETTE:** A list of widgets including ImageButton, ImageView, and VideoView. VideoView is selected.
- Component Tree:** A hierarchical view of the UI components. It shows a LinearLayout (vertical) containing a video1 (LinearLayout) and a videoview. video1 contains a textView and another LinearLayout (horizontal) containing textView3 and textView2.
- Design View:** A visual representation of the UI. It shows a video player with a title '분수대 예술의 전당' and a duration '00:00:12'. The video player is a VideoView widget.
- Properties Panel:** A list of attributes for the selected VideoView widget. The attributes are: id (videoview), layout_width (match_parent), layout_height (wrap_content), Layout_Margin ([?, ?, 8dp, ?, ?]), padding ([?, ?, ?, ?, ?]), Theme, elevation, visibility (invisible), accessibilityLiveRegion, accessibilityTraversalAfter, accessibilityTraversalBefore, alpha, background, backgroundTint, backgroundTintMode, clickable, contentDescription, contextClickable, drawingCacheQuality, duplicateParentState, and focusable.

2.5 Activity 제어(MainActivity.java)

- 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;
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);
22     }
23
24 }
```

비디오뷰 객체 생성(null)

• 비디오 제목을 클릭했을 때 호출되는 메소드(play()) 추가

40

```
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
```

재생 중 일 때

미디어 플레이어 중지

새로운 Drawable 객체 인식해서 선택한 비디오의 배경으로 설정

정지 일 때

uri위치의 동영상재생

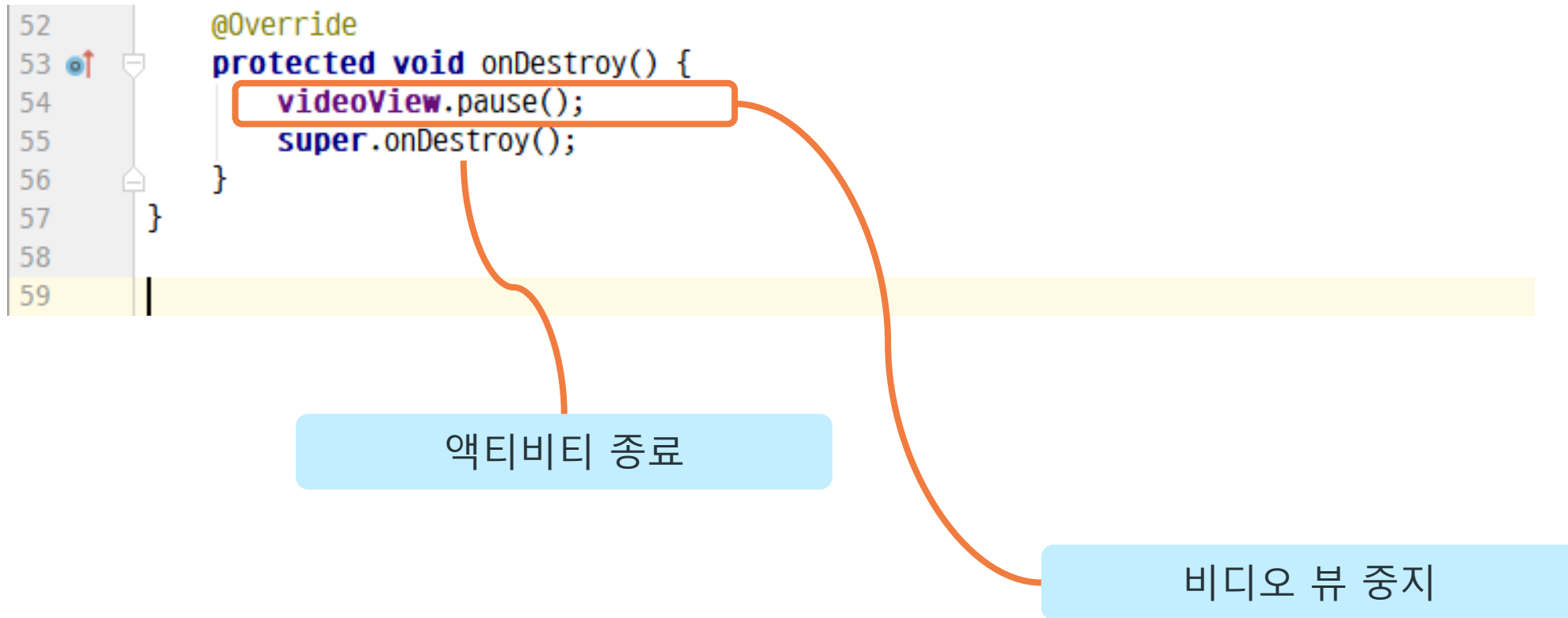
동영상 파일의 uri 인식

새로운 Drawable 객체 인식해서 선택한 비디오의 배경으로 설정

미디어 제어기 생성

비디오뷰의 미디어 제어기로 설정

- 프로젝트 리소스를 얻기 위한 함수 추가



클래스와 속성/메소드

- 클래스

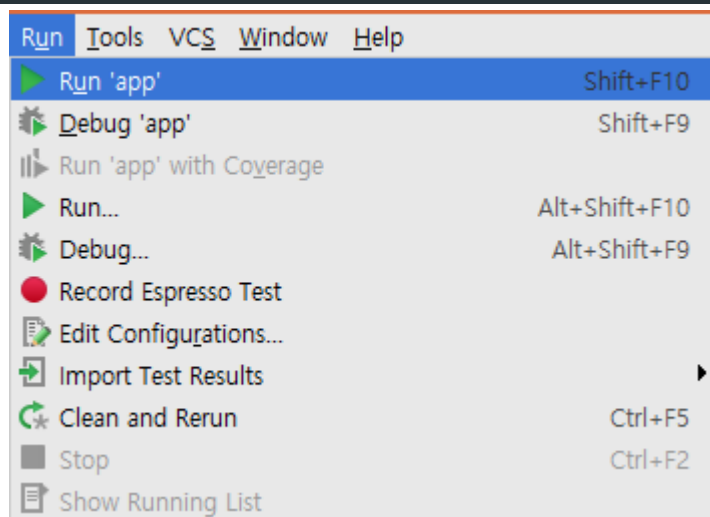
클래스	설명
MediaController	미디어 실행 제어를 포함하는 뷰

- 메소드

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

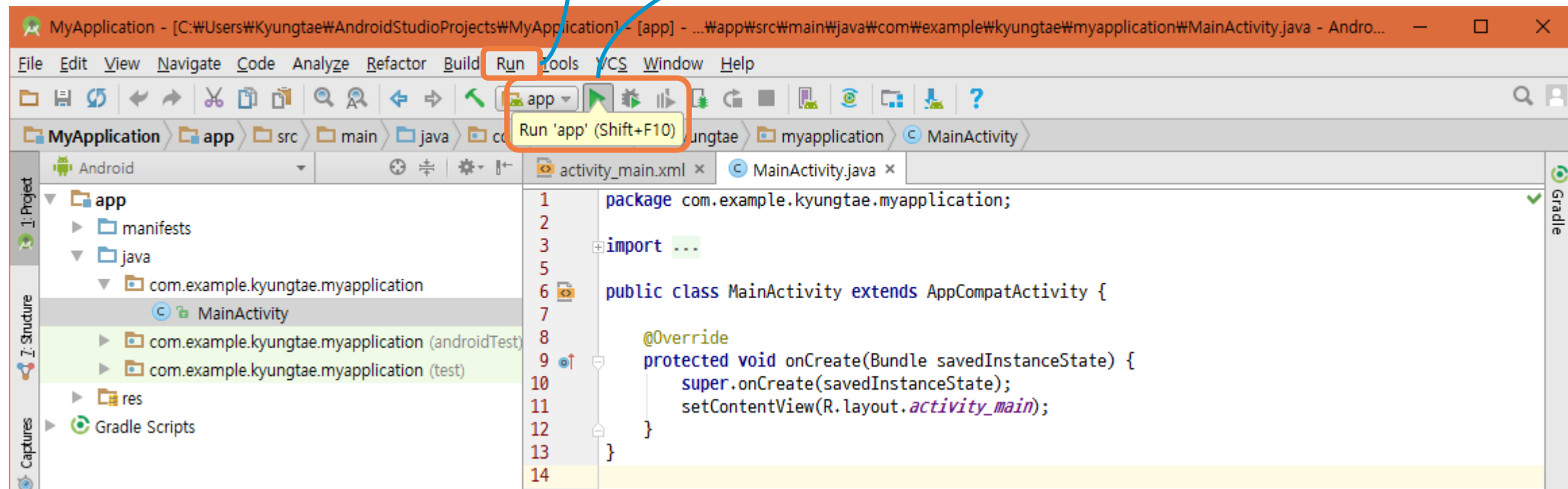
Step 3. 프로젝트 실행

43

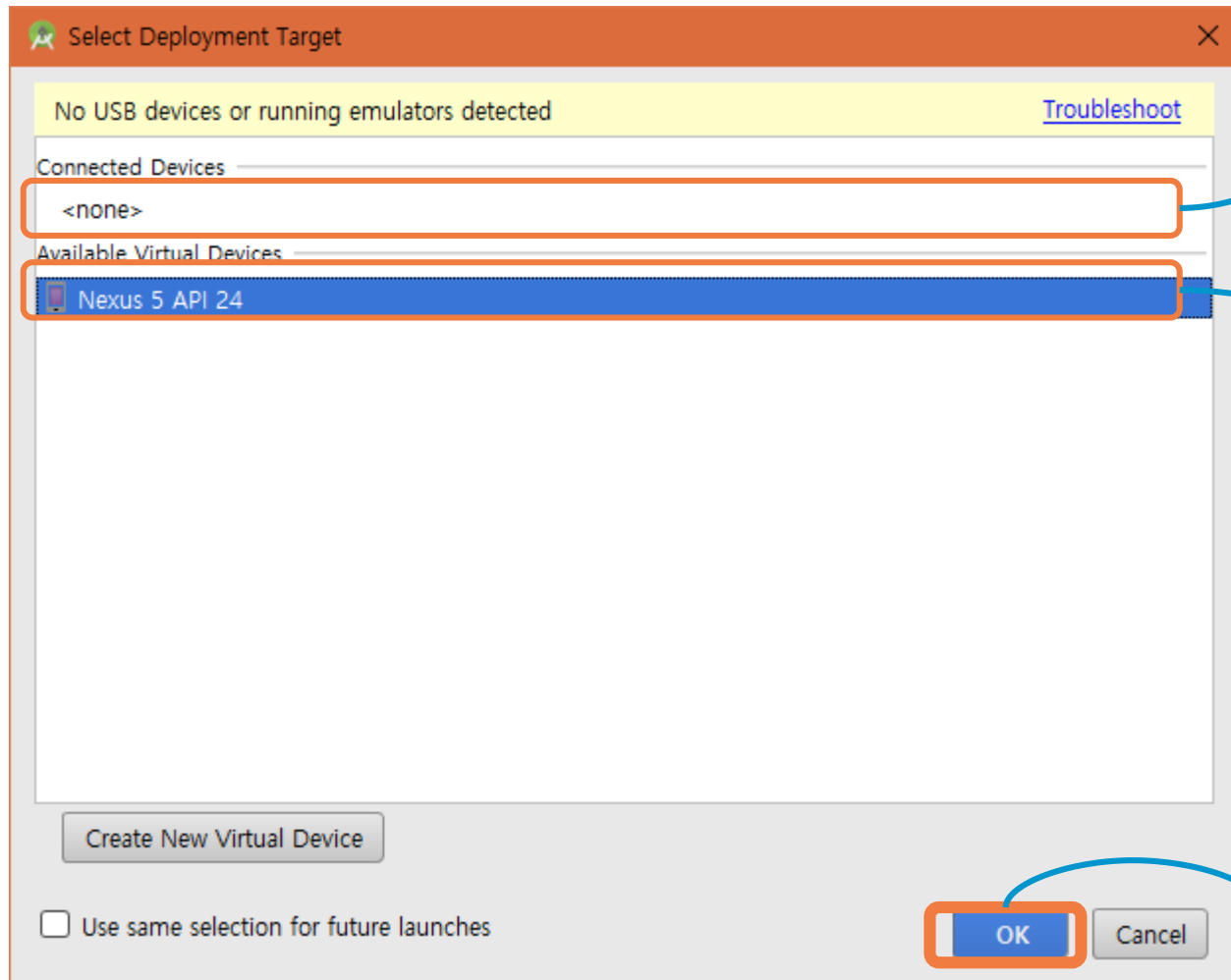


Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기

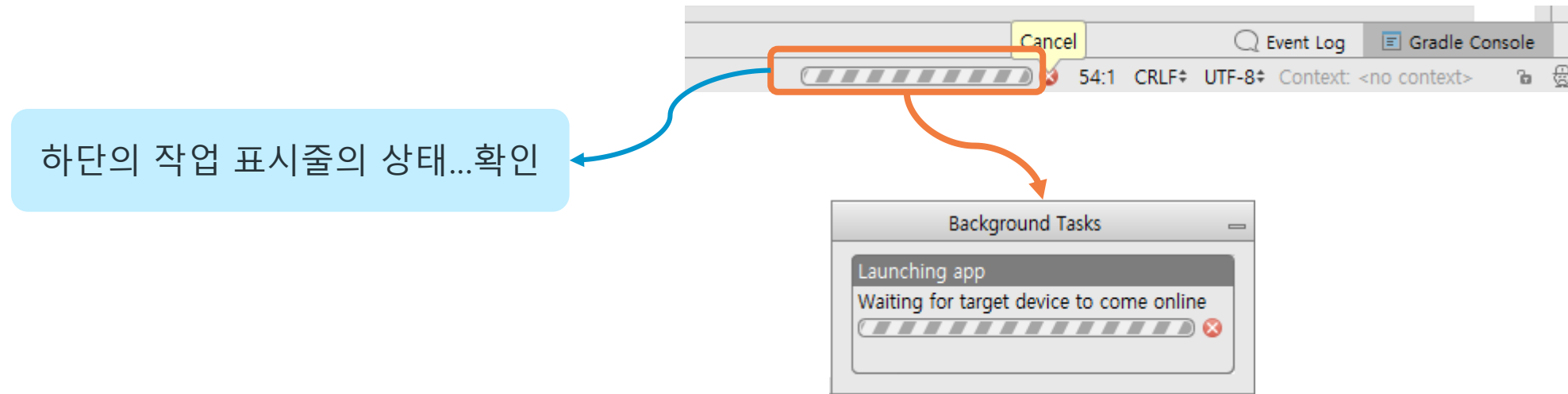


데이터 케이블로 연결된
스마트폰

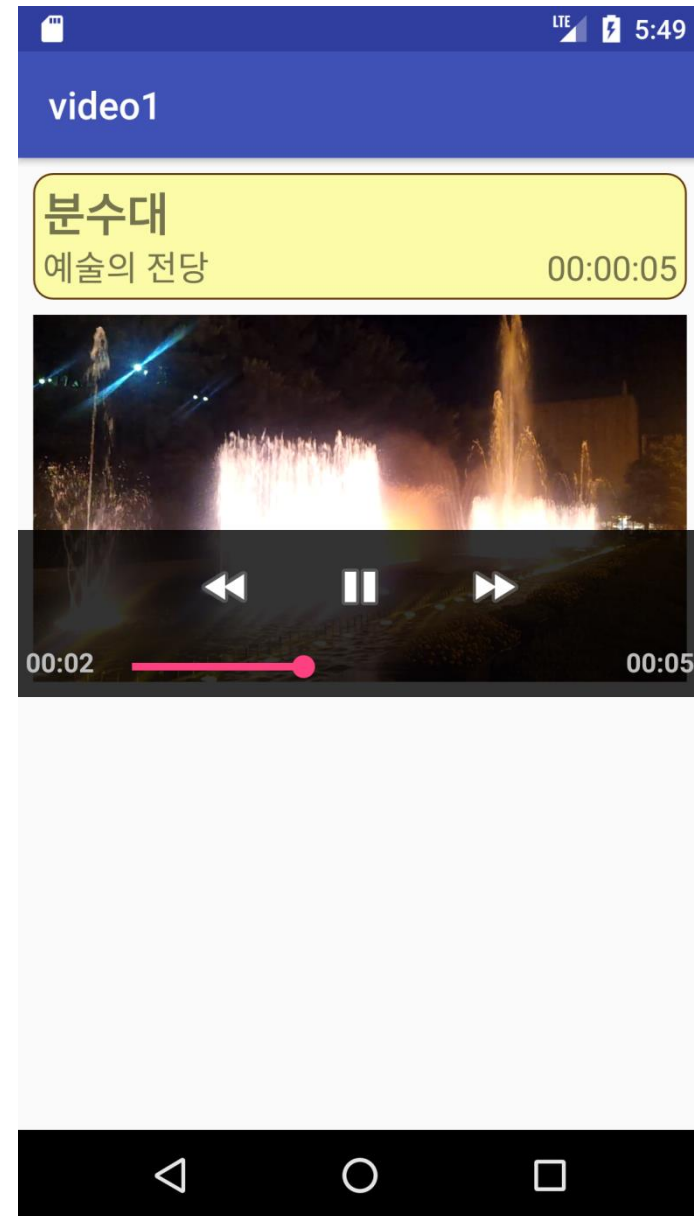
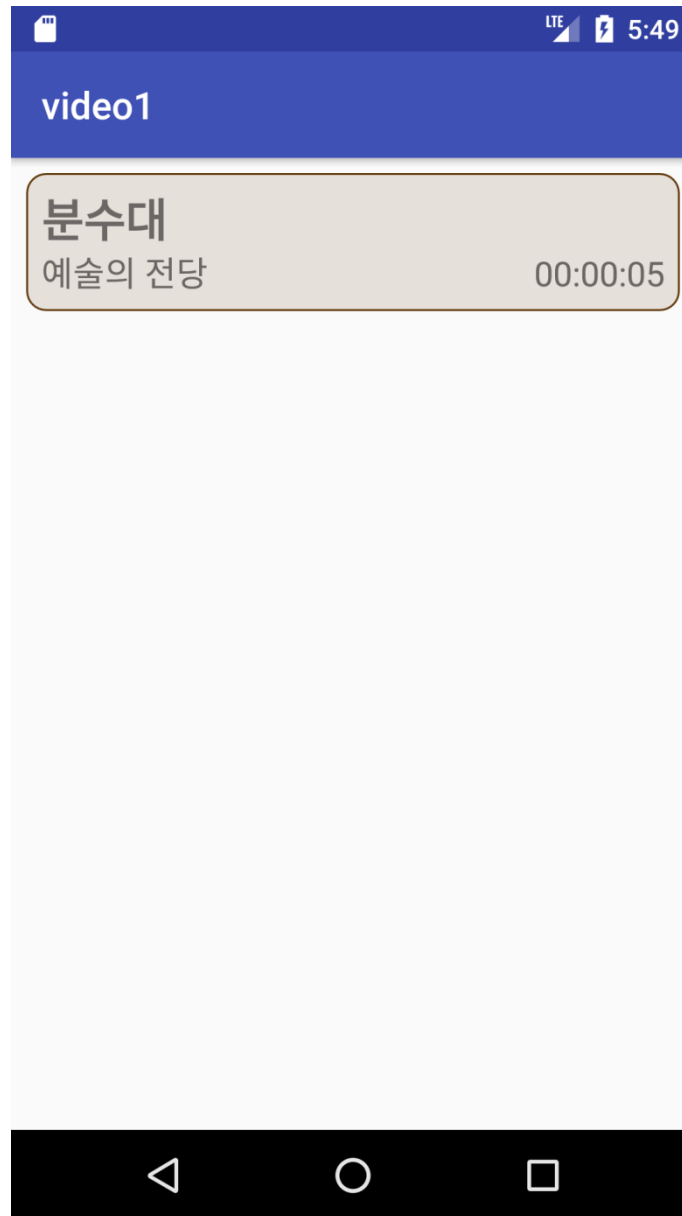
AVD

스마트폰 또는 AVD를 선택하고
'OK' 버튼을 클릭

- 에뮬레이터는 실행되지만 앱이 실행되지 않을 경우



O utputs



Q uestion & A nswer

47

