

Week08. 애니메이션



안드로이드 앱

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>

애니메이션 활용 앱의 예

4



(a) 겨울왕국



(b) 클래시 오브 클랜



애니메이션 원리

5



프레임1



프레임2



프레임3



프레임4

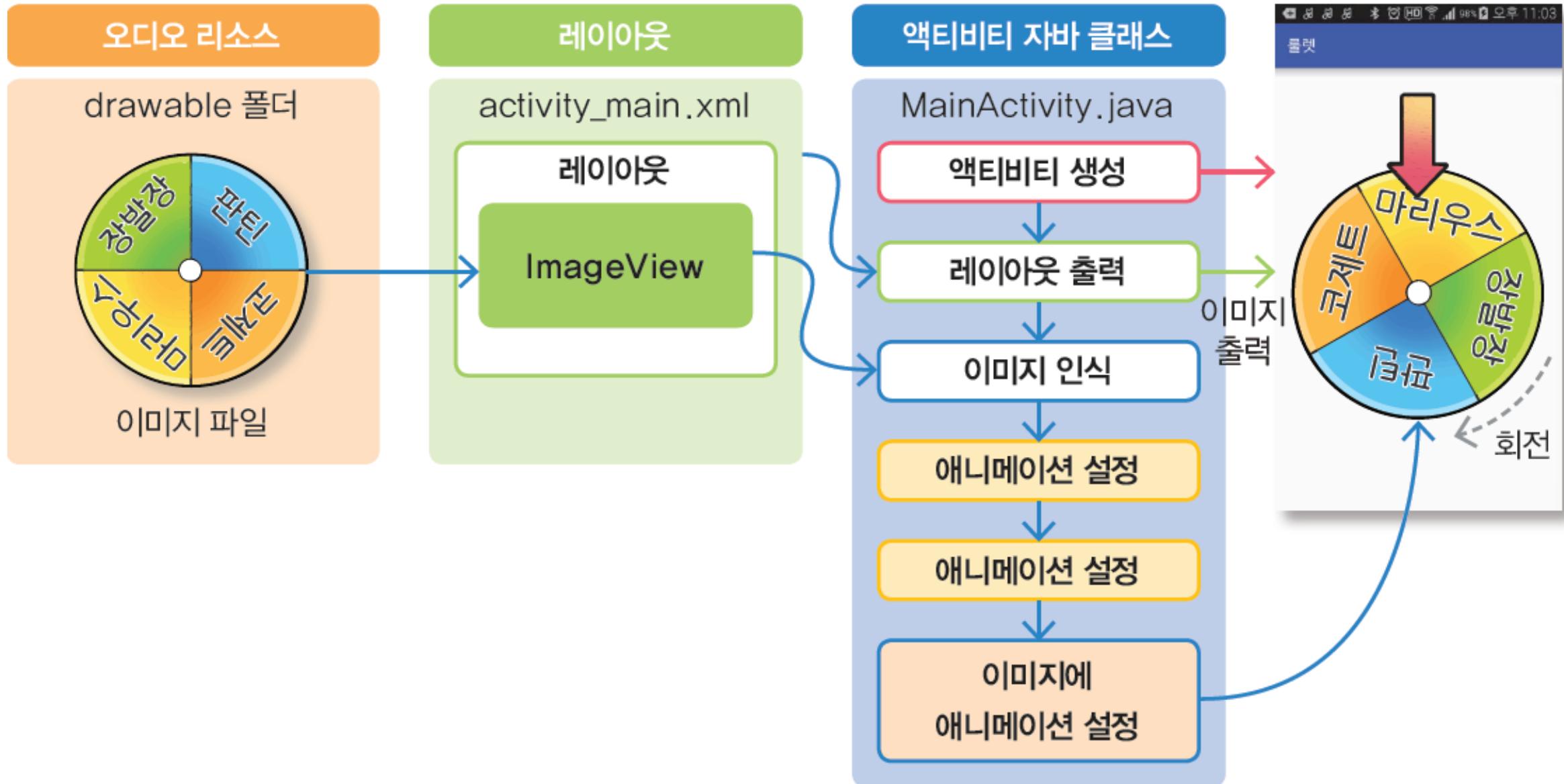
움직이는 동작



이동/크기 변화

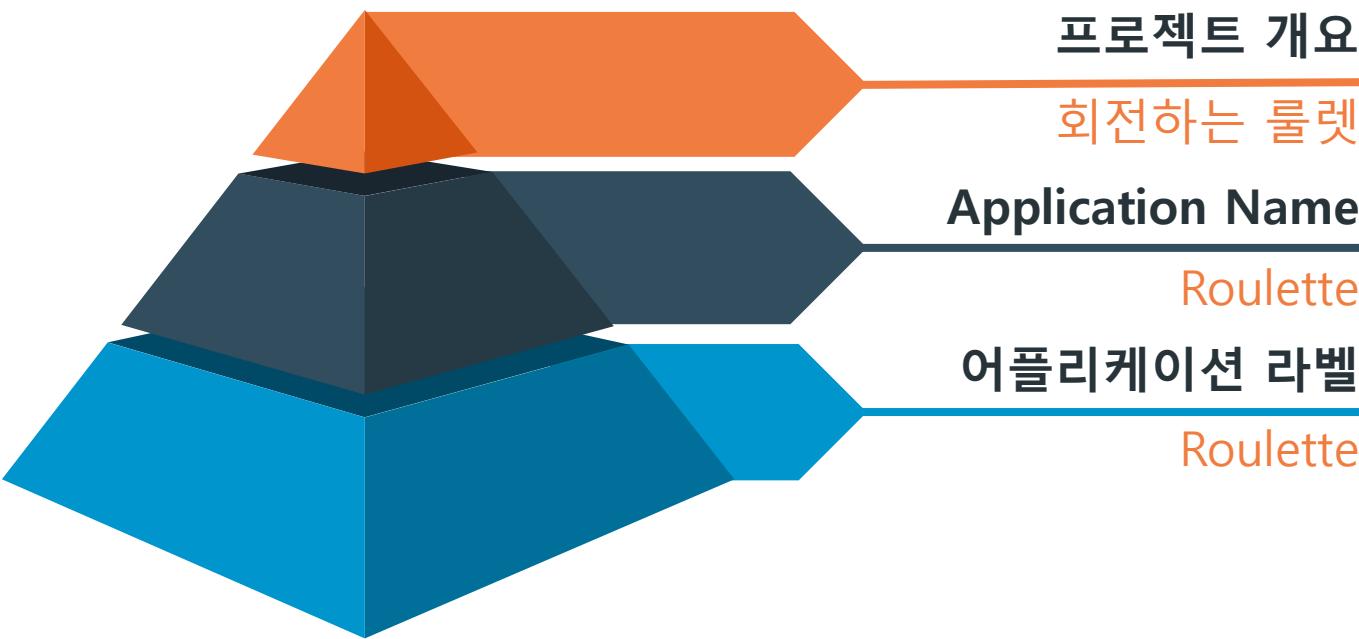
애니메이션 관련 자바 클래스 이용

6



Step 0. 프로젝트 개요

7



Step 1. 프로젝트 생성

8

절차	내용
① 프로젝트 시작	메뉴에서 'File → New Project' 클릭
② 프로젝트 구성	Application Name: Roulette Company Domain: kyungtae.example.com (디폴트 사용) Project location: ~₩user₩AndroidStudioProjects₩AnimationApp
③ 제품형태	Phone and Tablet(사용할 안드로이드 버전 지정: Android 8.1 Oreo)
④ 액티비티 유형	Empty Activity
⑤ 파일 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.kyungtae.video1	MainActivity.java	<ul style="list-style-type: none">• 룰렛 이미지 인식• 룰렛 이미지에 회전 애니메이션 실행
res	drawable	roulettle.png	<ul style="list-style-type: none">• 룰렛 이미지
		needle.png	<ul style="list-style-type: none">• 바늘 이미지
	layout	activity_main.xml	<ul style="list-style-type: none">• 룰렛과 바늘 이미지의 화면 배치
	mipmap	ic_launcher.png	
	values	colors.xml	
		dimens.xml	
	values	strings.xml	<ul style="list-style-type: none">• 어플리케이션 라벨
		styles.xml	

이미지 리소스



roulette.png, needle.png(drawable)



화면 레이아웃

```
RelativeLayout
    ImageView
        id @+id/roulette
        src @drawable/roulette
    ImageView
        src @drawable/needle
```

activity_main.xml (layout)

룰렛 이미지 ID

액티비티 제어

```
onCreate()
    super.onCreate()
    setContentView(R.layout.activity_main)

    ImageView iv =
        (ImageView)findViewById(R.id.roulette);
    ObjectAnimator anim =
        ObjectAnimator.ofFloat
            (iv, " rotation ",0, 405);

    anim.setTarget(iv);
    anim.start();
```

MainActivity.java (layout)

어플리케이션 구성
액티비티의 자바 클래스

어플리케이션 기본 정보

```
application
    icon @mipmap/ic_launcher
    label @string/app_name
    theme @style/AppTheme
    activity
        name MainActivity
```

AndroidManifest.xml (manifest)

룰렛 이미지 인식

컴파일/빌더

애니메이션 객체 생성



앱 라벨

텍스트 자원

```
string
    app_name Roulette1
```

strings.xml (values)

컴파일/빌더 정보

```
build.gradle(Project)
build.gradle(Module app)
gradle properties
settings.gradle
local.properties
```

(Gradle Scripts)

2.1 이미지 파일 복사

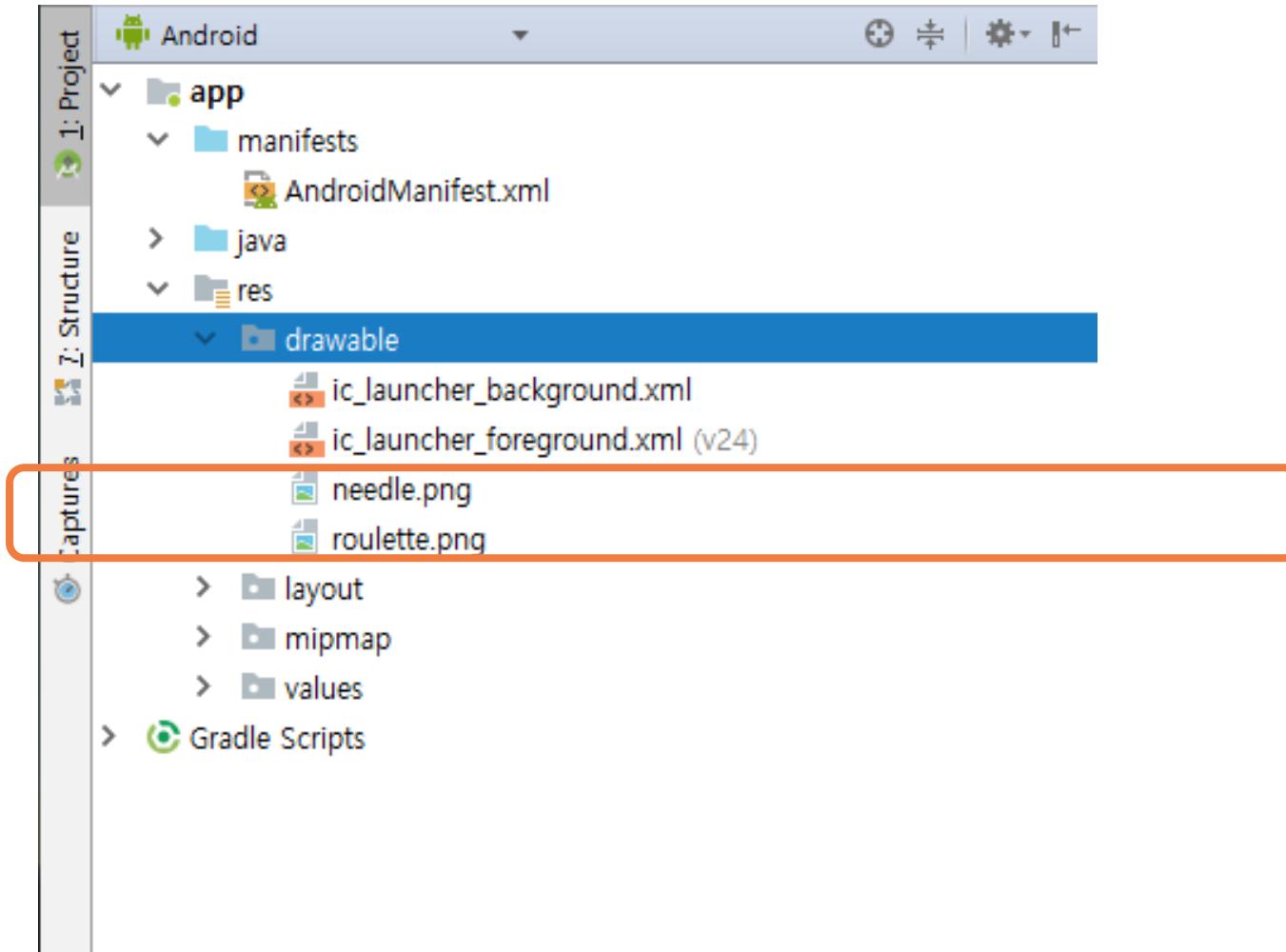
11

- res 폴더에 있는 **drawable** 폴더에 이미지 파일 저장

모듈	폴더	소스 파일	내용
res	drawable	roulette.png	룰렛 이미지
		needle.png	바늘 이미지

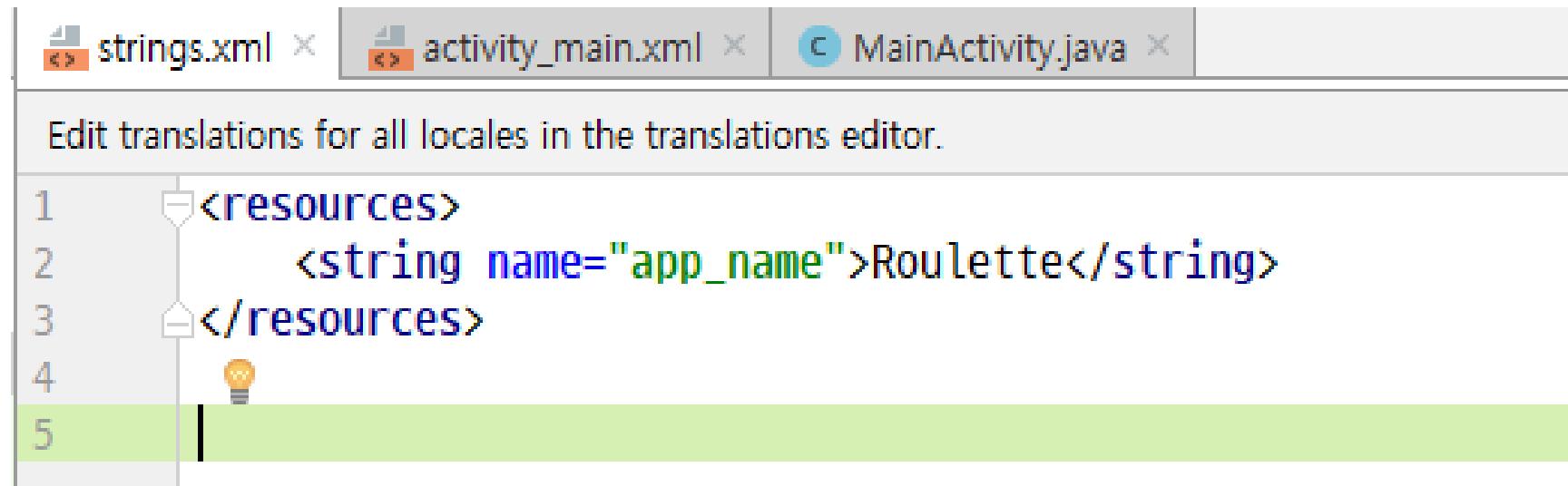
Step 2.2 res/drawable에 이미지 붙여넣기

12



Step 2.3 텍스트 자원의 편집

- strings.xml

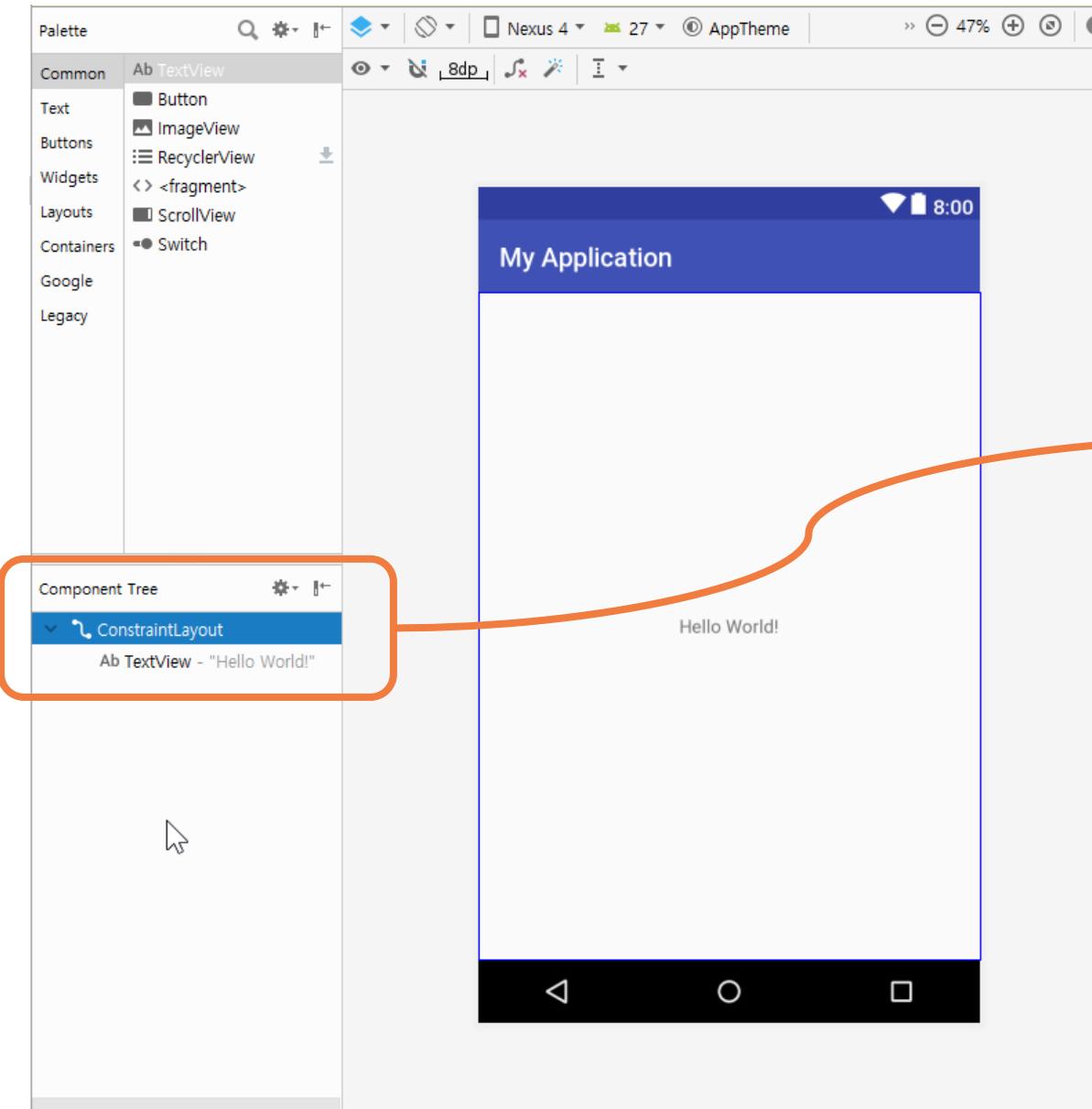


The screenshot shows the Android Studio interface with the tabs "strings.xml", "activity_main.xml", and "MainActivity.java". The "strings.xml" tab is active. The code in the editor is:

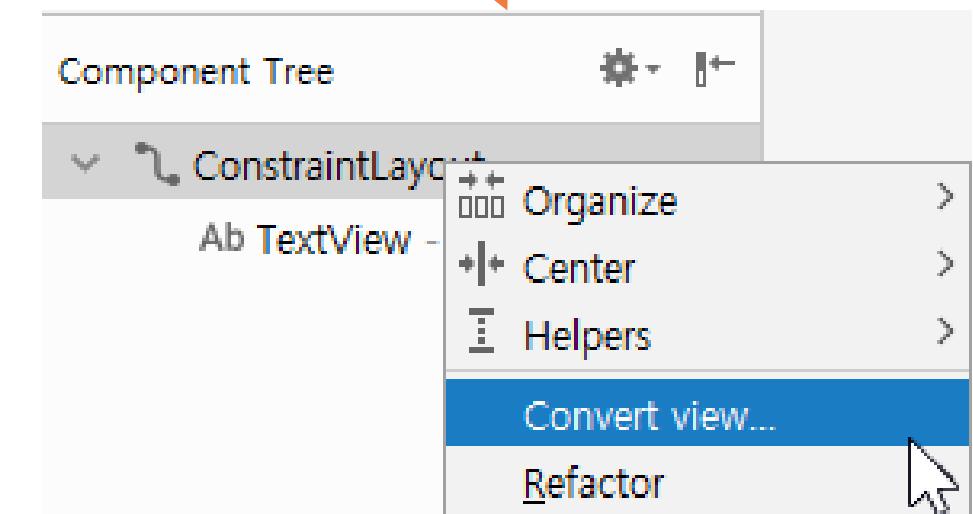
```
1 <resources>
2     <string name="app_name">Roulette</string>
3 </resources>
```

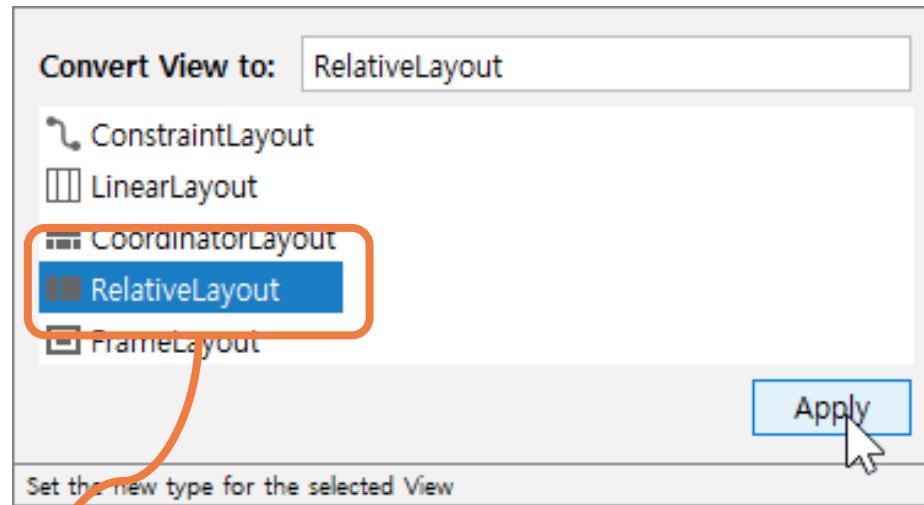
Line 1 has a small icon above it, and line 5 has a lightbulb icon above it. A green bar highlights the entire code block.

2.4 화면 설계

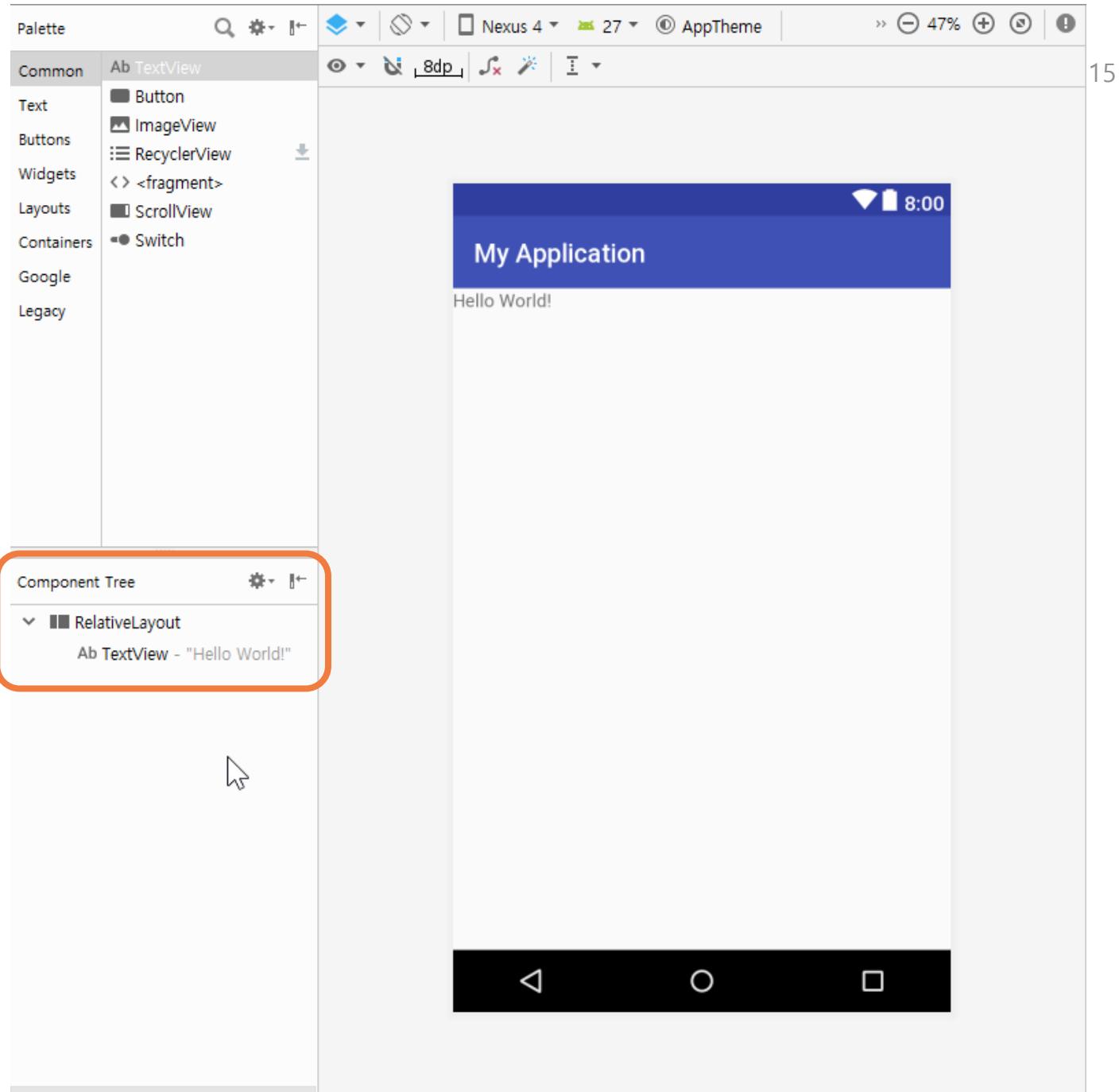


ConstraintLayout →
RelativeLayout으로 변경

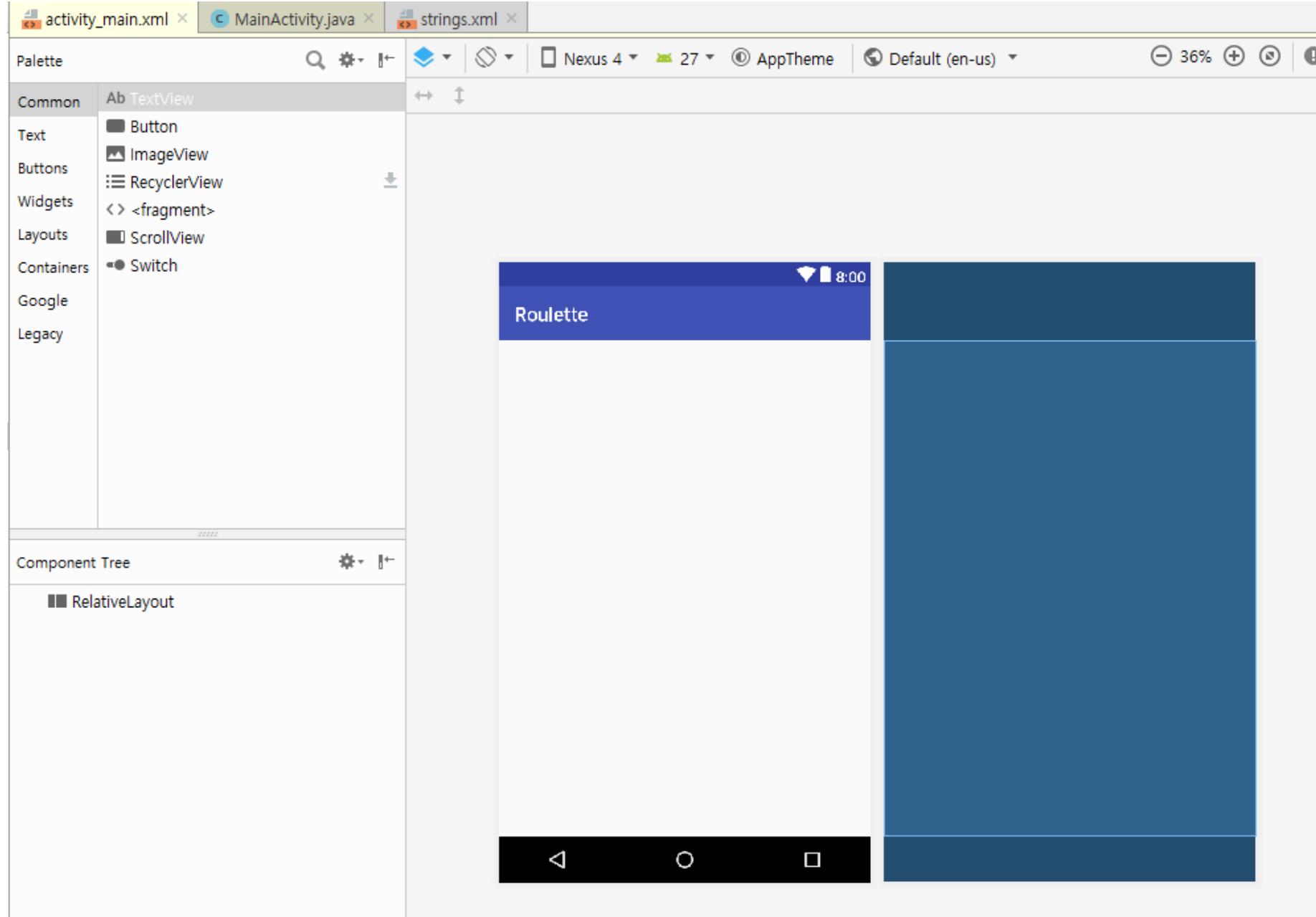




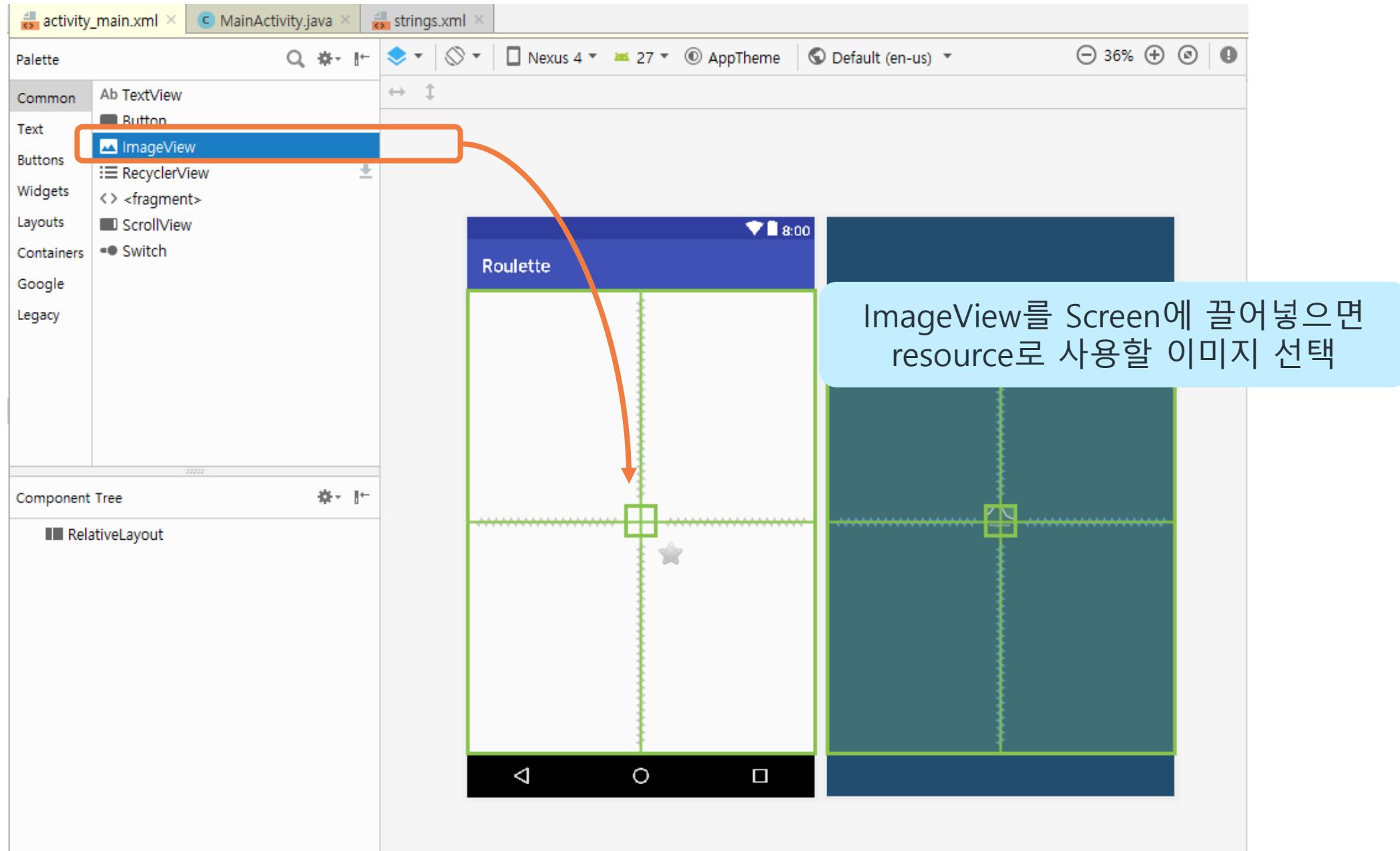
RelativeLayout 선택



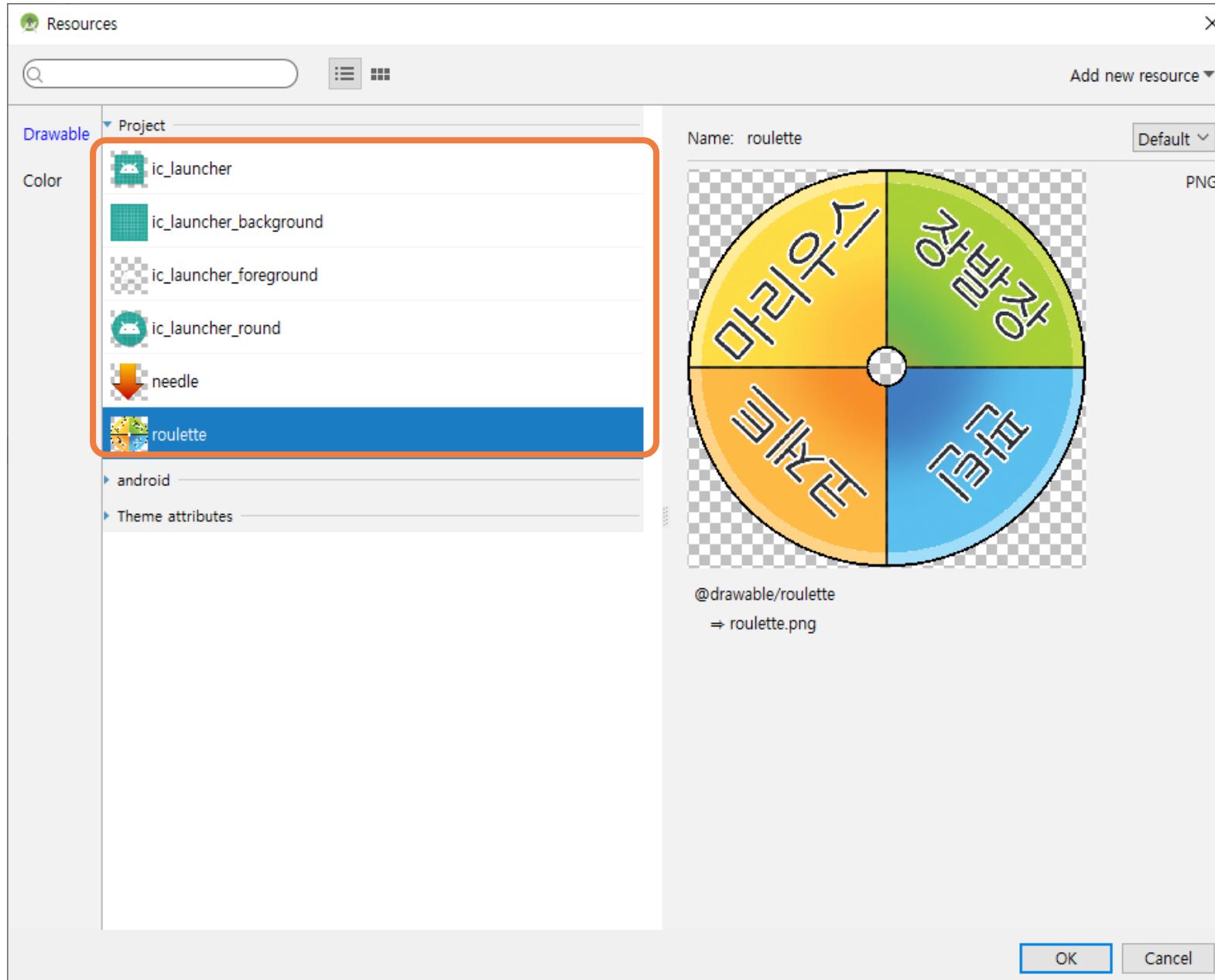
• Layout 초기화 설정 – 기본 TextView 삭제



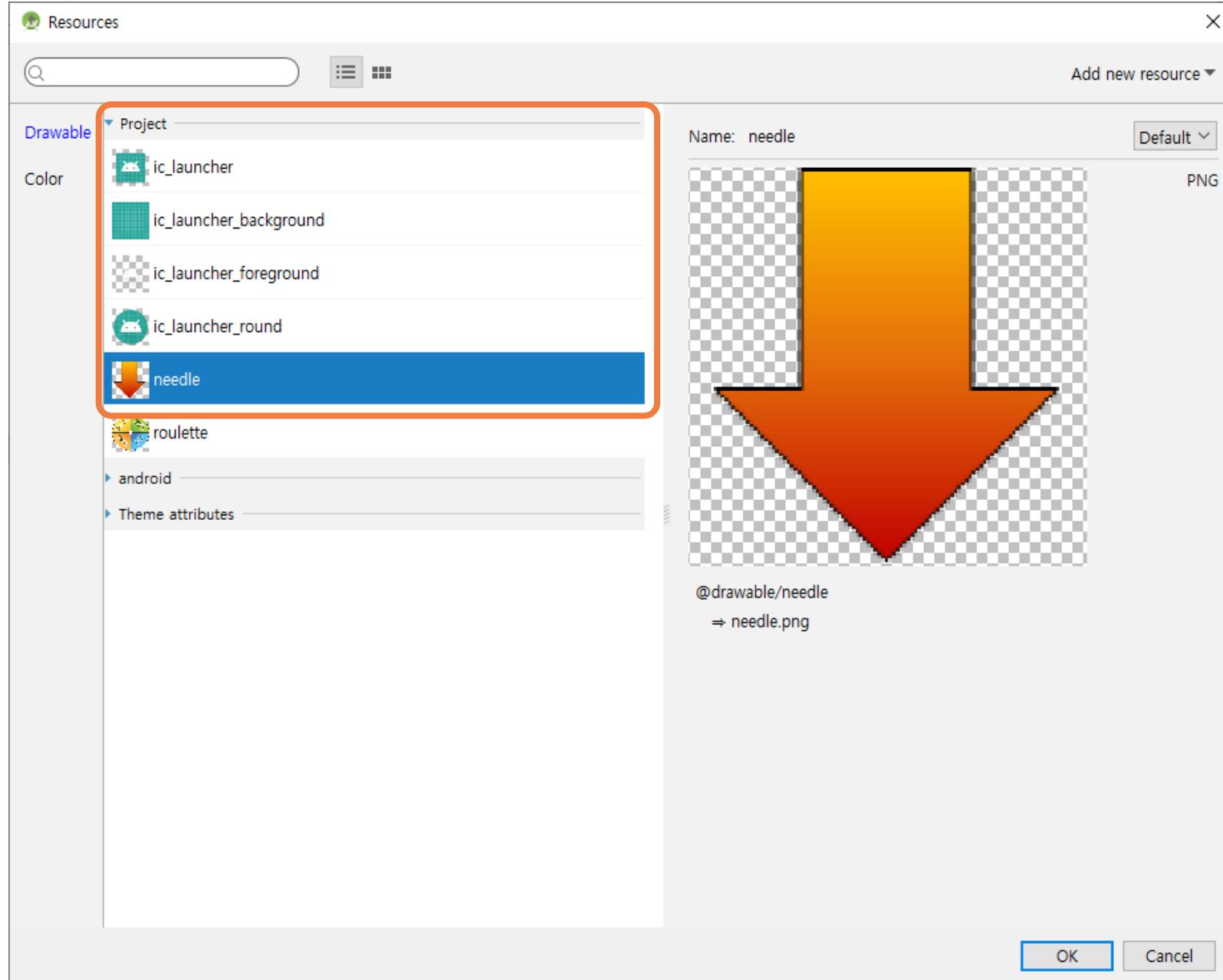
• ImageView 컴포넌트 삽입하기



- Drawable리소스에서 roulette 이미지 선택



- Drawable리소스에서 needle 이미지 선택



• ImageView – needle.png 속성 설정

The screenshot shows the Android Studio XML layout editor. On the left, the preview window displays a roulette wheel with four colored segments: yellow (top-left), green (top-right), orange (bottom-left), and blue (bottom-right). A white dot at the center represents the roulette ball. An orange arrow points from the center of the roulette wheel to the top edge of the green segment. The green segment contains Korean text: '마리우스' (Marius) on the top-left and '장발장' (Jangbaljang) on the top-right. The orange segment contains '룰렛' (Roulette) and the blue segment contains '꽁지' (Free Bet).

The XML code for the ImageView is as follows:

```
<ImageView  
    android:id="@+id/needle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"/>  
    <!-- Other attributes -->
```

The Attributes panel on the right lists the properties of the ImageView component. Two specific properties are highlighted with orange boxes and arrows pointing to them:

- layout_alignParentTop**: A checked checkbox under the Attributes panel.
- layout_centerHorizontal**: A checked checkbox under the Attributes panel.

A callout bubble highlights these two properties with the text:

layout_alignParentTop: check
layout_centerHorizontal: check

id	needle
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
layout_alignParentTop	<input checked="" type="checkbox"/>
layout_centerHorizontal	<input checked="" type="checkbox"/>
srcCompat	@drawable/needle
accessibilityLiveRegion	
accessibilityTraversalA	
accessibilityTraversalB	
adjustViewBounds	<input type="checkbox"/>
alpha	
autofillHints	
background	
backgroundTint	
baseline	
baselineAlignBottom	<input type="checkbox"/>
clickable	<input type="checkbox"/>
contentDescription	
contextClickable	<input type="checkbox"/>
cropToPadding	<input type="checkbox"/>
defaultFocusHighlight	<input type="checkbox"/>
drawingCacheQuality	
duplicateParentState	<input type="checkbox"/>
fadeScrollbars	<input type="checkbox"/>
fadingEdge	<input type="checkbox"/>
fadingEdgeLength	

- ImageView – roulette.png 속성 설정

The screenshot shows the Android Studio XML layout editor. On the left, there's a preview of an activity titled "Roulette". It features a roulette wheel with four segments: yellow (top-left), green (top-right), orange (bottom-left), and blue (bottom-right). The segments contain Korean text: "마리우시" (Mariuji) in yellow, "장발장" (Jangbaljang) in green, "룰렛" (Roulette) in orange, and "회전" (Hyeonjeon) in blue. A red arrow points from the "layout_centerInParent" attribute in the Attributes panel to the roulette wheel in the preview. On the right, there are two image views. The top one has the text "ImageView" and the bottom one also has the text "ImageView".

Attributes

id	roulette
layout_width	wrap_content
layout_height	wrap_content
> Layout_Margin [?, ?, ?, ?, ?]	
> Padding [?, ?, ?, ?, ?]	
> Theme	
elevation	
layout_centerInParent	<input checked="" type="checkbox"/>
srcCompat	@drawable/roulette
accessibilityLiveRegion	
accessibilityTraversalA	
accessibilityTraversalB	
adjustViewBounds	<input type="checkbox"/>
alpha	

layout_centerInParent: check

2.5 Activity 제어(MainActivity.java)

22

- 이미지 리소스에 대한 ImageView를 획득과 애니메이션 재생

```
1 package com.example.user.animationapp;  
2  
3 import android.animation.ObjectAnimator;  
4 import android.support.v7.app.AppCompatActivity;  
5 import android.os.Bundle;  
6 import android.view.animation.AccelerateDecelerateInterpolator;  
7 import android.widget.ImageView;  
8  
9 public class MainActivity extends AppCompatActivity {  
10  
11     @Override  
12     protected void onCreate(Bundle savedInstanceState) {  
13         super.onCreate(savedInstanceState);  
14         setContentView(R.layout.activity_main);  
15  
16         ImageView iv = (ImageView) findViewById(R.id.roulette);  
17  
18         // 0~405도까지의 rotation 효과  
19         ObjectAnimator object = ObjectAnimator.ofFloat(iv, "rotation", ...values: 0.0f, 405f);  
20         // 점점 가속되다가 감속되는 애니메이션  
21         object.setInterpolator( new AccelerateDecelerateInterpolator());  
22         // 10초 동안  
23         object.setDuration(10000);  
24         object.start();  
25     }  
26  
27 }
```

이미지뷰 객체 획득

0~405도 회전시키는 회전 객체 생성

애니메이션이 처음 실행될 때는 가속되다가 가속도를 줄여 정지.

BounceInterpolator 사용해보기

23

The screenshot shows the Android Studio interface with three tabs at the top: strings.xml, activity_main.xml, and MainActivity.java. The MainActivity.java tab is active, displaying Java code for an Android application.

```
1 package com.example.user.animationapp;
2
3 import android.animation.ObjectAnimator;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.animation.AccelerateDecelerateInterpolator;
7 import android.view.animation.BounceInterpolator;
8 import android.widget.ImageView;
9
10 public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         ImageView iv = findViewById(R.id.roulette);
18
19         ObjectAnimator object = ObjectAnimator.ofFloat(iv, "propertyName: rotation", ...values: 0.0f, 405f);
20
21         object.setInterpolator(new BounceInterpolator());
22         object.setDuration(10000);
23         object.start();
24
25     }
26
27 }
```

The line of code `object.setInterpolator(new BounceInterpolator());` is highlighted with an orange rectangle, indicating it is the focus of the current slide. The code uses the `ObjectAnimator` to animate the rotation of an `ImageView` with a duration of 10,000ms, using a `BounceInterpolator` for the animation's behavior.

클래스와 속성/메소드

• 클래스

클래스	설명
ObjectAnimator	목표 객체에 대한 애니메이션 특성을 설정
AccelerateDecelerateInterpolator	애니메이션 변화가 처음과 끝은 작지만 중간 부분에는 빠름

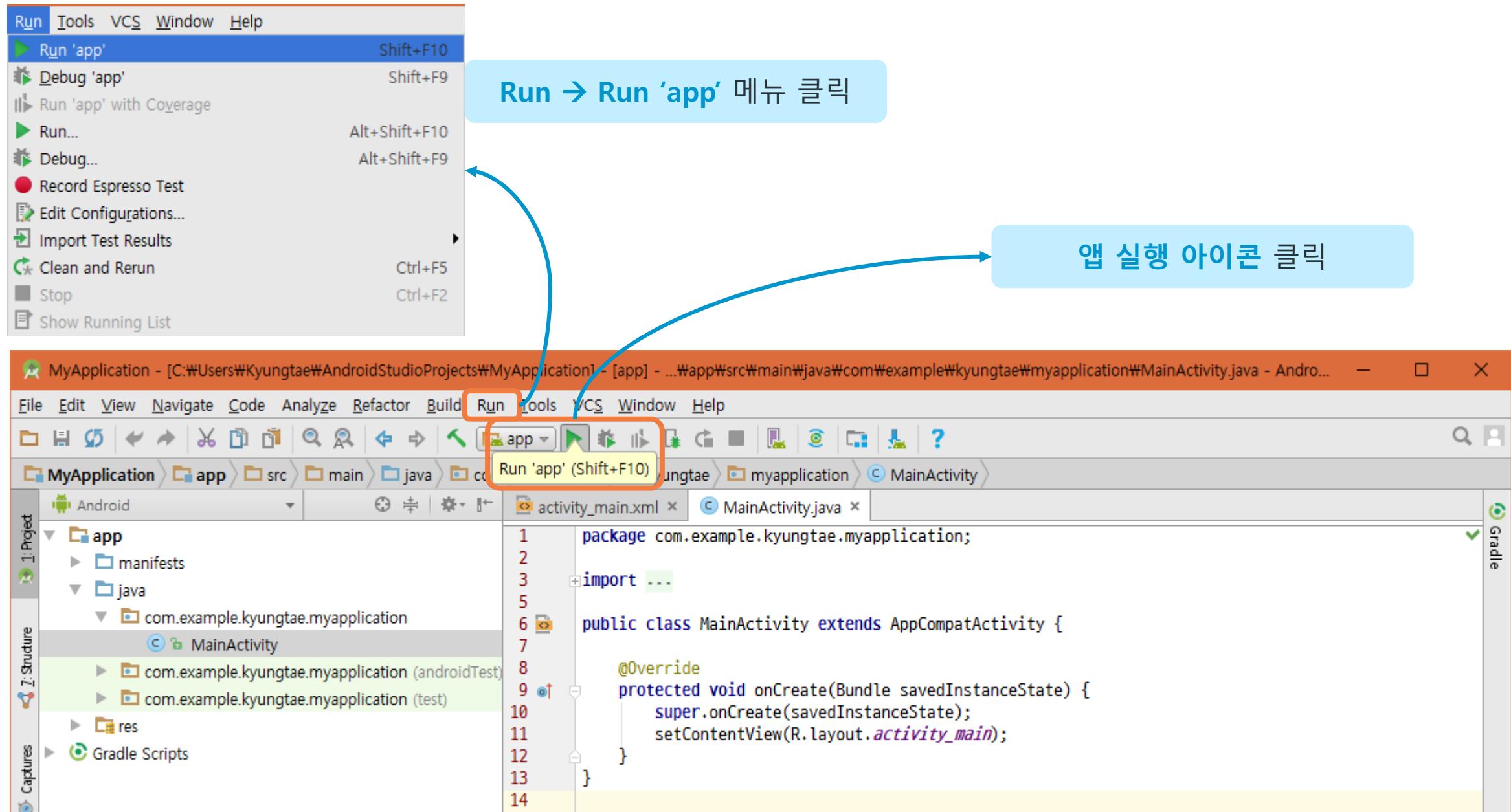
• 메소드

클래스	메소드	설명								
ObjectAnimator	<code>static ObjectAnimator ofFloat(Object target, String propertyName, float ...values)</code>	values 사이의 애니메이션을 만들고 ObjectAnimator를 반환함 <table border="1"><thead><tr><th>매개변수</th><th>설명</th></tr></thead><tbody><tr><td>target</td><td>애니메이션 대상</td></tr><tr><td>propertyName</td><td>애니메이션 특성 이름</td></tr><tr><td>values</td><td>시간에 따라 애니메이션 될 값들</td></tr></tbody></table>	매개변수	설명	target	애니메이션 대상	propertyName	애니메이션 특성 이름	values	시간에 따라 애니메이션 될 값들
매개변수	설명									
target	애니메이션 대상									
propertyName	애니메이션 특성 이름									
values	시간에 따라 애니메이션 될 값들									
<code>ObjectAnimator setDuration(long duration)</code>	애니메이션 시간 설정, 밀리초 단위이며, 기본값은 300밀리초로 설정됨									
<code>void start()</code>	애니메이션 시작									

클래스	메소드	설명
ObjectAnimator	void setInterpolator (TimeInterpolator value)	<p>시간에 따른 애니메이션 변화율 설정.</p> <p>TimeInterpolator 서브클래스의 종류에 따라 속도의 변화는 다양하며, 그 효과는 다음과 같다.</p> <ul style="list-style-type: none"> - AccelerateDecelerateInterpolator: 애니메이션 시작시에는 서서히 움직이다가 점차 빨라지고 애니메이션이 끝날 때는 점차 서서히 멈춤 - LinearInterpolator: 애니메이션 속도가 일정함

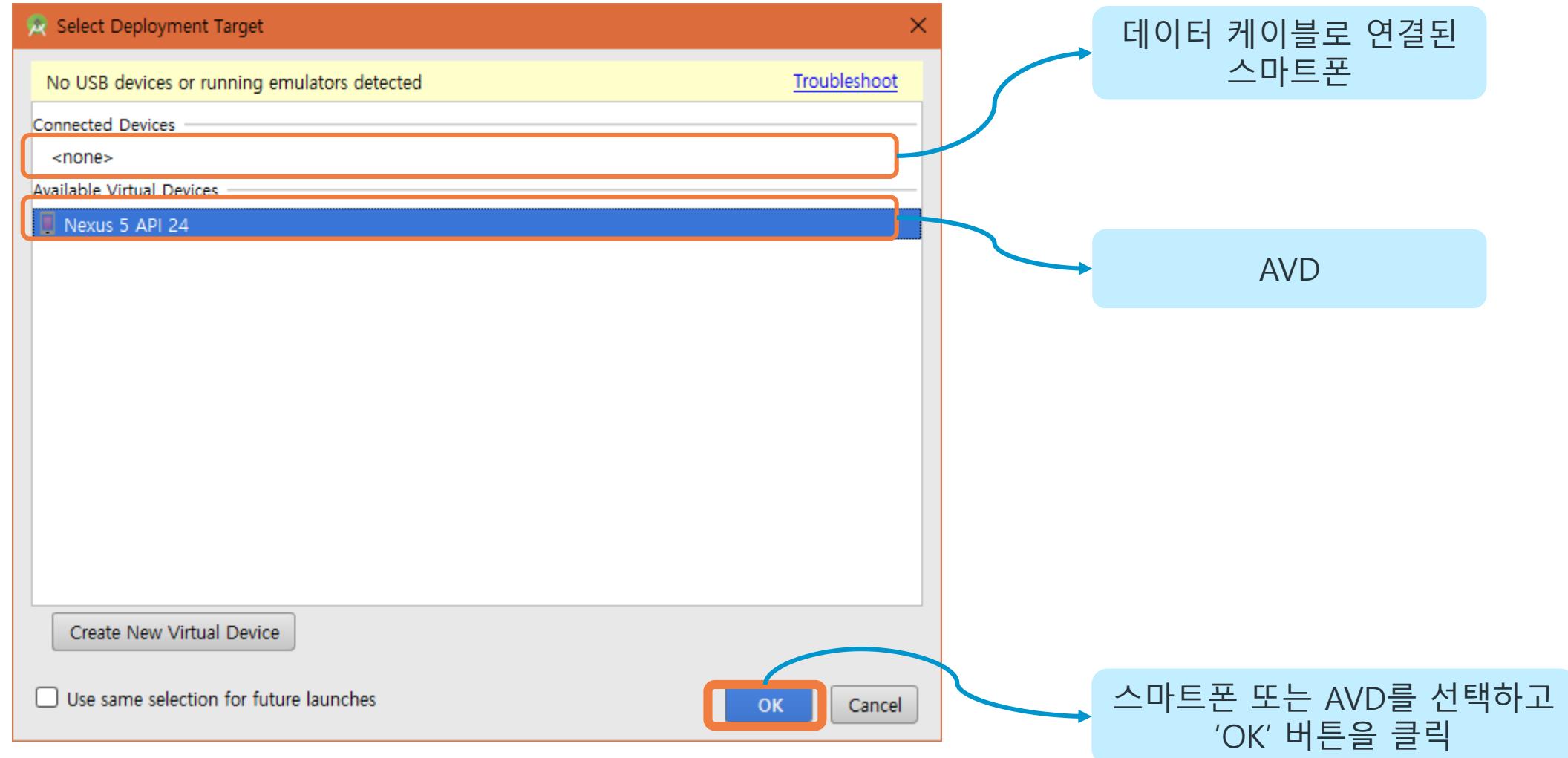
Step 3. 프로젝트 실행

26



• AVD 장비 선택하기

27

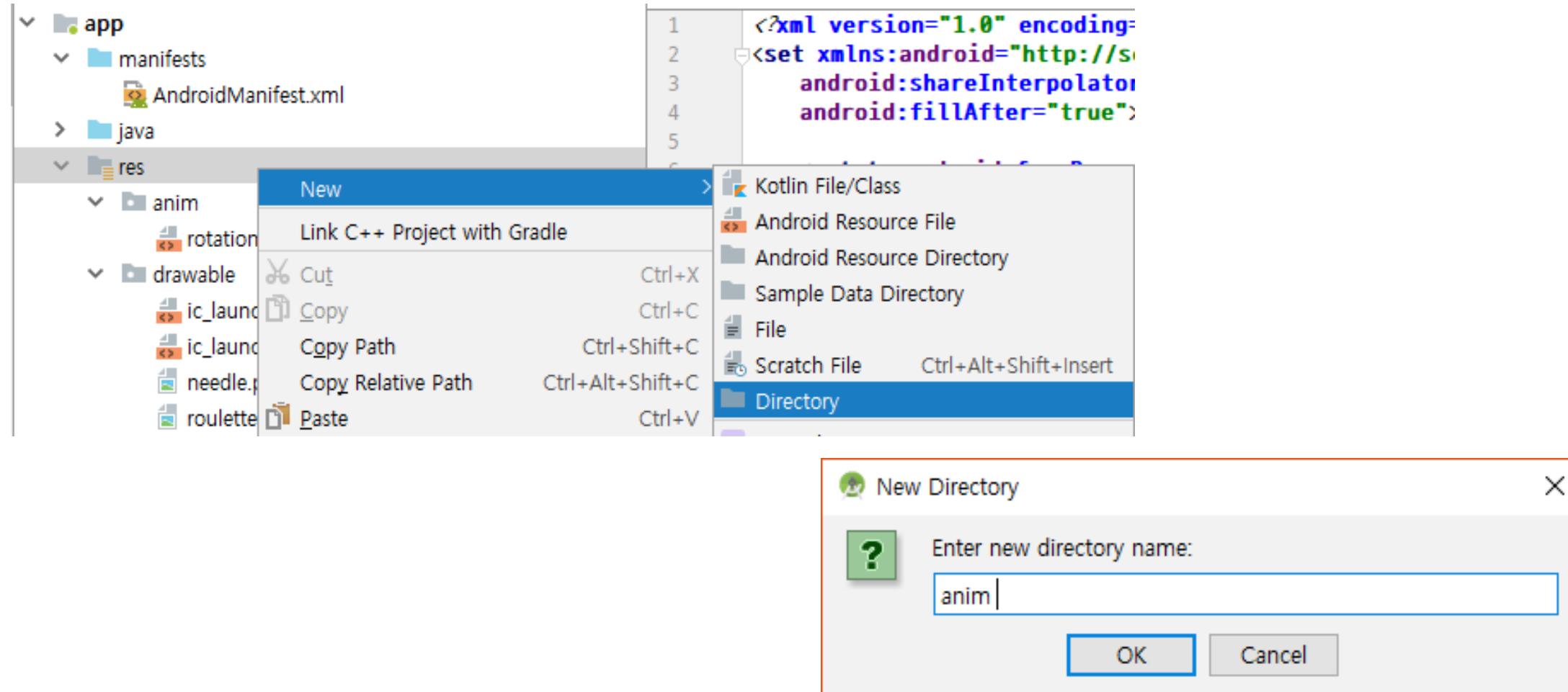




Animation 객체를 이용한 에니메이션

Animation 객체 추가 – res/anim 폴더 만들기

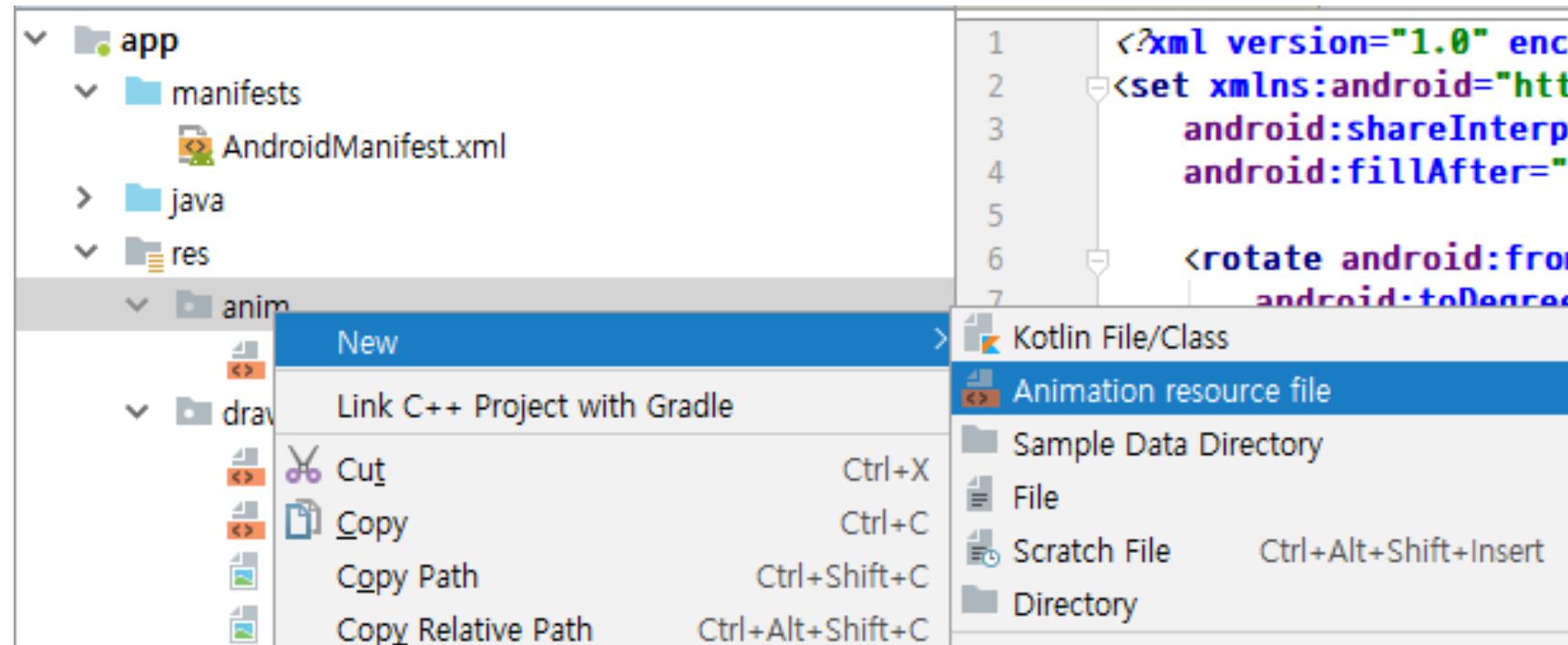
- 애니메이션 설정을 위한 xml 파일 생성



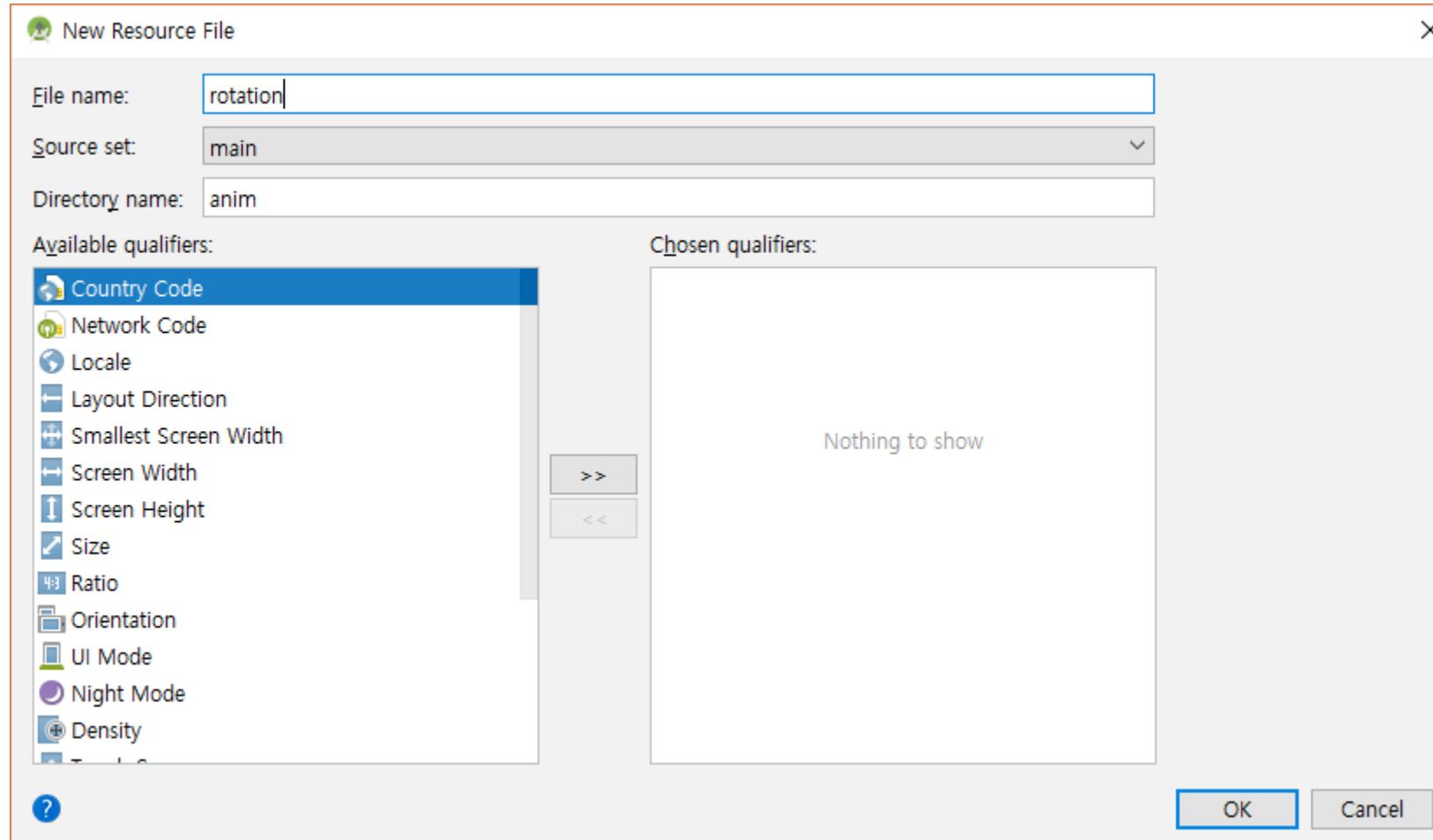
Animation 객체 추가 – rotation.xml 파일 만들기

30

- 애니메이션 설정을 위한 xml 파일 생성



rotation.xml 파일 만들기



Animation 객체 추가 – rotation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="@android:anim/accelerate_decelerate_interpolator"
    android:fillAfter="true">

    <rotate android:fromDegrees="0"
        android:toDegrees="764"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="10000"
        android:repeatCount="0" />
</set>
```

• 애니메이션이 처음 실행될 때는 가속되다가 감속되어 정지함.
• 애니메이션 종료 후 처음으로 복귀하지 않음

MainActivity.java

33

The screenshot shows the Android Studio interface with the code editor open to the `MainActivity.java` file. The tab bar at the top includes `activity_main.xml`, `AndroidManifest.xml`, `MainActivity.java` (which is the active tab), `rotation.xml`, and `strings.xml`. The code editor displays the following Java code:

```
1 package com.example.user.animationapp;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.animation.Animation;
6 import android.view.animation.AnimationUtils;
7 import android.widget.ImageView;
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         ImageView iv = (ImageView) findViewById(R.id.roulette);
17
18         Animation animation = AnimationUtils.loadAnimation(this, R.anim.rotation);
19         iv.startAnimation(animation);
20     }
21 }
22
23 }
```

The code implements a simple rotation animation for an `ImageView` with the ID `roulette` when the activity is created.

O utputs

