

Week08.

애니메이션

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>

애니메이션 활용 앱의 예

4



(a) 겨울왕국

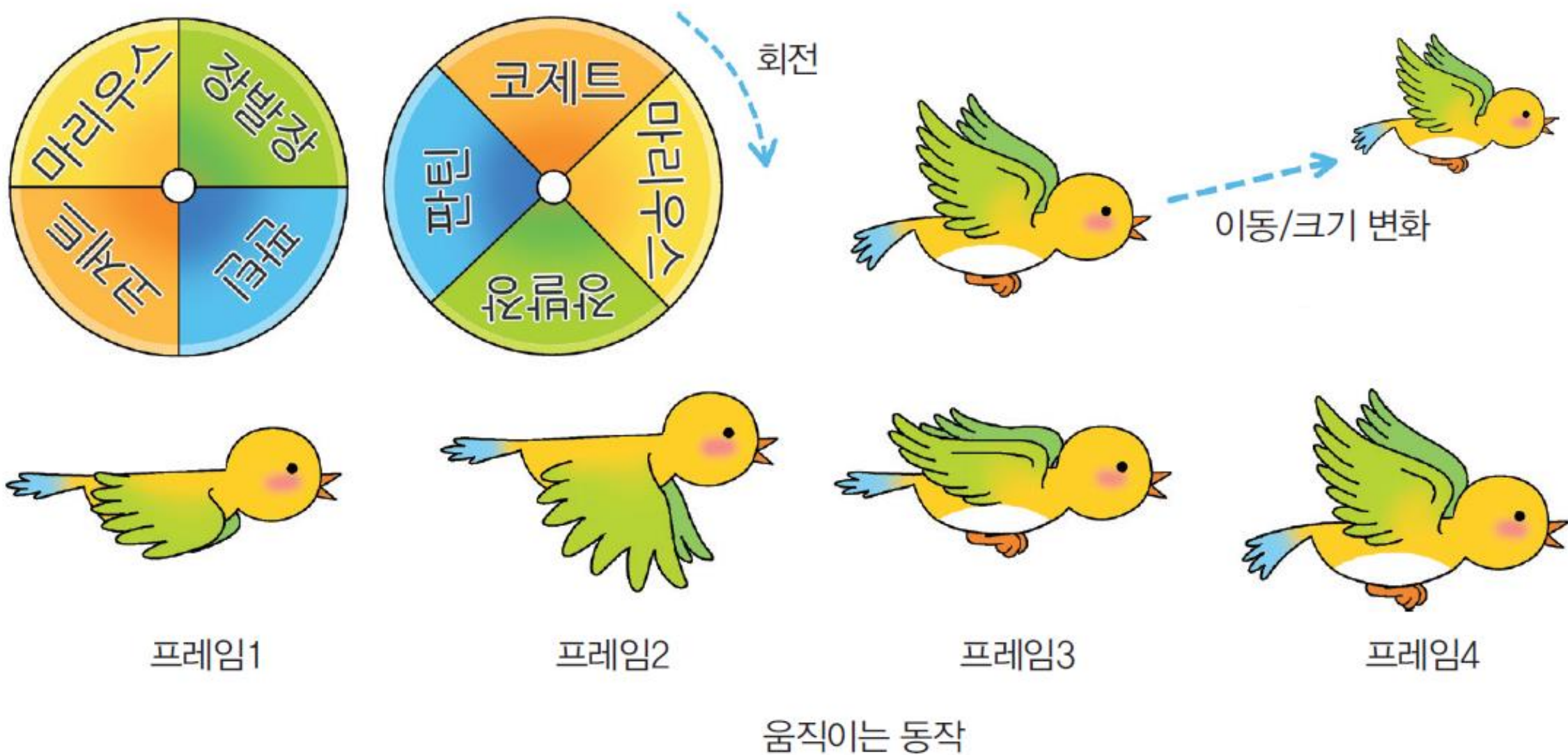


(b) 클래시 오브 클랜



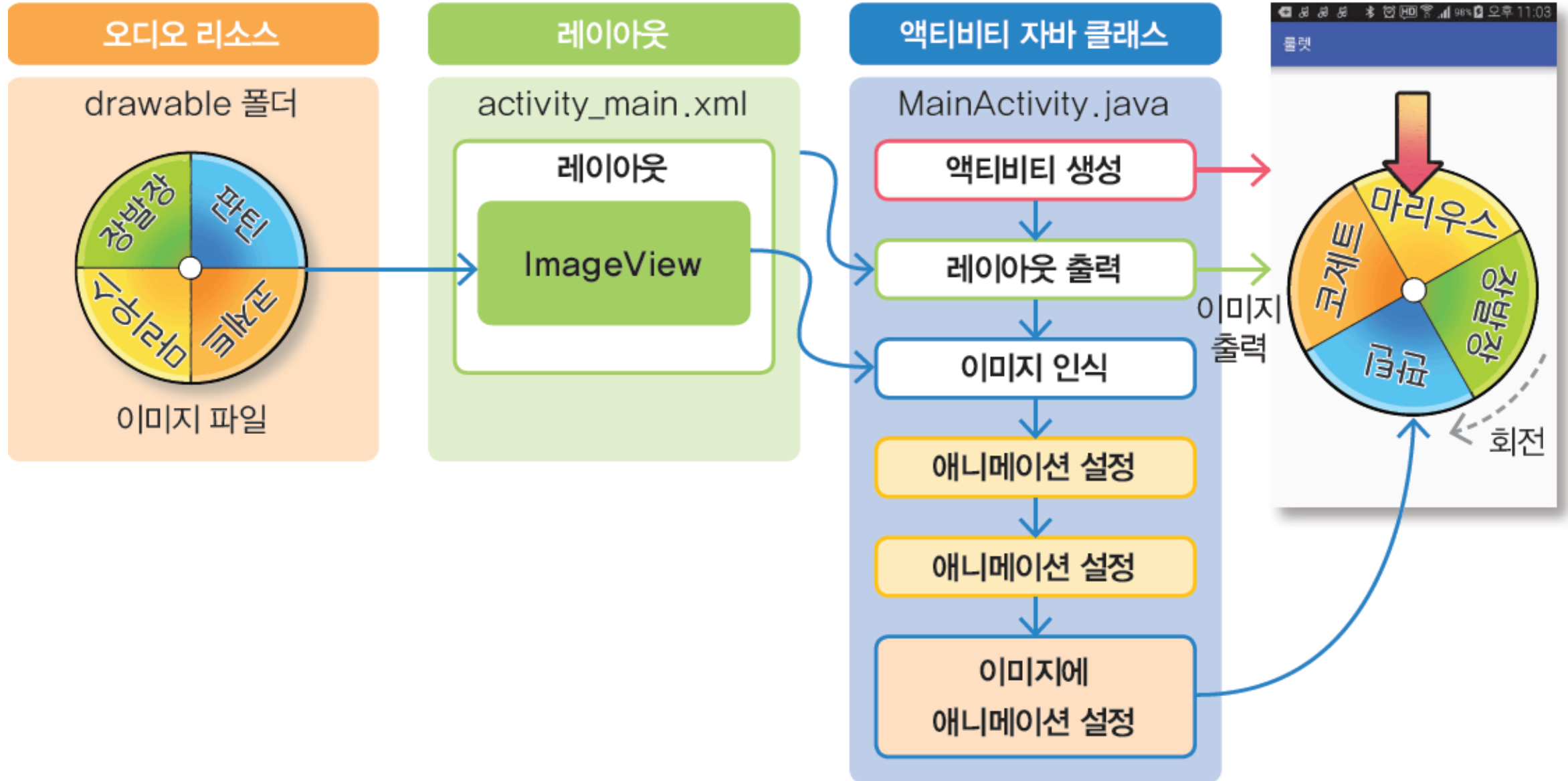
애니메이션 원리

5



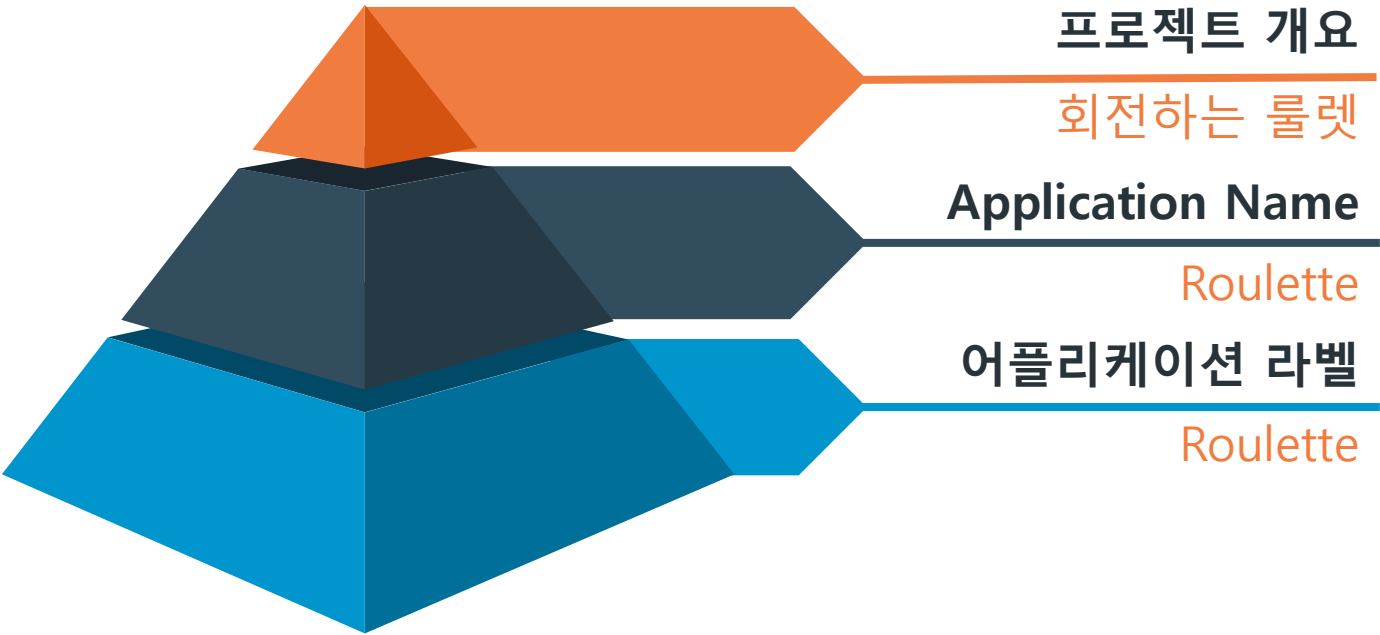
애니메이션 관련 자바 클래스 이용

6



Step 0. 프로젝트 개요

7



Step 1. 프로젝트 생성

| 절차 | 내 용 |
|----------|---|
| ①프로젝트 시작 | 메뉴에서 'File → New Project' 클릭 |
| ②프로젝트 구성 | Application Name: Roulette |
| | Company Domain: kyungtae.example.com (디폴트 사용) |
| | Project location: ~\user\AndroidStudioProjects\AnimationApp |
| ③제품형태 | Phone and Tablet (사용할 안드로이드 버전 지정: Android 8.1 Oreo) |
| ④액티비티 유형 | Empty Activity |
| ⑤파일 옵션 | Activity Name: MainActivity (디폴트 사용) |
| | Layout Name: activity_main (디폴트 사용) |

Step 2. 파일 편집

9

| 모듈 | 폴더 | 소스 파일 | 편집 내용 |
|-----------|-----------------------------|---------------------|---|
| manifests | | AndroidManifest.xml | |
| java | com.example.kyungtae.video1 | MainActivity.java | <ul style="list-style-type: none">룰렛 이미지 인식룰렛 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 | |
| | | 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

룰렛 이미지 인식

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)

애니메이션 객체 생성

텍스트 자원

```
string
    app_name Roulette1
```

strings.xml (values)

앱 라벨

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)
```

컴파일/빌더

컴파일/빌더 정보

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



(Gradle Scripts)



2.1 이미지 파일 복사

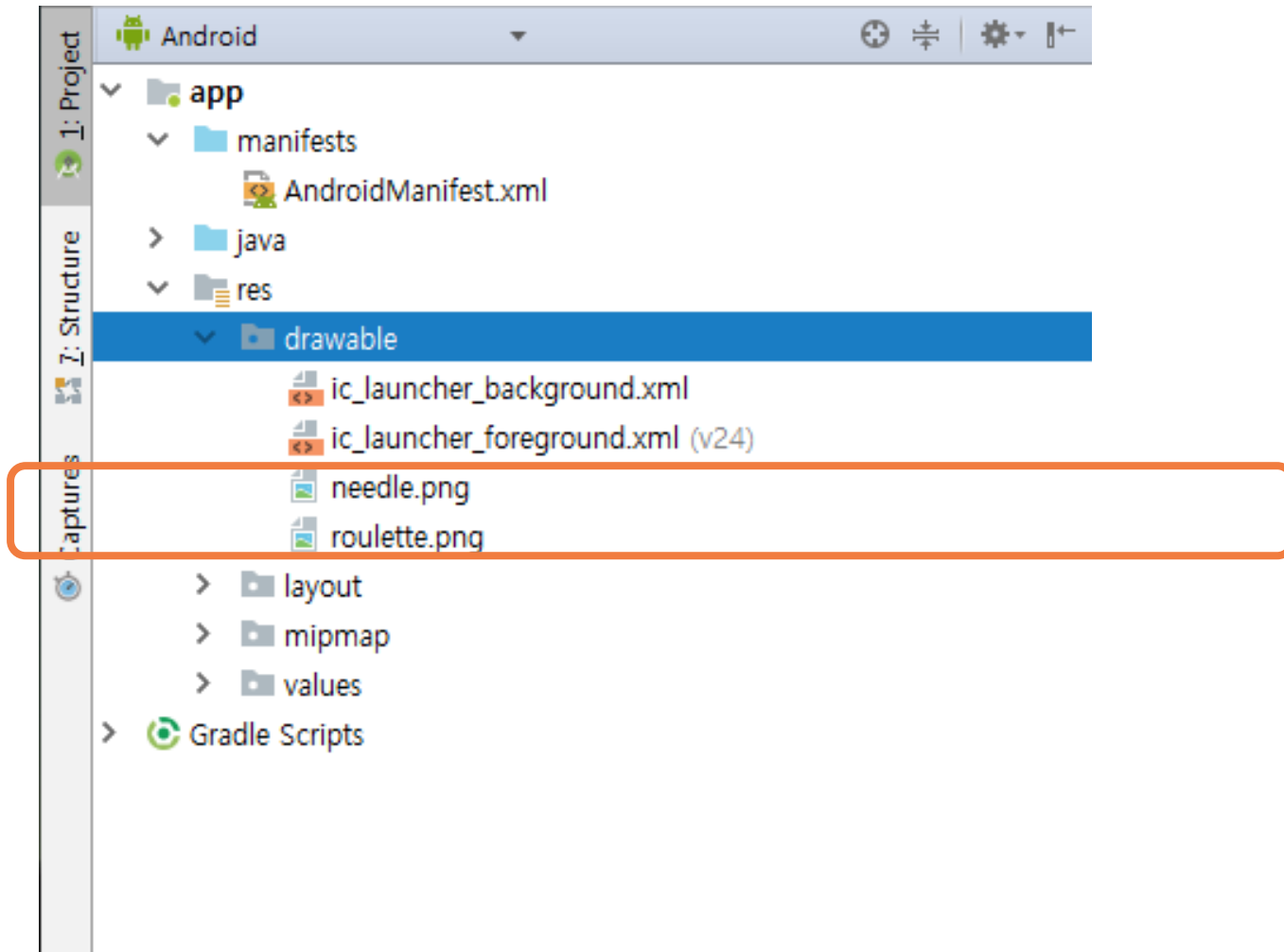
11

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

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

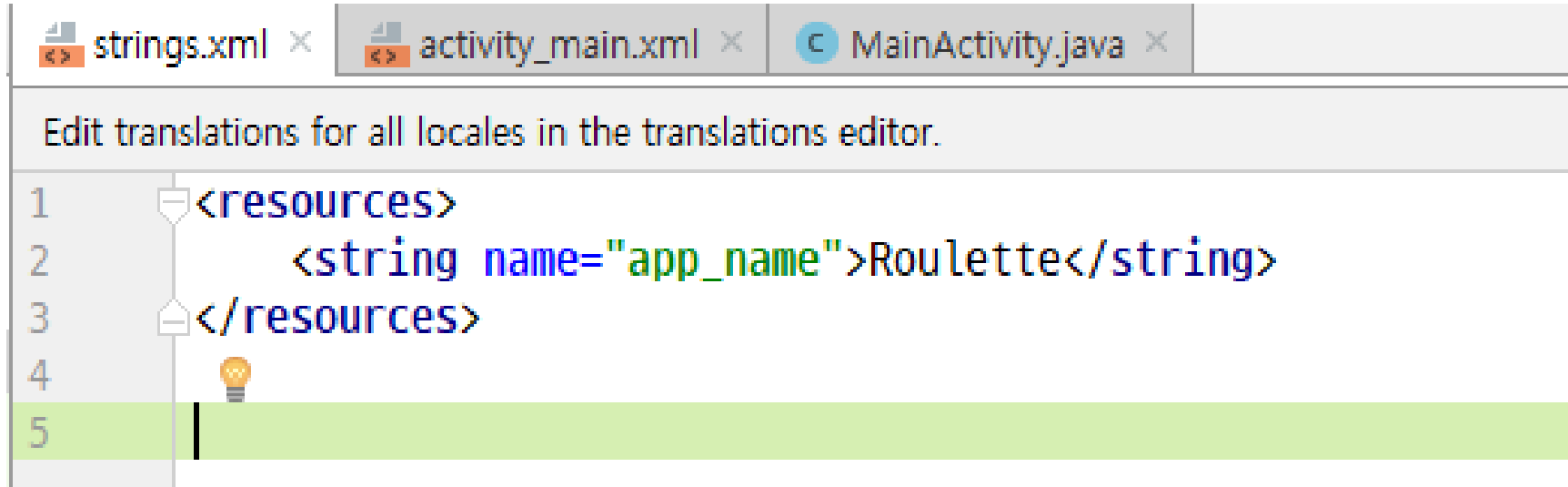
Step 2.2 res/drawable에 이미지 붙여넣기

12



Step 2.3 텍스트 자원의 편집

- strings.xml

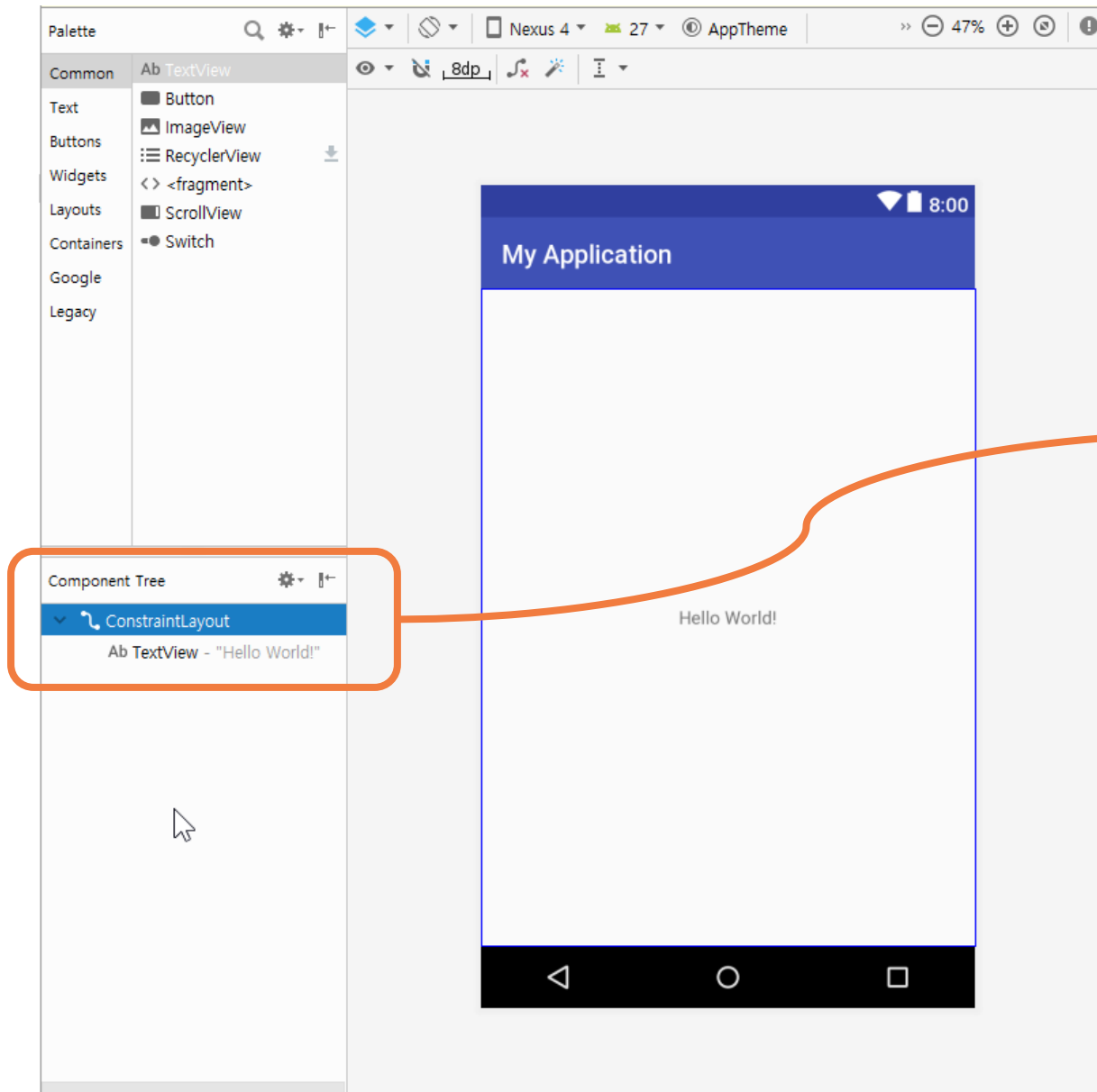


The screenshot shows an IDE window with three tabs: `strings.xml`, `activity_main.xml`, and `MainActivity.java`. The `strings.xml` tab is active, displaying the XML content for editing translations. A message at the top states: "Edit translations for all locales in the translations editor." The XML code is as follows:

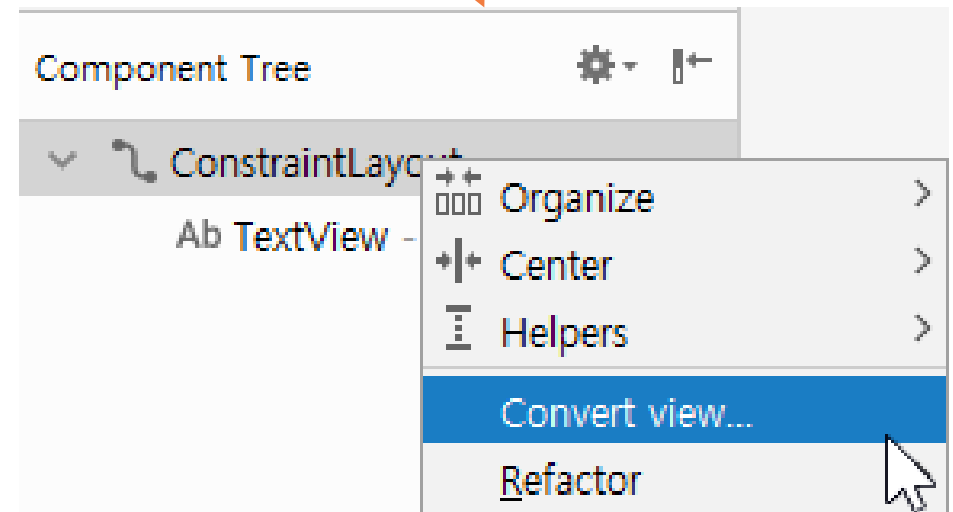
```
1 <resources>
2   <string name="app_name">Roulette</string>
3 </resources>
```

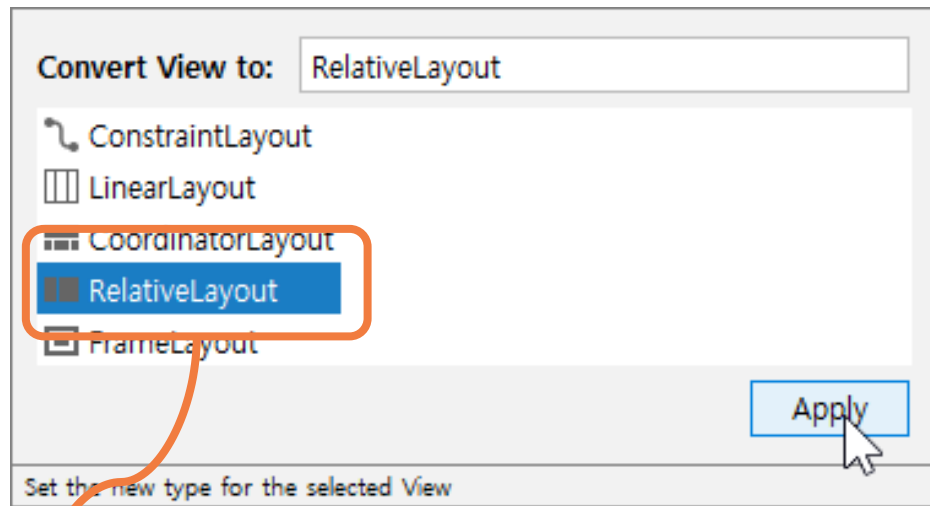
Line 4 is empty and contains a lightbulb icon, indicating a suggestion or tip. Line 5 is highlighted in green and is currently empty.

2.4 화면 설계

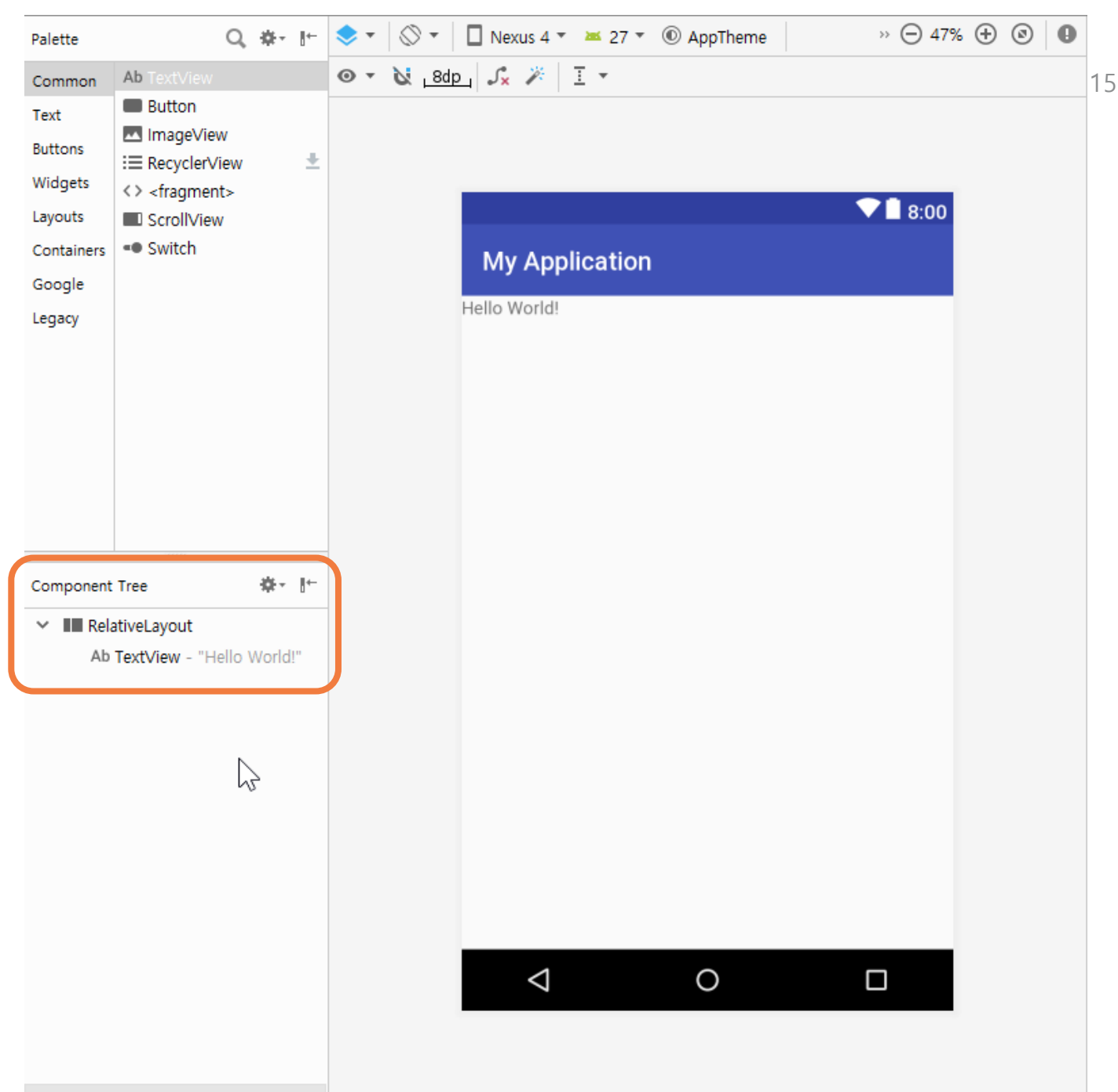


ConstraintLayout →
RelativeLayout으로 변경

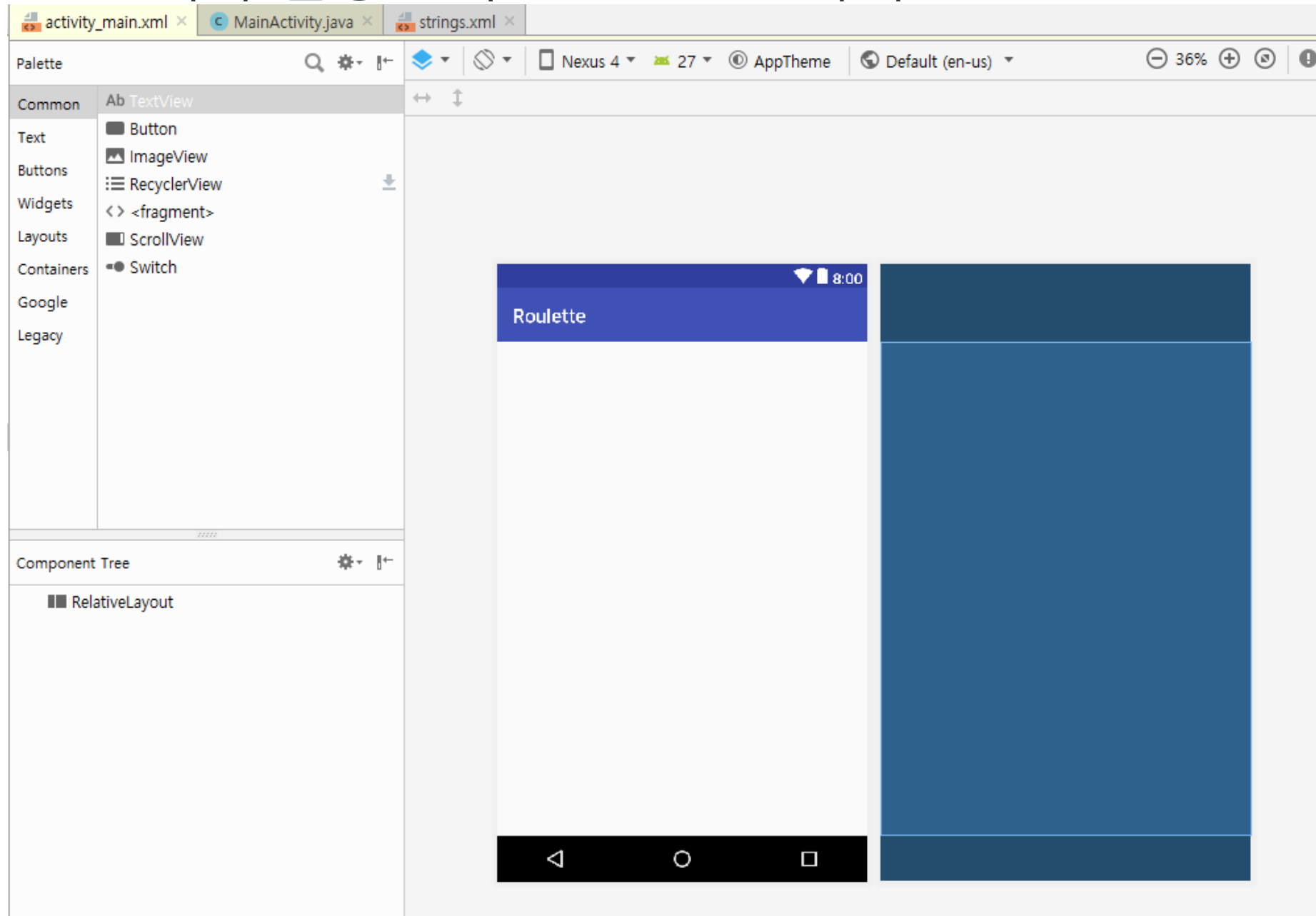




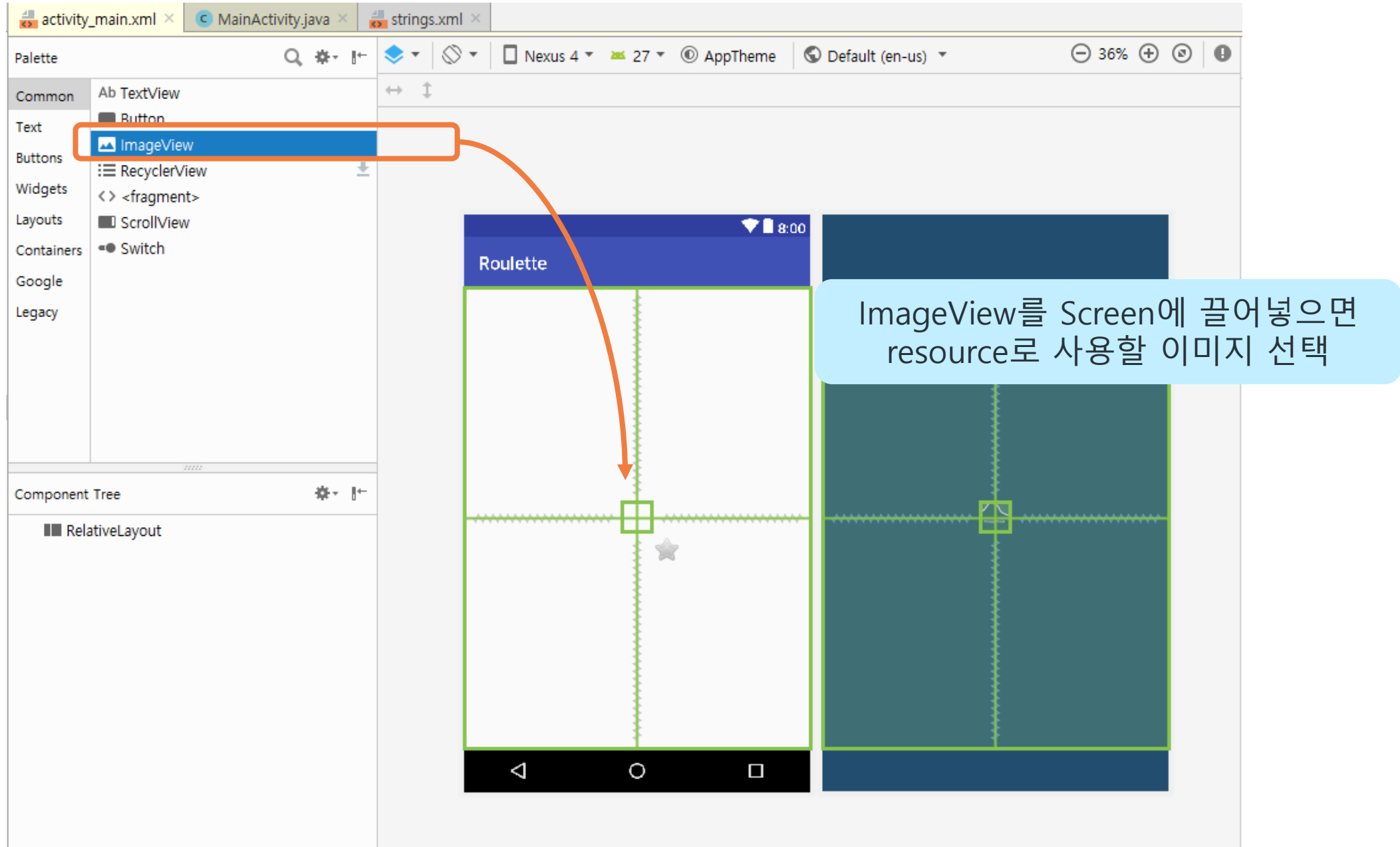
RelativeLayout 선택



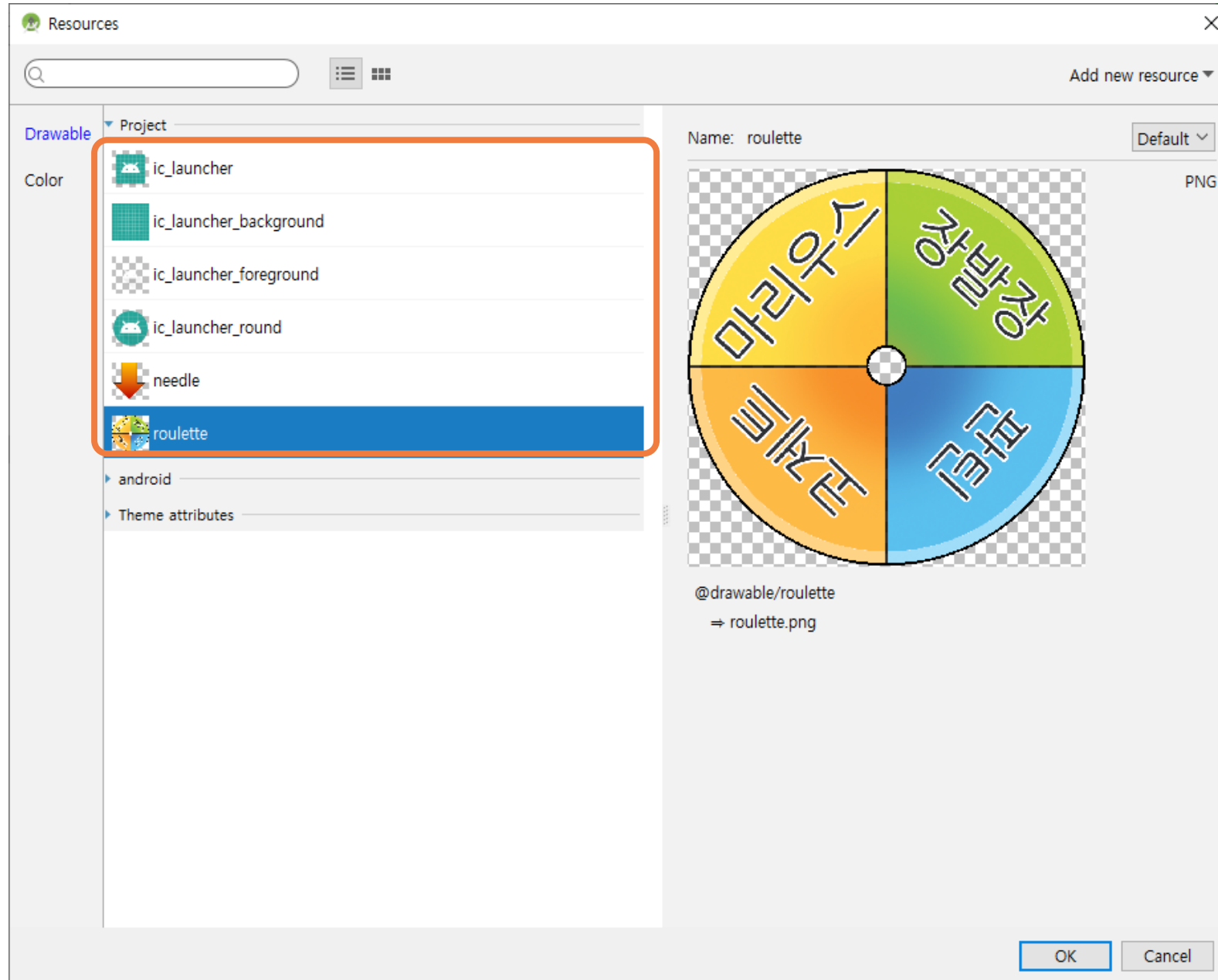
• Layout 초기화 설정 – 기본 TextView 삭제



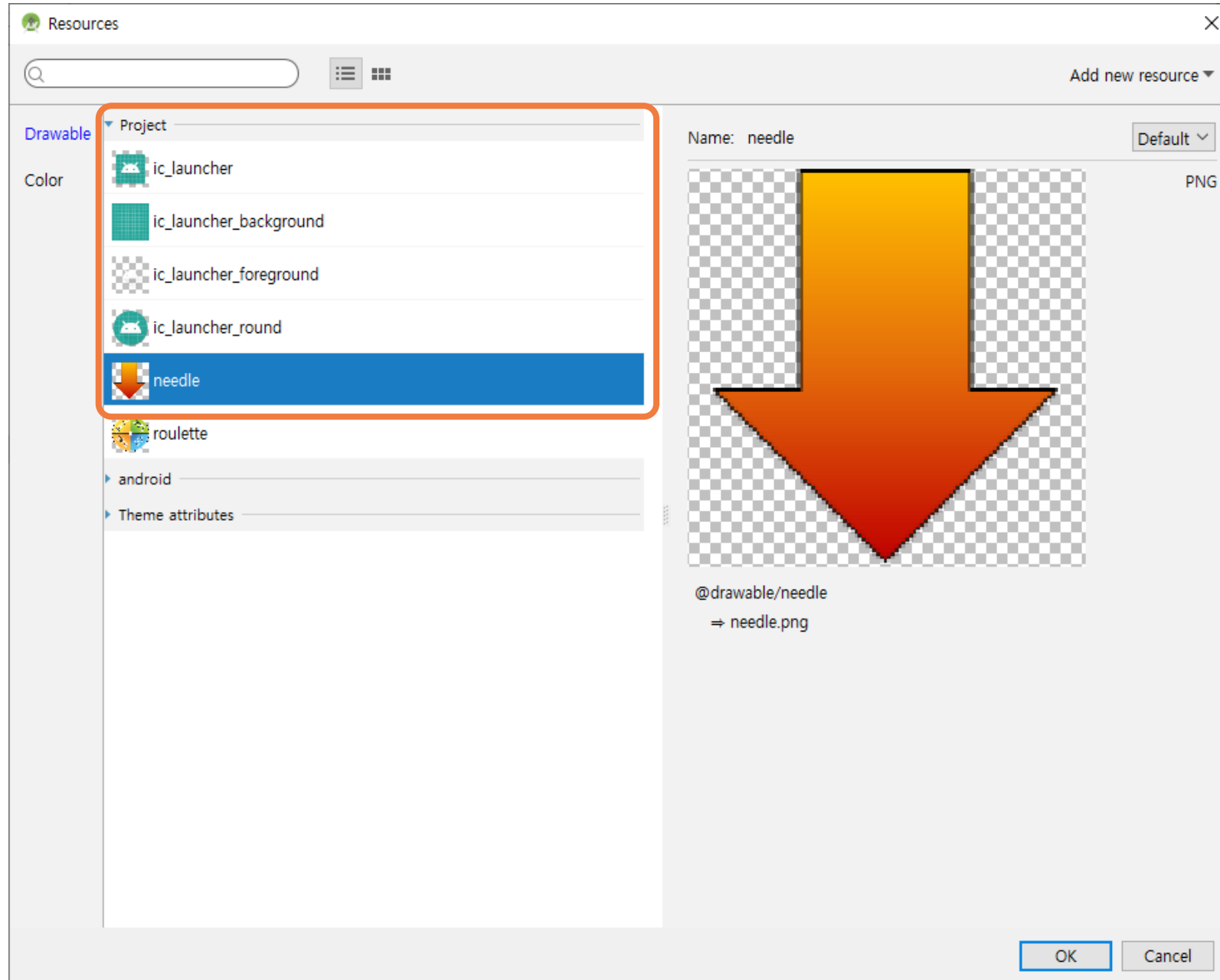
• ImageView 컴포넌트 삽입하기



- Drawable리소스에서 roulette 이미지 선택



- Drawable리소스에서 needle이미지 선택



• ImageView – needle.png 속성 설정

The screenshot displays the Android Studio IDE. On the left, a preview of a roulette application is shown. The roulette wheel is divided into four colored segments: yellow (top-left), green (top-right), orange (bottom-left), and blue (bottom-right). Each segment contains Korean text. A red arrow points to the top of the wheel. Below the wheel is a black bar with three white icons. The top of the screen shows a status bar with the time 8:00 and a battery icon. On the right, the 'Attributes' panel for the 'needle' ImageView is visible. It lists various properties and their values. A blue callout box points to the 'id', 'layout_width', and 'layout_height' attributes. Another blue callout box points to the 'layout_alignParentTop' and 'layout_centerHorizontal' attributes. The 'srcCompat' attribute is set to '@drawable/needle'.

id: needle
layout_width: match_parent
layout_height: wrap_content

layout_alignParentTop: check
layout_centerHorizontal: check

| Attribute | Value |
|-------------------------|-------------------------------------|
| 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 | |
| autofillHint | |
| background | |
| backgroundTint | |
| backgroundTintMode | |
| 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 interface with a Nexus 4 virtual device. The roulette.png image is displayed on the left, and the ImageView properties are shown on the right. The roulette.png image is a circular graphic divided into four quadrants with Korean text: '마리아시' (Maria), '장바탕' (Jangbang), '복사본' (Boksabon), and '파워' (Power). A large orange arrow points down to the roulette.png image.

id: roulette
layout_width: wrap_content
layout_height: wrap_content

| 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 | |
| backgroundTint | |
| backgroundTintMode | |
| 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 | |
| filterTouchesWhenObscured | <input type="checkbox"/> |

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", 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

```
strings.xml × activity_main.xml × MainActivity.java ×
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, "rotation", 0.0f, 405f);
20
21         object.setInterpolator(new BounceInterpolator());
22         object.setDuration(10000);
23         object.start();
24     }
25 }
26
27
```

클래스와 속성/메소드

- 클래스

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

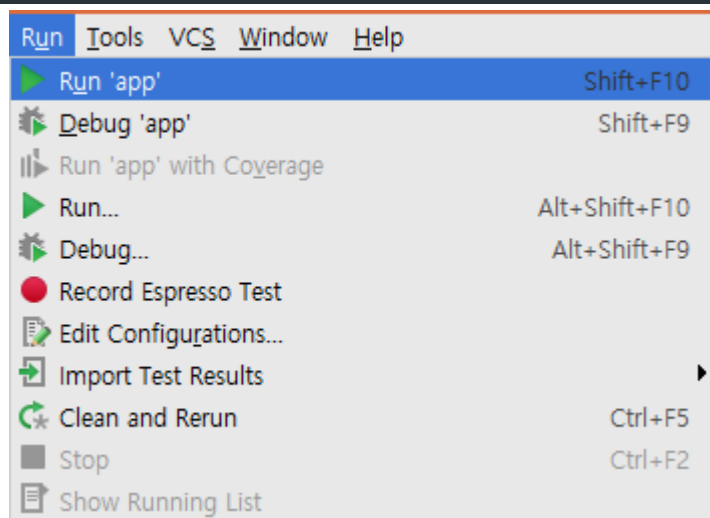
- 메소드

| 클래스 | 메소드 | 설명 | | | | | | | | |
|--|---|--|------|----|--------|----------|--------------|-------------|--------|-------------------|
| ObjectAnimator | static ObjectAnimator ofFloat(Object target, String propertyName, float ...values) | values 사이의 애니메이션을 만들고 ObjectAnimator를 반환함 <table><tr><th>매개변수</th><th>설명</th></tr><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></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 | | | | | | | | | |
| ObjectAnimator setDuration(long duration) | 애니메이션 시간 설정, 밀리초 단위이며, 기본값은 300밀리초로 설정됨 | | | | | | | | | |
| void start() | 애니메이션 시작 | | | | | | | | | |

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

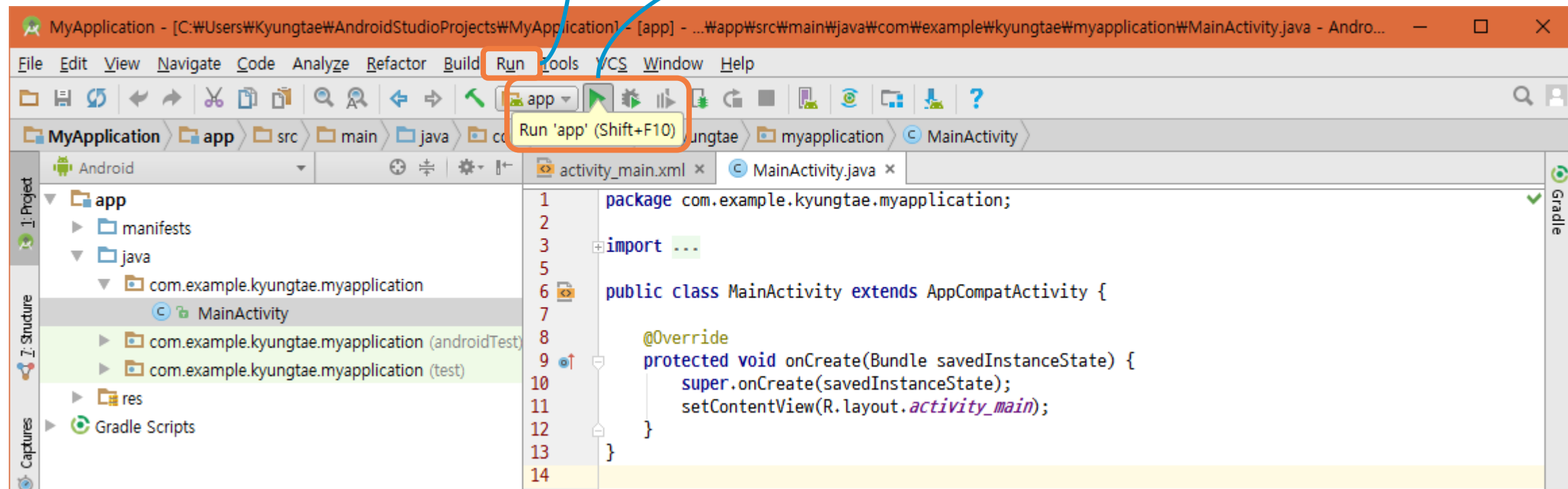
Step 3. 프로젝트 실행

26



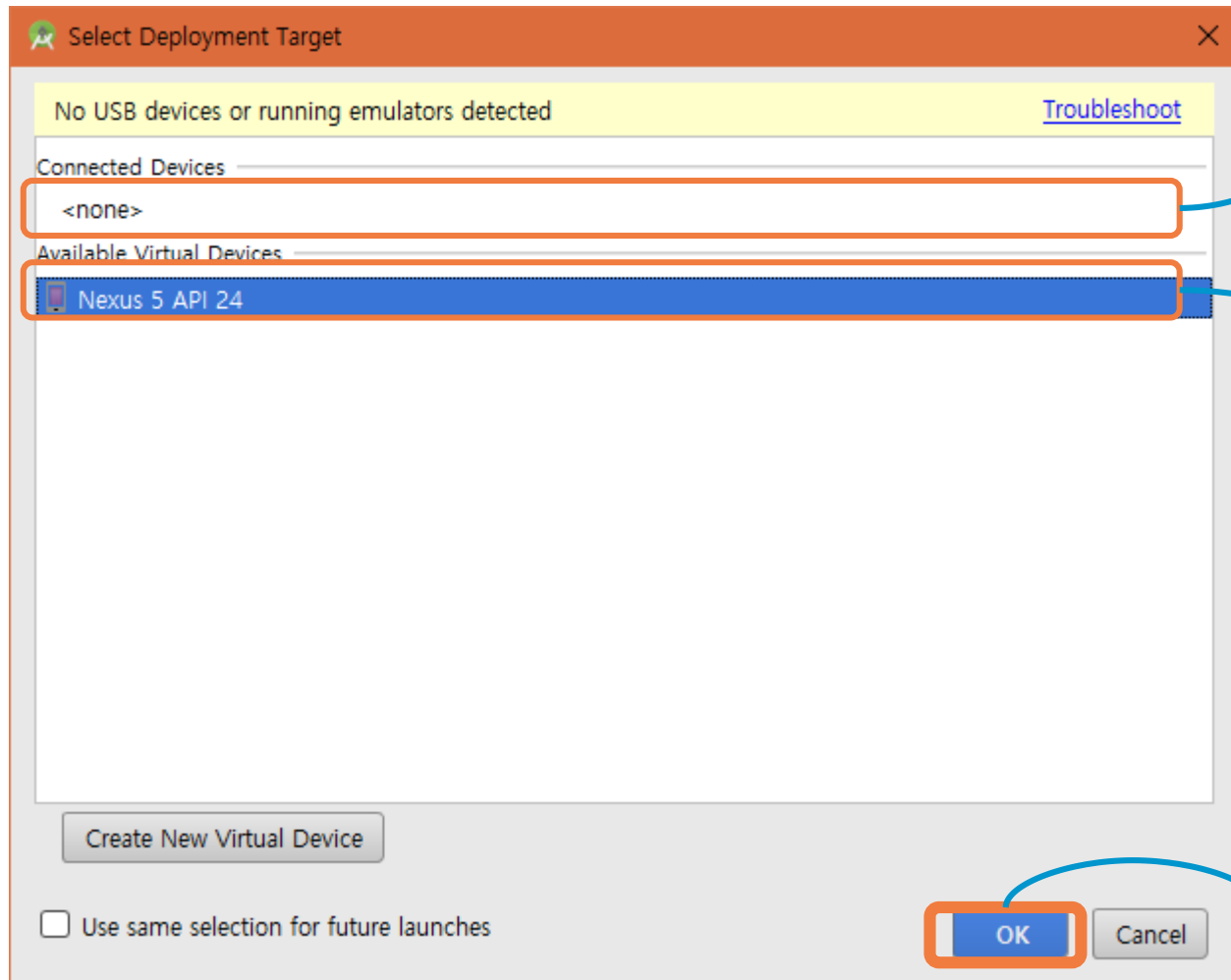
Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기

27



데이터 케이블로 연결된
스마트폰

AVD

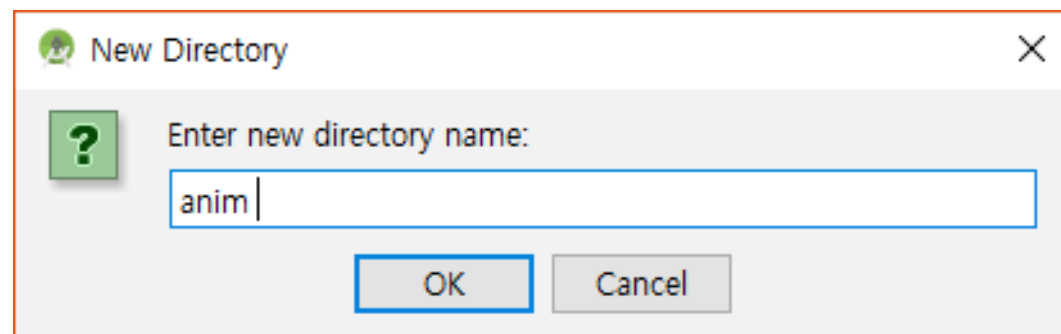
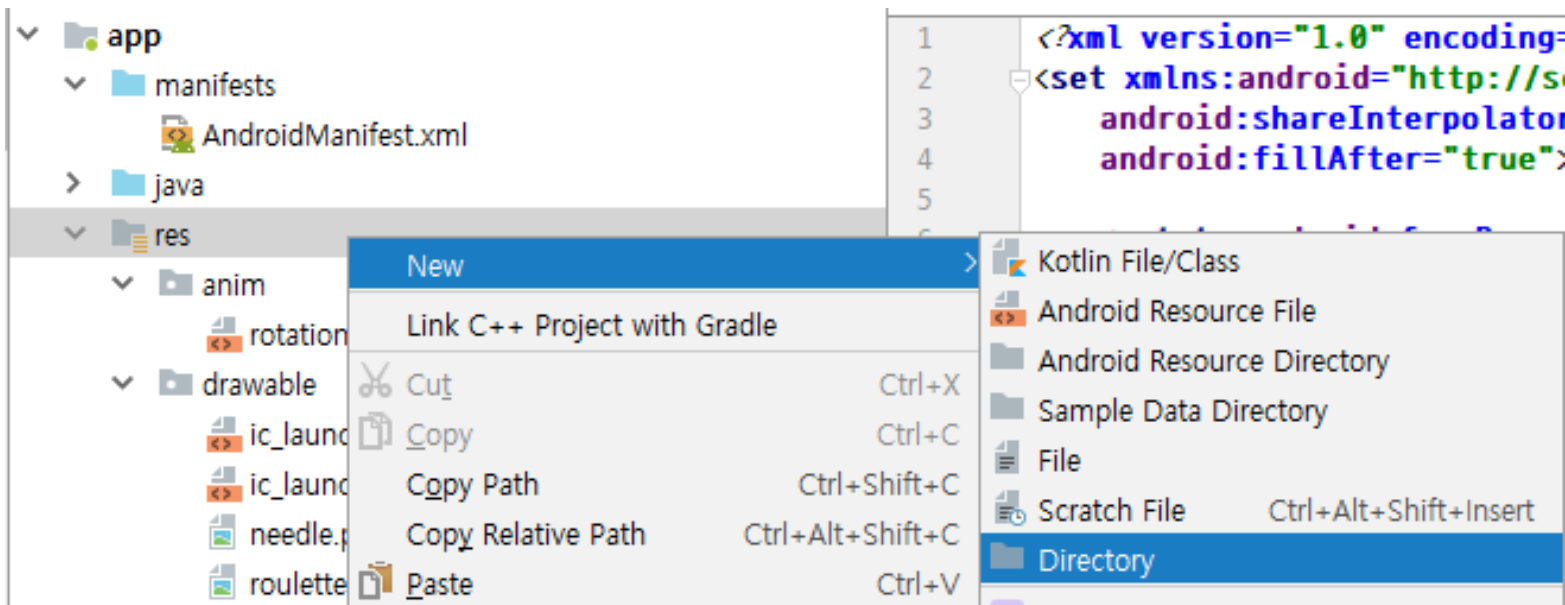
스마트폰 또는 AVD를 선택하고
'OK' 버튼을 클릭



Animation 객체를 이용한 애니메이션

Animation 객체 추가 – res/anim 폴더 만들기

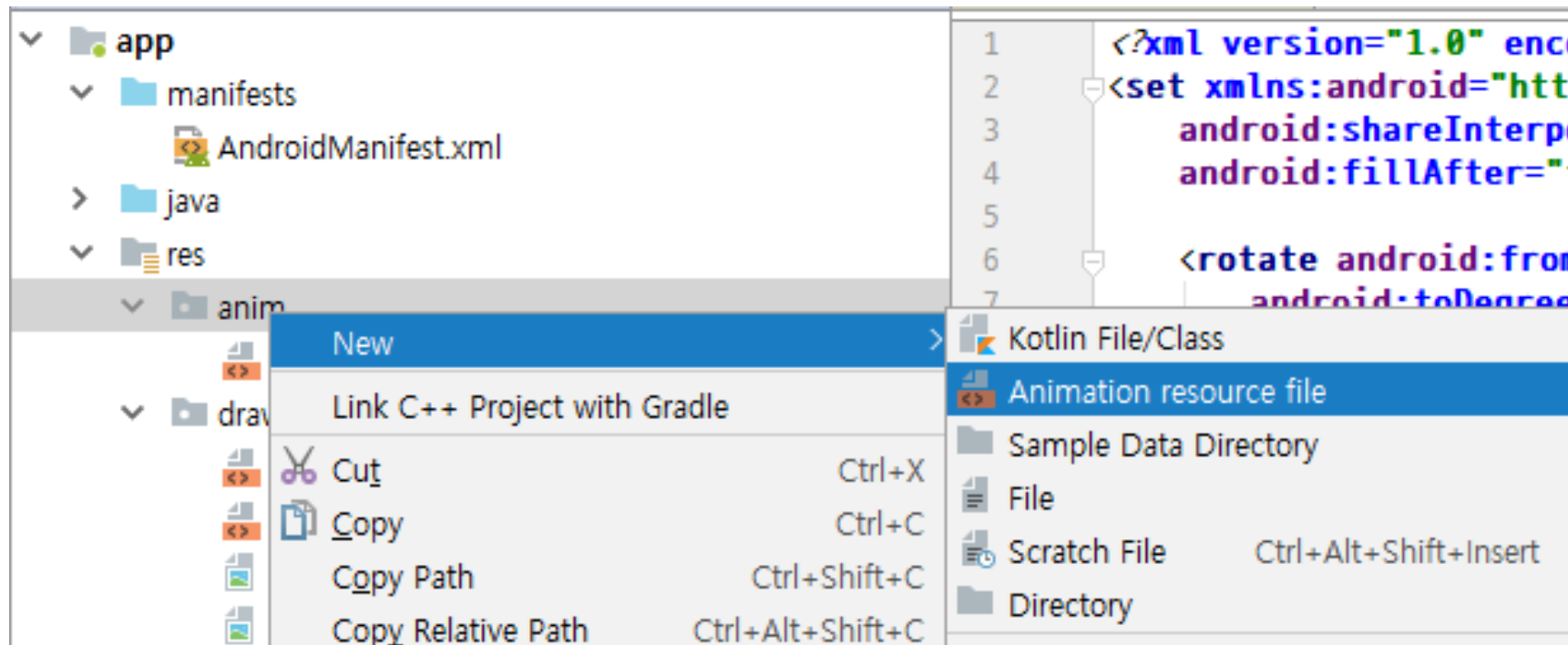
- 애니메이션 설정을 위한 xml 파일 생성



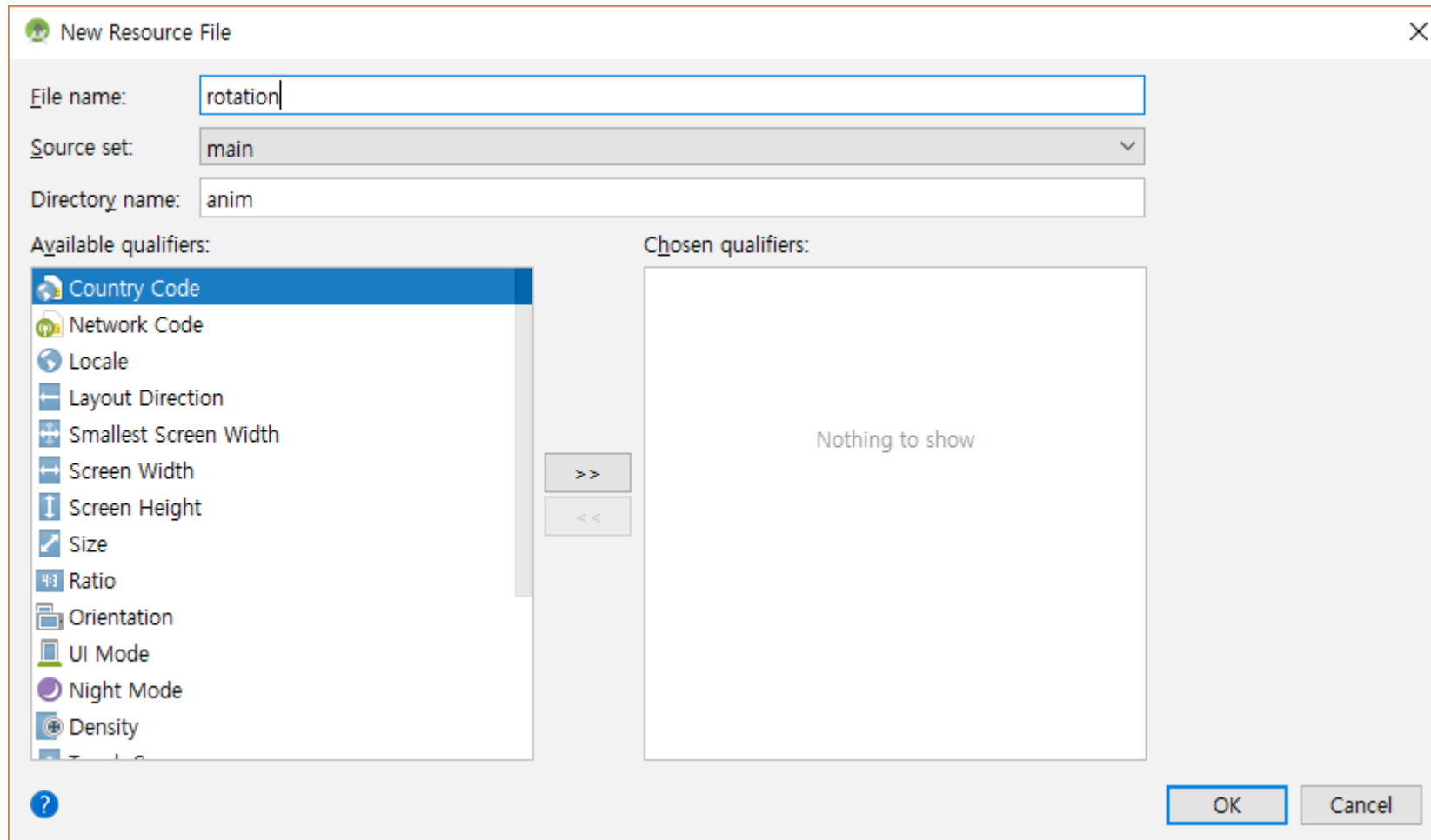
Animation 객체 추가 – rotation.xml 파일 만들기

30

- 애니메이션 설정을 위한 xml 파일 생성



rotation.xml 파일 만들기



Animation 객체 추가 – rotation.xml

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

- Project Explorer (Left):** Displays the project structure. The `anim` folder under `res` is expanded, showing `rotation.xml` selected.
- Editor (Right):** Displays the content of `rotation.xml`. The XML code is as follows:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <set xmlns:android="http://schemas.android.com/apk/res/android"
3   android:shareInterpolator="@android:anim/accelerate_decelerate_interpolator"
4   android:fillAfter="true">
5
6   <rotate android:fromDegrees="0"
7     android:toDegrees="764"
8     android:pivotX="50%"
9     android:pivotY="50%"
10    android:duration="10000"
11    android:repeatCount="0" />
12 </set>
```

A callout box with an orange border highlights the `android:shareInterpolator` and `android:fillAfter="true"` attributes. A blue arrow points from this box to a light blue callout box containing the following text:

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

MainActivity.java

33



```
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(context: this, R.anim.rotation);
19         iv.startAnimation(animation);
20     }
21 }
22
23
```

