

Week08.
애니메이션



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://thinkdata.co.kr/wordpress/?page_id=29

애니메이션 활용 앱의 예

4



(a) 겨울왕국



(b) 클래시 오브 클랜

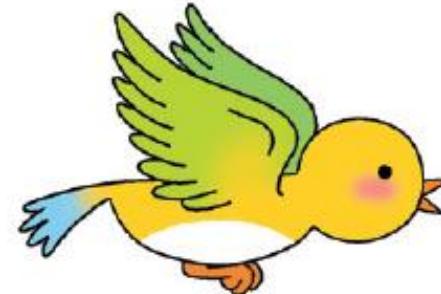


애니메이션 원리

5



회전



이동/크기 변화



프레임1



프레임2



프레임3

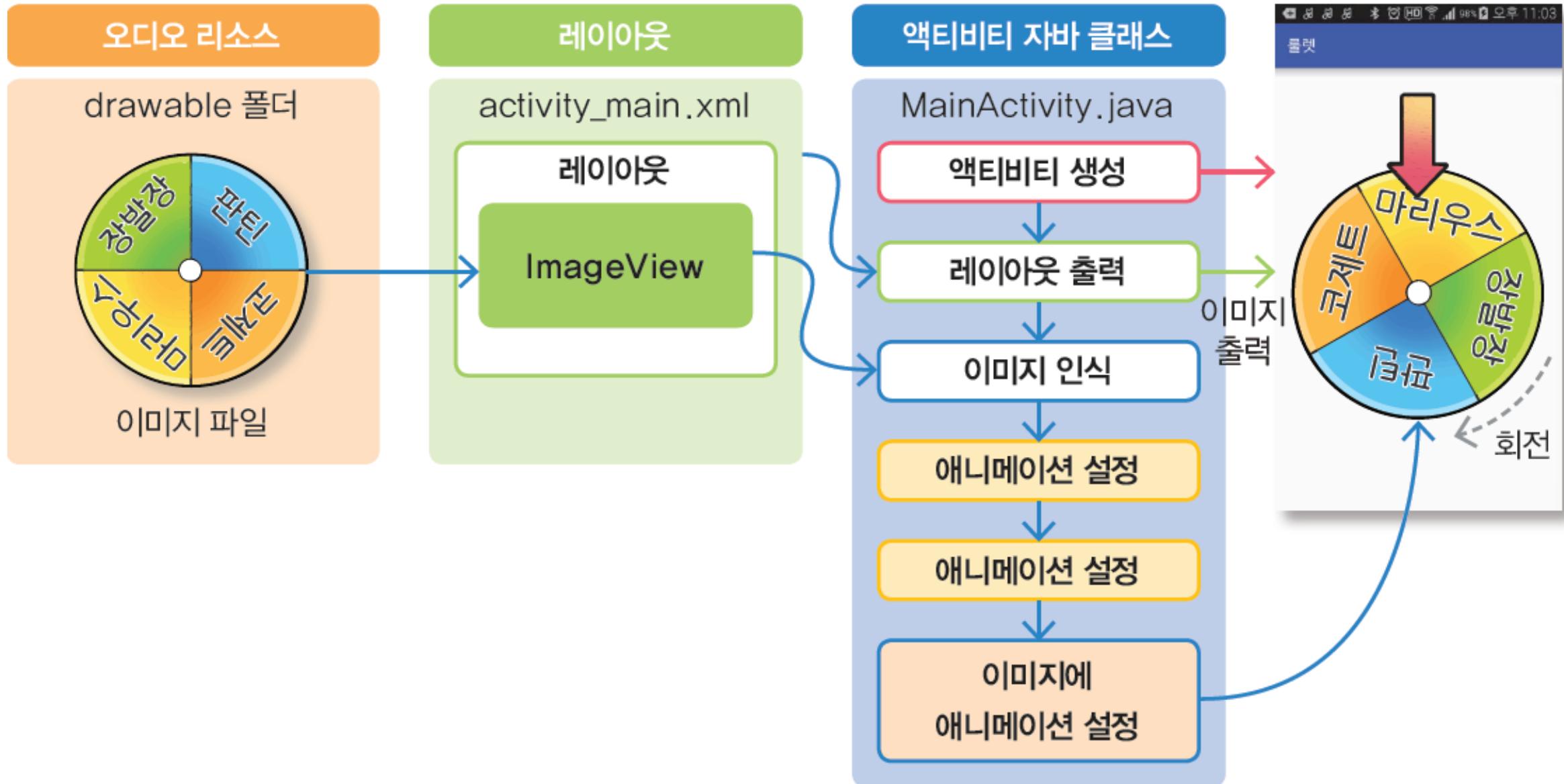


프레임4

움직이는 동작

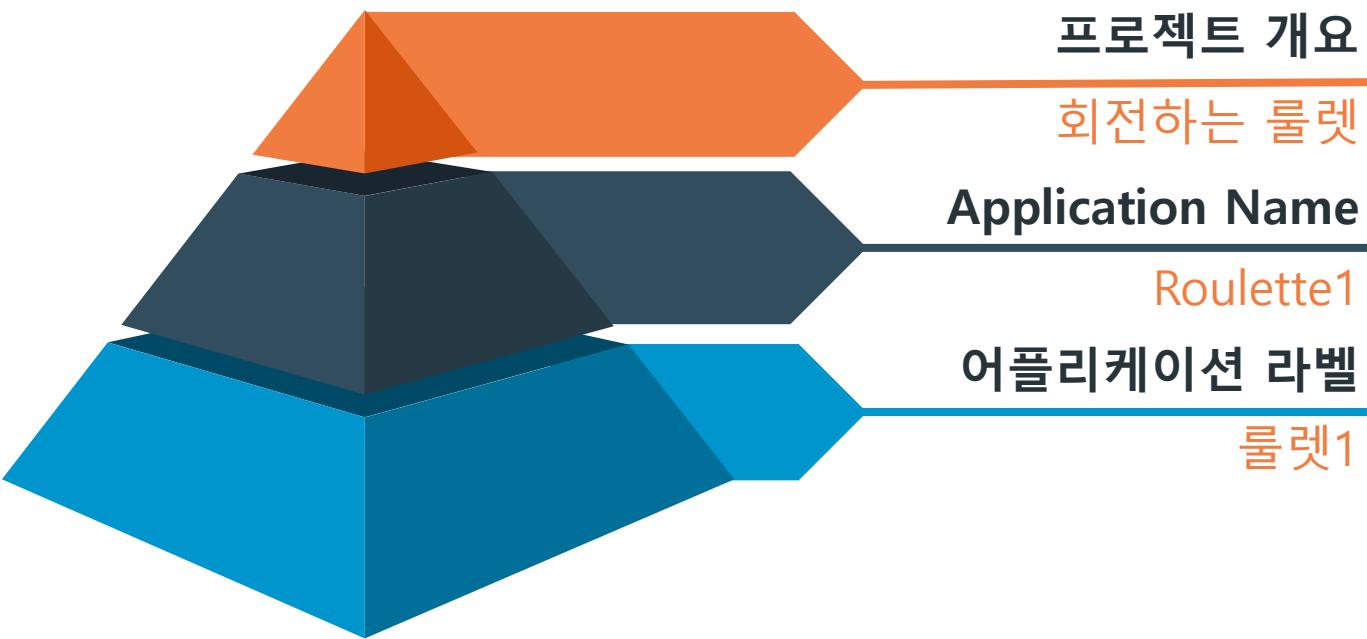
애니메이션 관련 자바 클래스 이용

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: Roulette1 Company Domain: kyungtae.example.com (디폴트 사용) Project location: ~\user00\AndroidStudioProjects\ktpark\Roulette1
③ 제품 형태	Phone and Tablet(사용할 안드로이드 버전 지정: Android 7.0 Nougat)
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

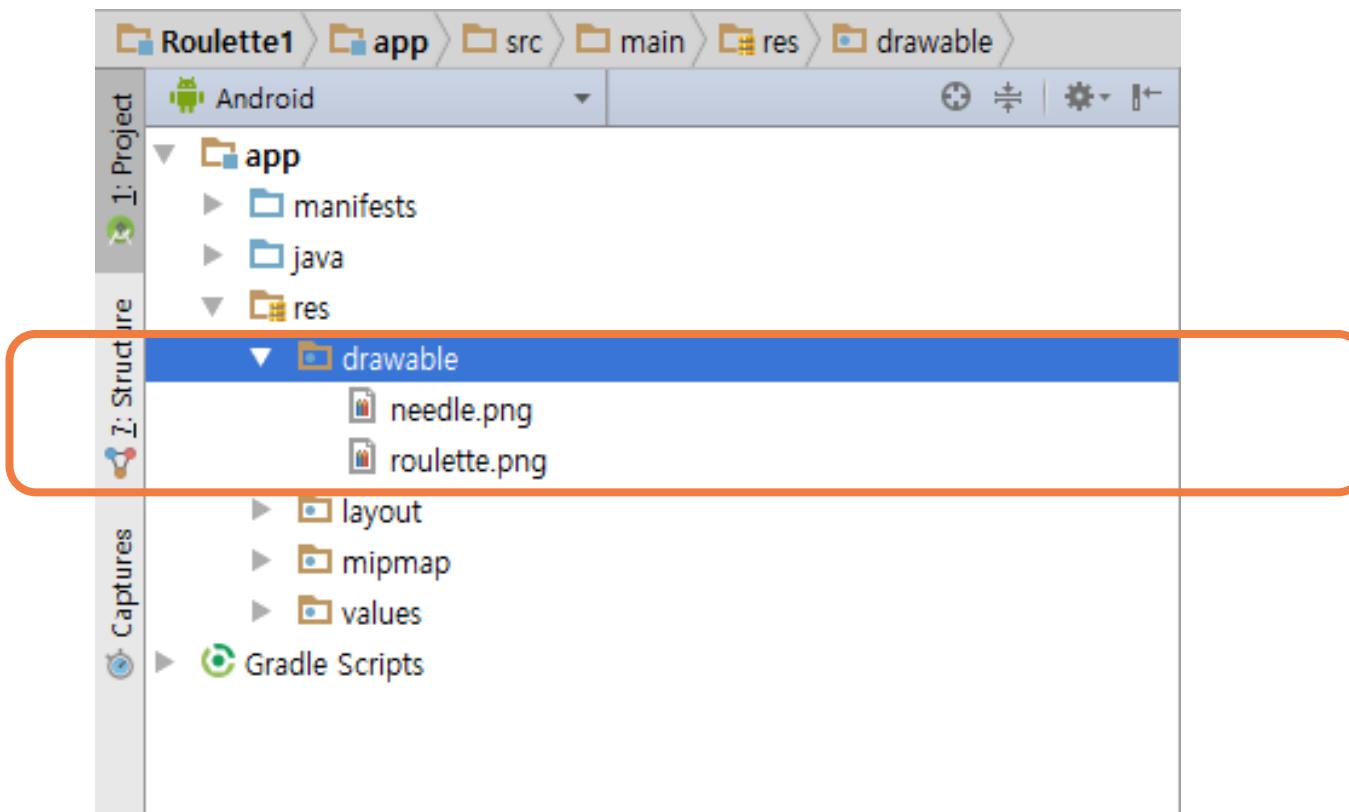
11

- 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

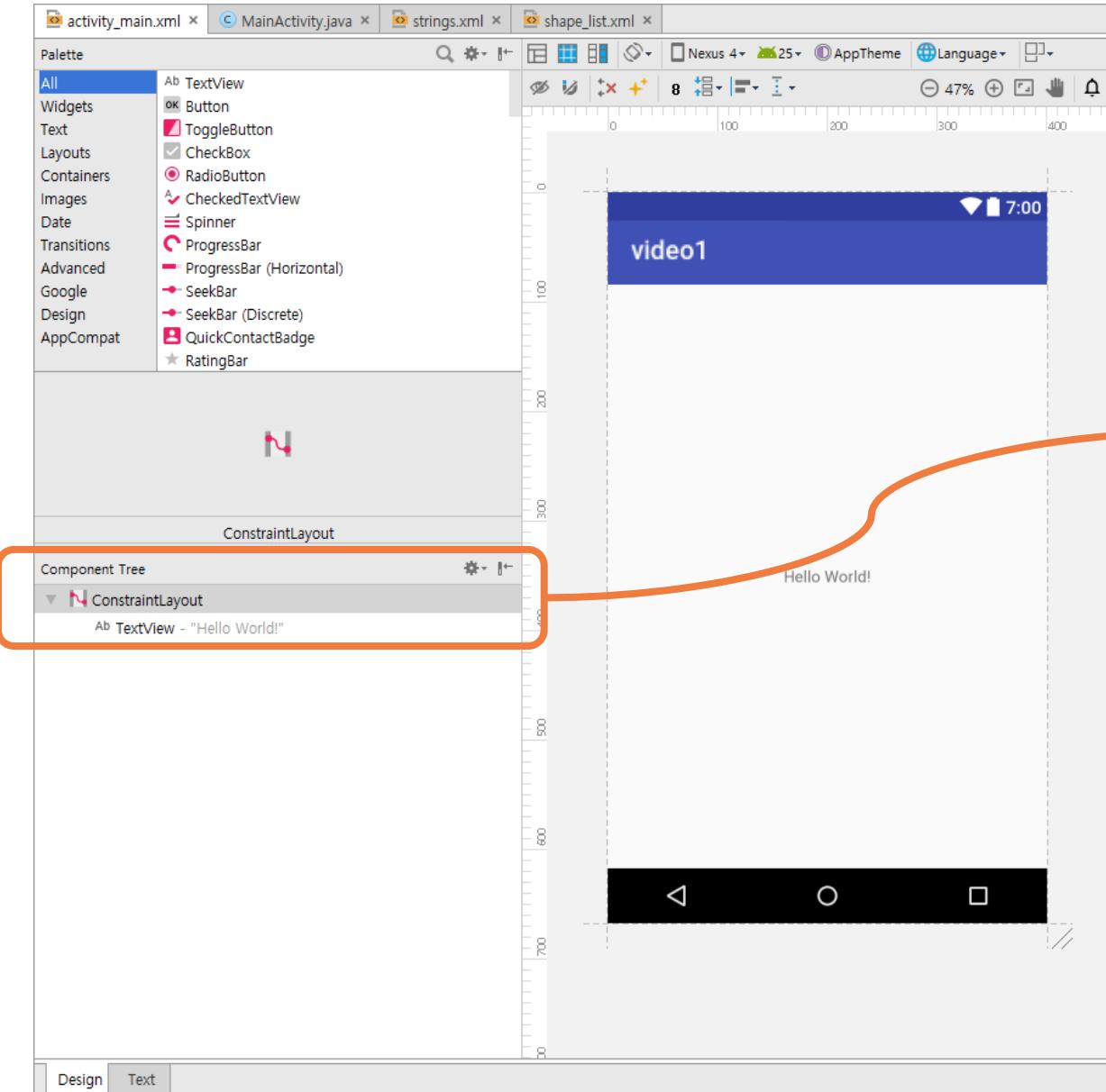
- strings.xml

The screenshot shows the Android Studio interface with the tabs bar at the top containing activity_main.xml, strings.xml (which is selected), rotation.xml, dimens.xml, and MainActivity.java. Below the tabs is a message: "Edit translations for all locales in the translations editor." The main area displays the XML code for strings.xml:

```
1 <resources>
2   <string name="app_name">Roulette1</string>
3 </resources>
4
5 |
```

The code is color-coded: blue for tags like <resources>, <string>, and </resources>, green for attributes like name="app_name", and red for the string value "Roulette1". A yellow lightbulb icon is positioned next to the closing </resources> tag. The fifth line, which is empty, has a cursor and is highlighted with a yellow background.

2.4 화면 설계



ConstraintLayout →
RelativeLayout으로 변경

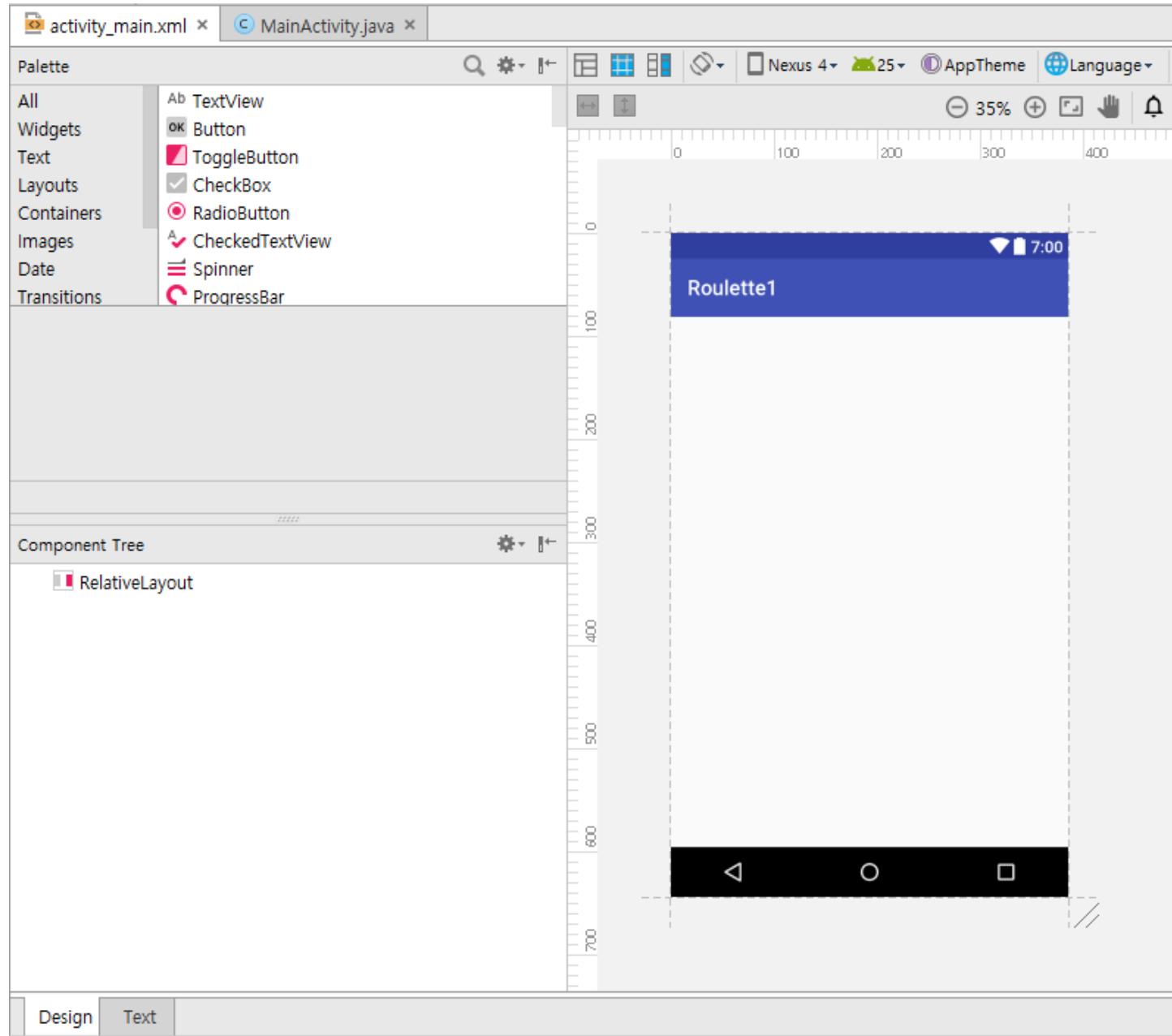
Text 에디터에서 수정

```
activity_main.xml x MainActivity.java x strings.xml x shape_list.xml x  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <android.support.constraint.ConstraintLayout  
3     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  
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 </android.support.constraint.ConstraintLayout>
```

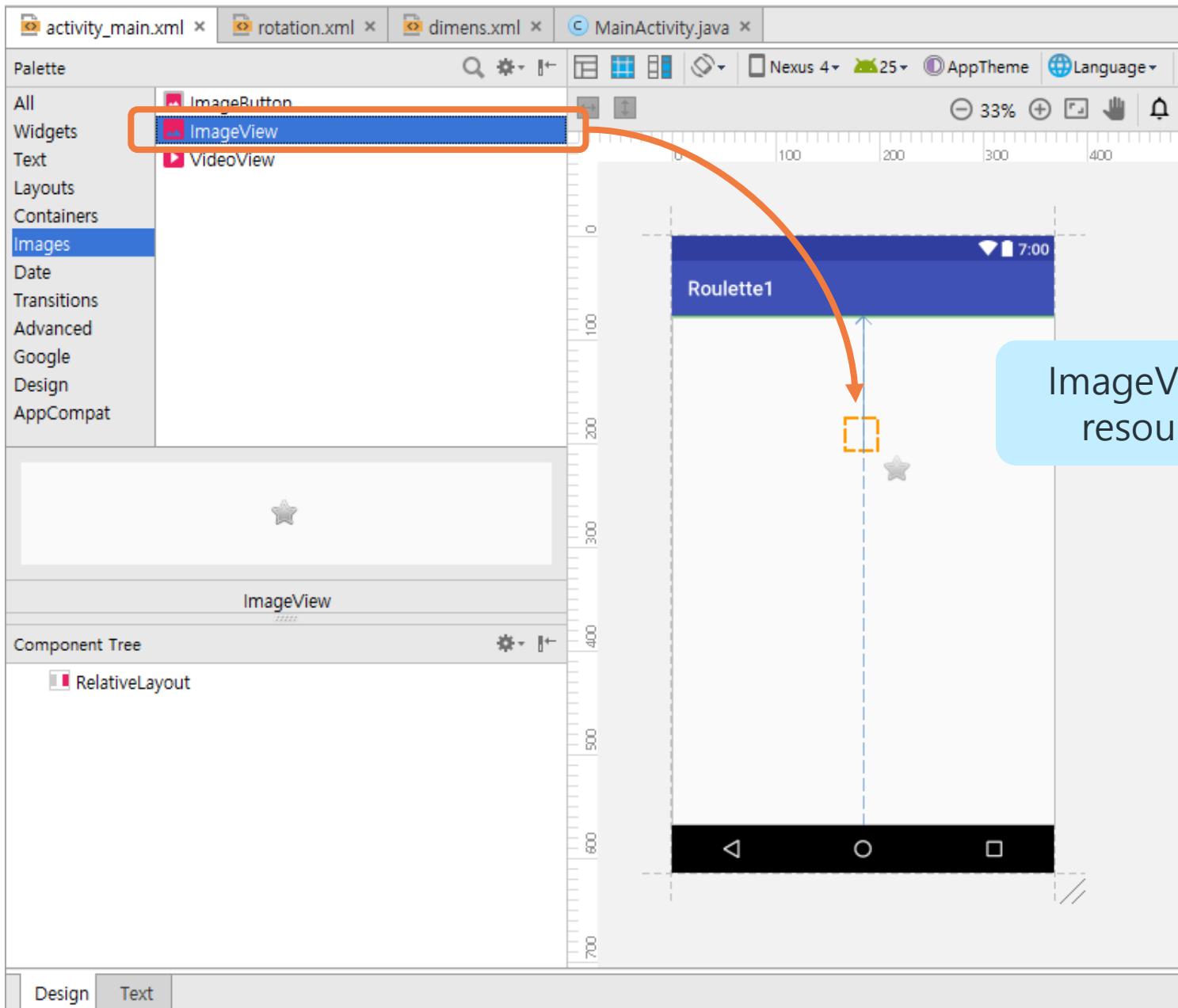
```
activity_main.xml x MainActivity.java x  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <RelativeLayout  
3     xmlns:android="http://schemas.android.com/apk/res/android"  
4     xmlns:app="http://schemas.android.com/apk/res-auto"  
5     xmlns:tools="http://schemas.android.com/tools"  
6     android:layout_width="match_parent"  
7     android:layout_height="match_parent"  
8     tools:context="com.example.kyungtae.roulette1.MainActivity">  
9  
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 </RelativeLayout>
```

삭제

• Layout 초기화 설정 - 기본 TextView 삭제

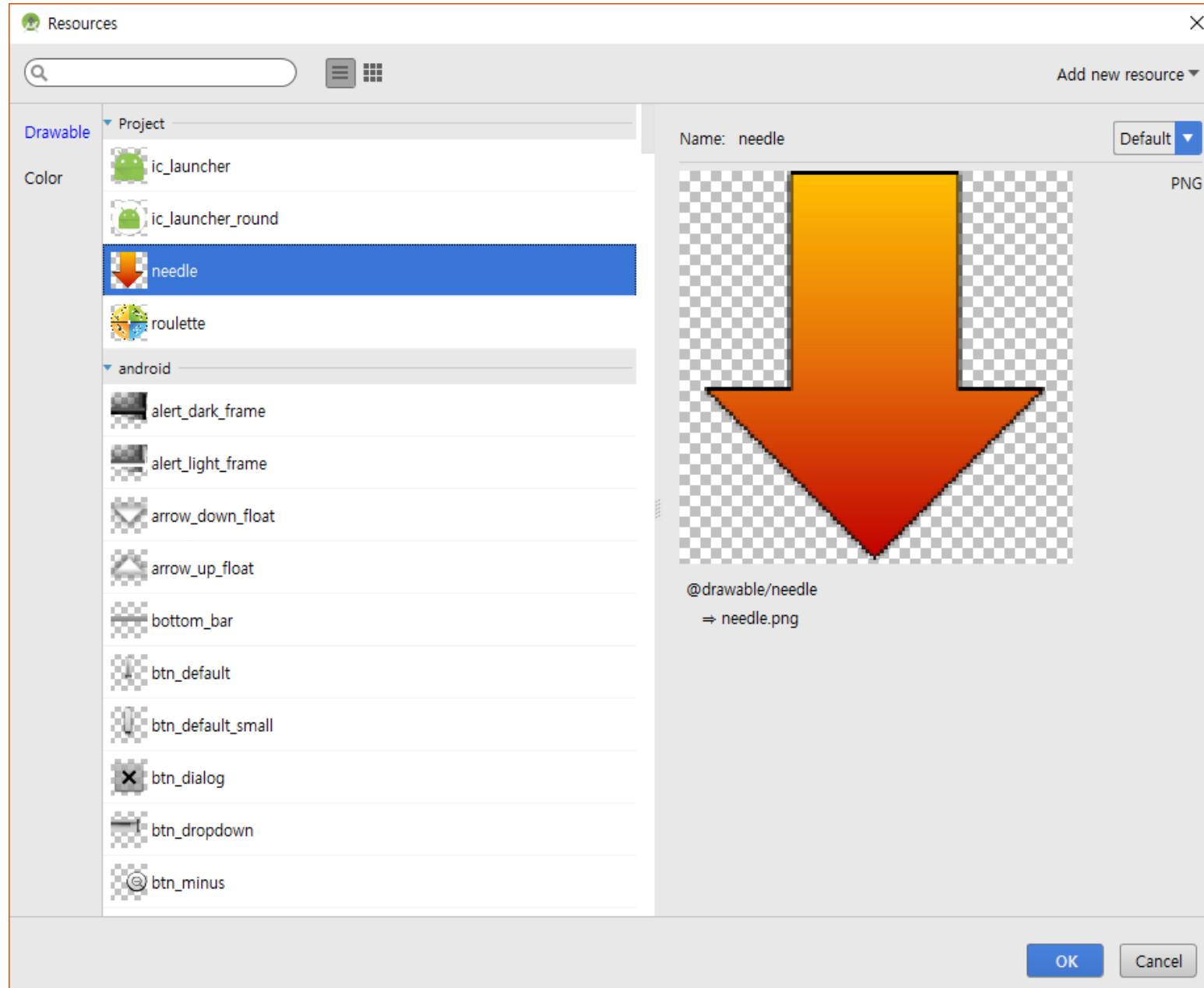


• ImageView 컴포넌트 삽입하기

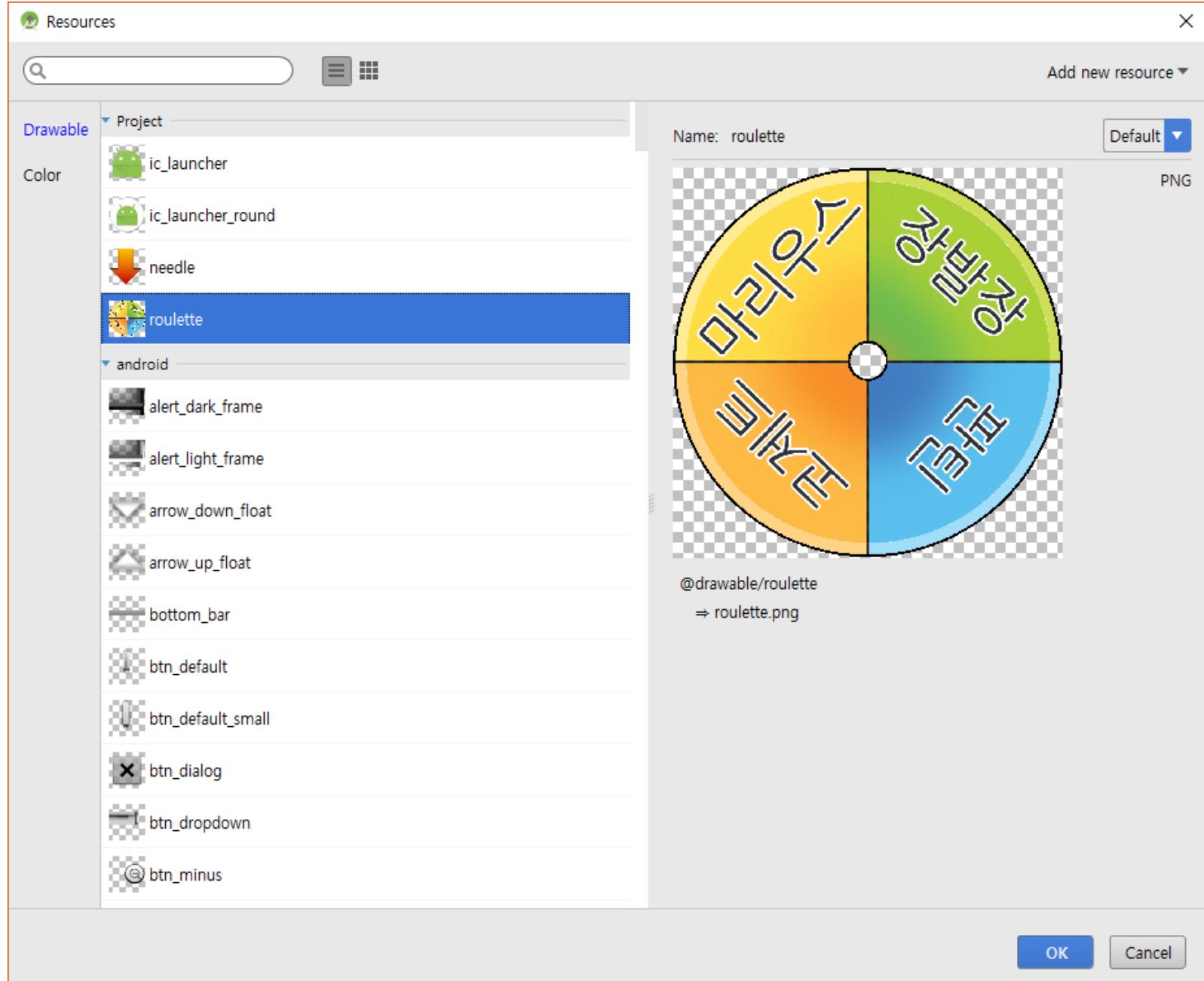


ImageView를 Screen에 끌어넣으면
resource로 사용할 이미지 선택

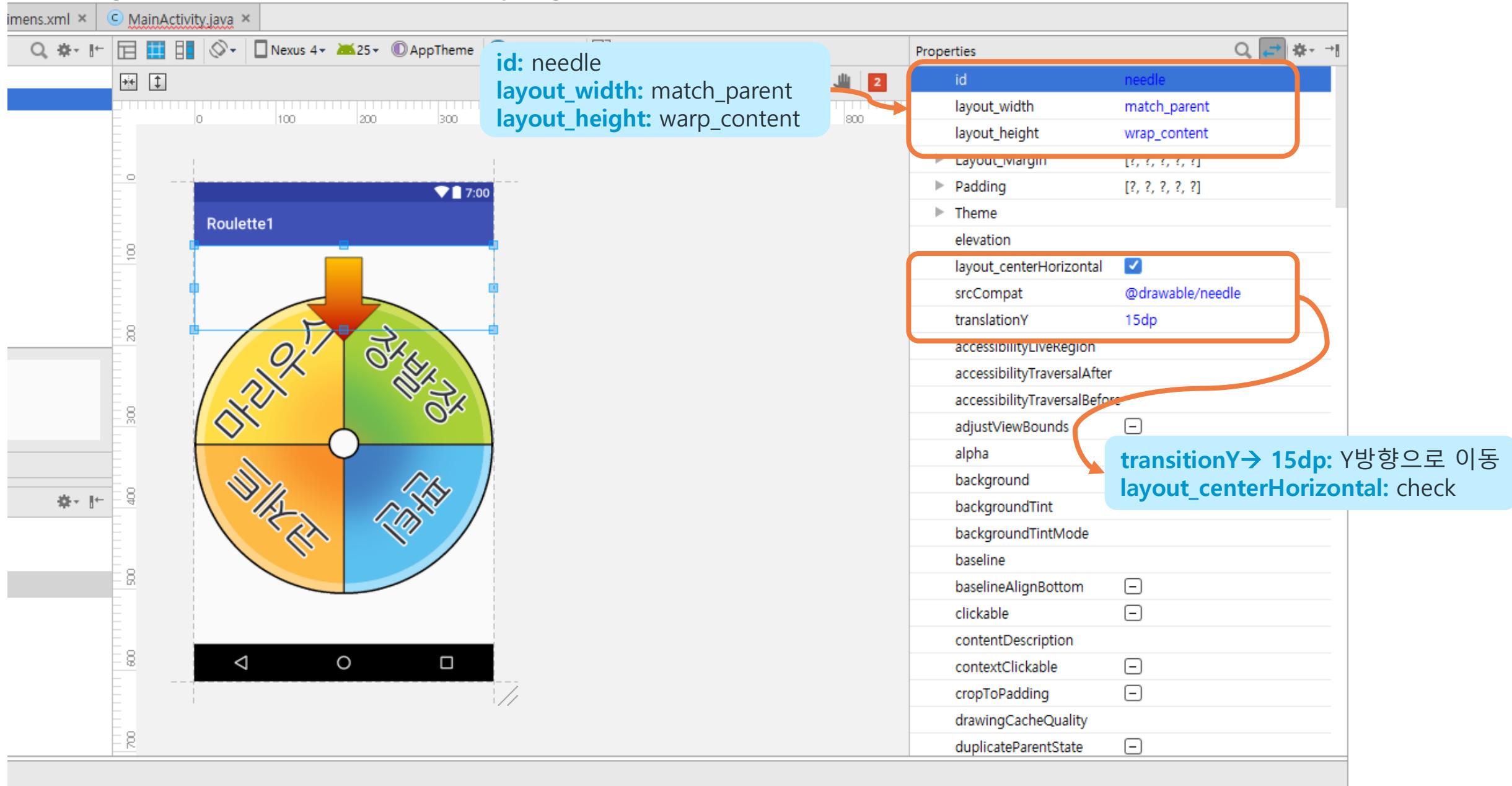
- Drawable리소스에서 needle이미지 선택



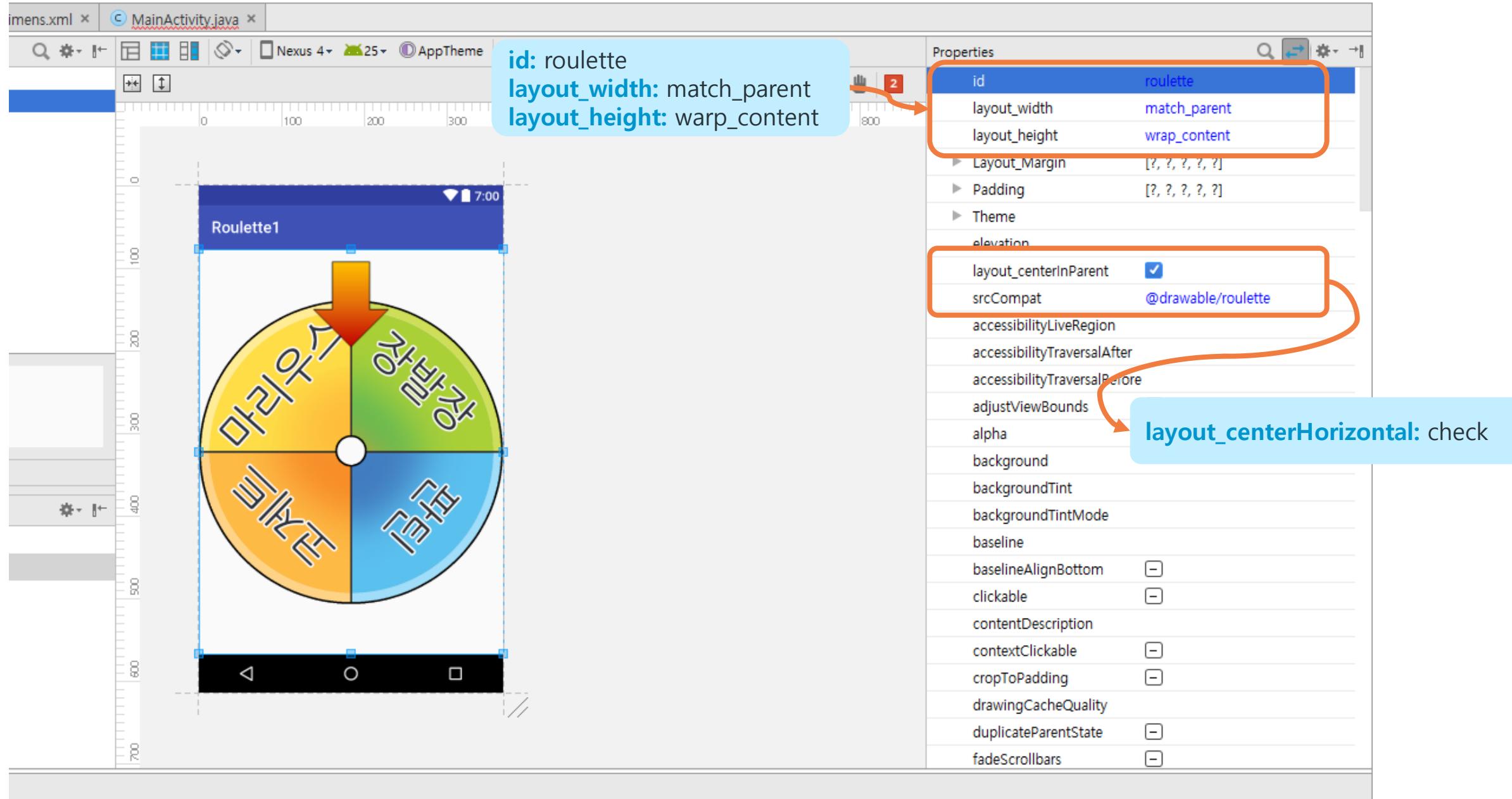
• Drawable리소스에서 roulette 이미지 선택



• ImageView - needle.png 속성 설정



• ImageView - roulette.png 속성 설정



2.5 Activity 제어(MainActivity.java)

22

- 이미지 리소스에 대한 ImageView를 획득과 애니메이션 재생

```
1 package com.example.kyungtae.roulette1;
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
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 ivRoulette = (ImageView) findViewById(R.id.roulette);
18
19         ObjectAnimator object = ObjectAnimator.ofFloat(ivRoulette, "rotation", 0.0f, 405.0f);
20
21         object.setInterpolator(new AccelerateDecelerateInterpolator());
22         object.setDuration(3000);
23         object.start();
24     }
25
26 }
```

이미지뷰 객체 획득

0~405도 회전시키는 회전 객체 생성

애니메이션이 처음 실행될 때는 가속되다가
가속도를 줄여 정지.

클래스와 속성/메소드

• 클래스

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

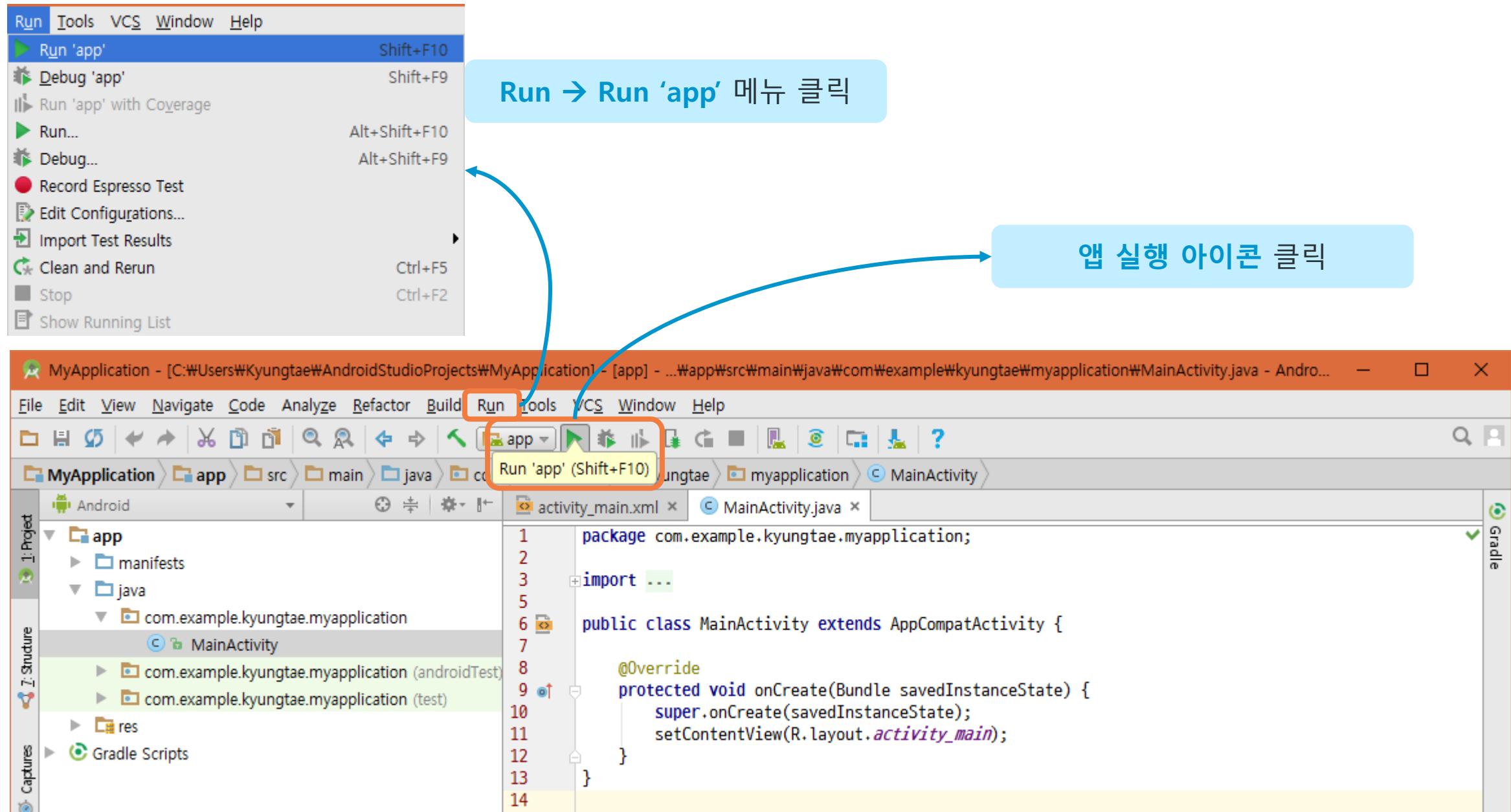
• 메소드

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

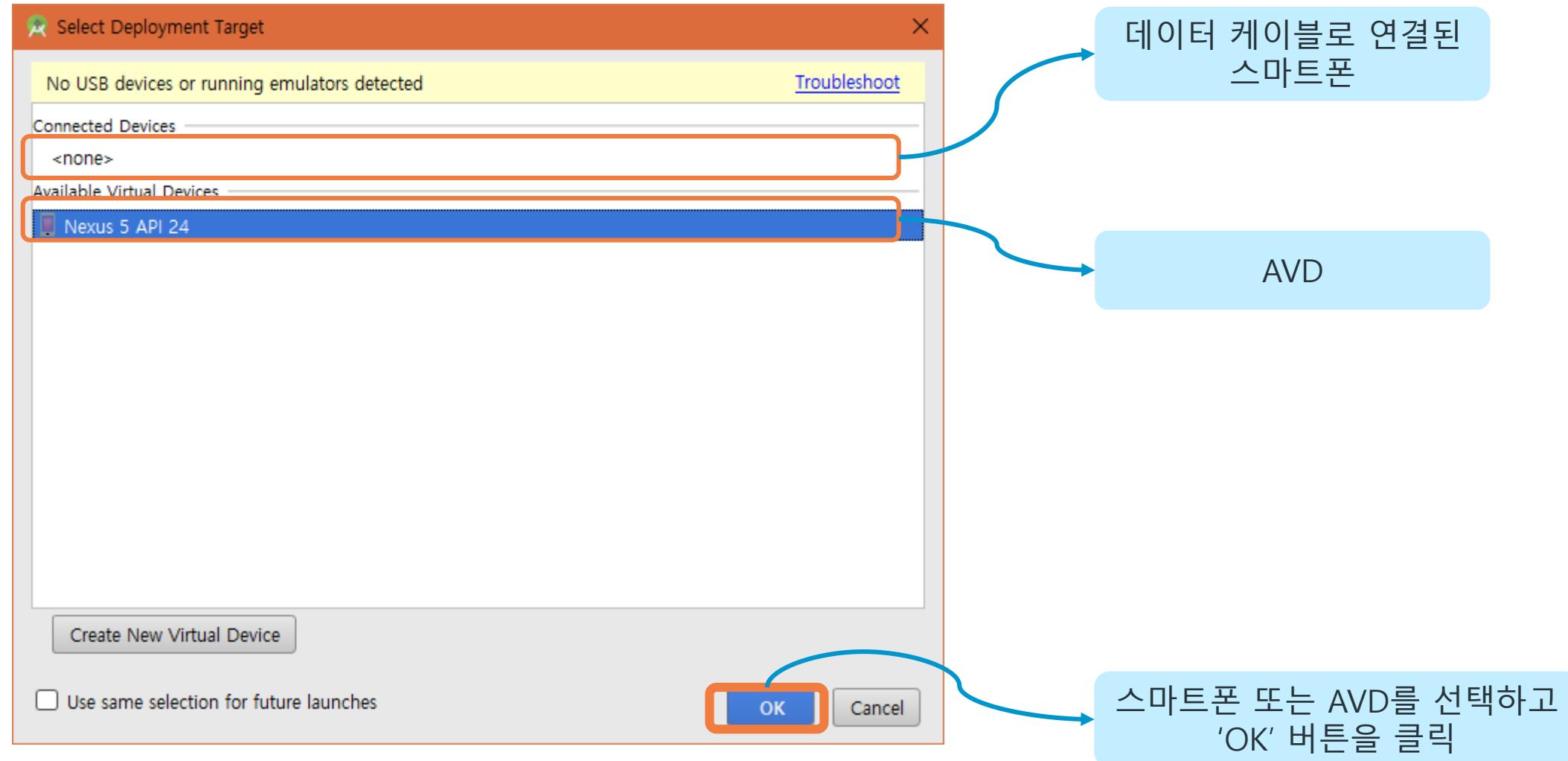
Step 3. 프로젝트 실행

25



• AVD 장비 선택하기

26



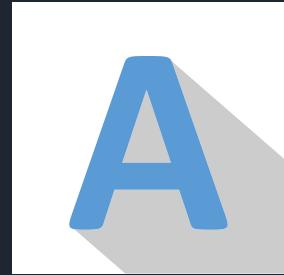
O utputs





question

&



answer

