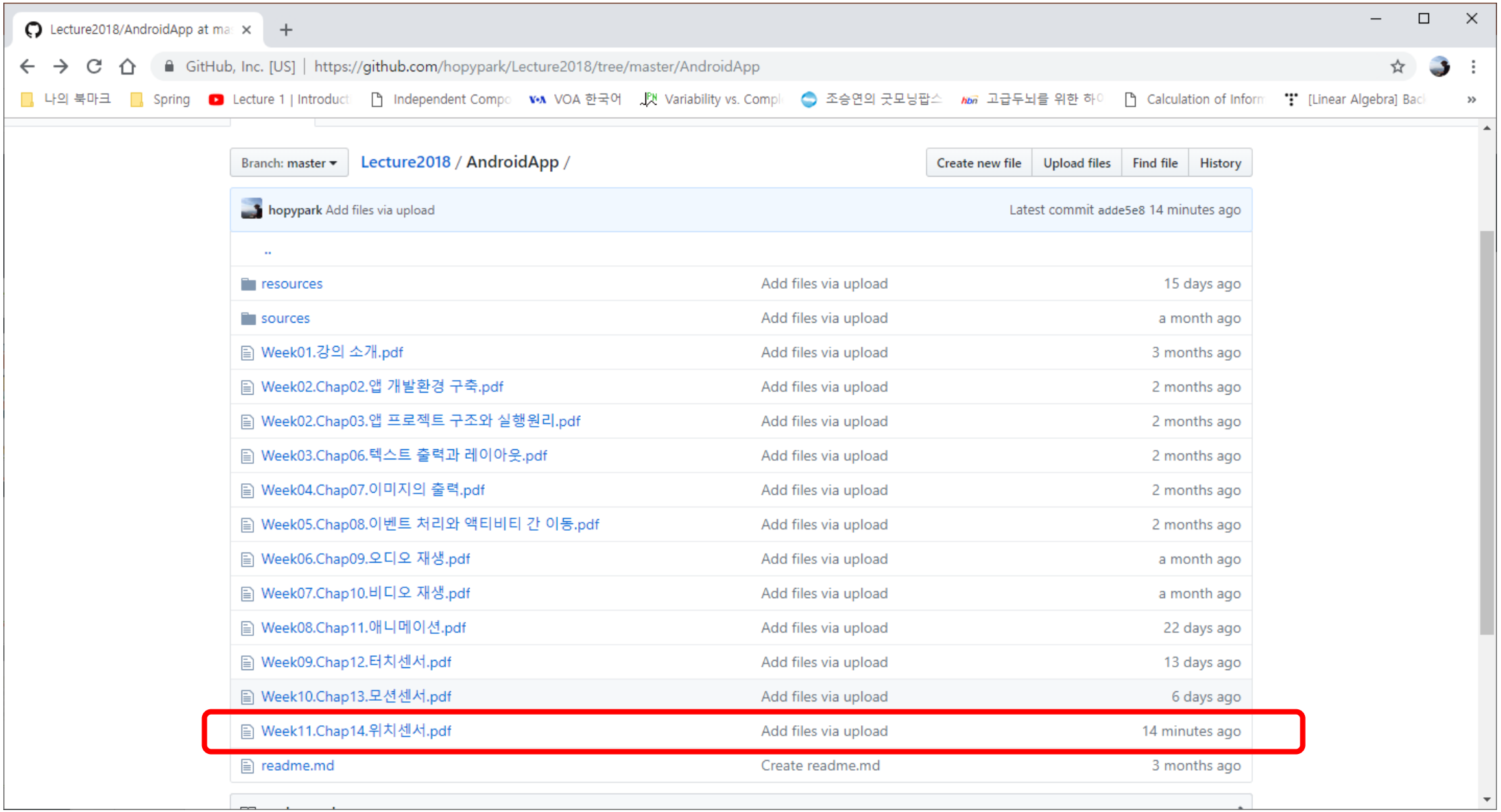
Week11.
위치센서

개발환경 구축 절차

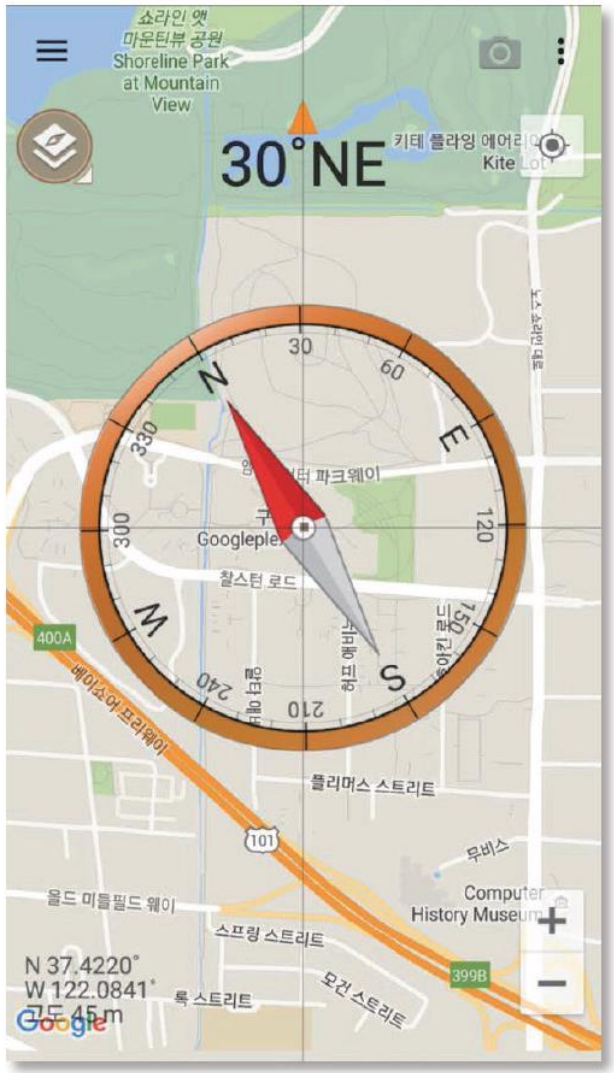
주 차	수 업 내 용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	이미지의 출력
5	이벤트 처리와 액티비티 간 이동
6	오디오 재생
7	비디오 재생
8	**중간고사**
9	애니메이션
10	**사물인터넷과 센서** – 터치 센서, 모션 센서
11	사물인터넷과 센서 – **위치 센서**, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활요
14	구글 맵과 위치 추적
15	**기말 고사**

강의 자료-https://github.com/hopypark

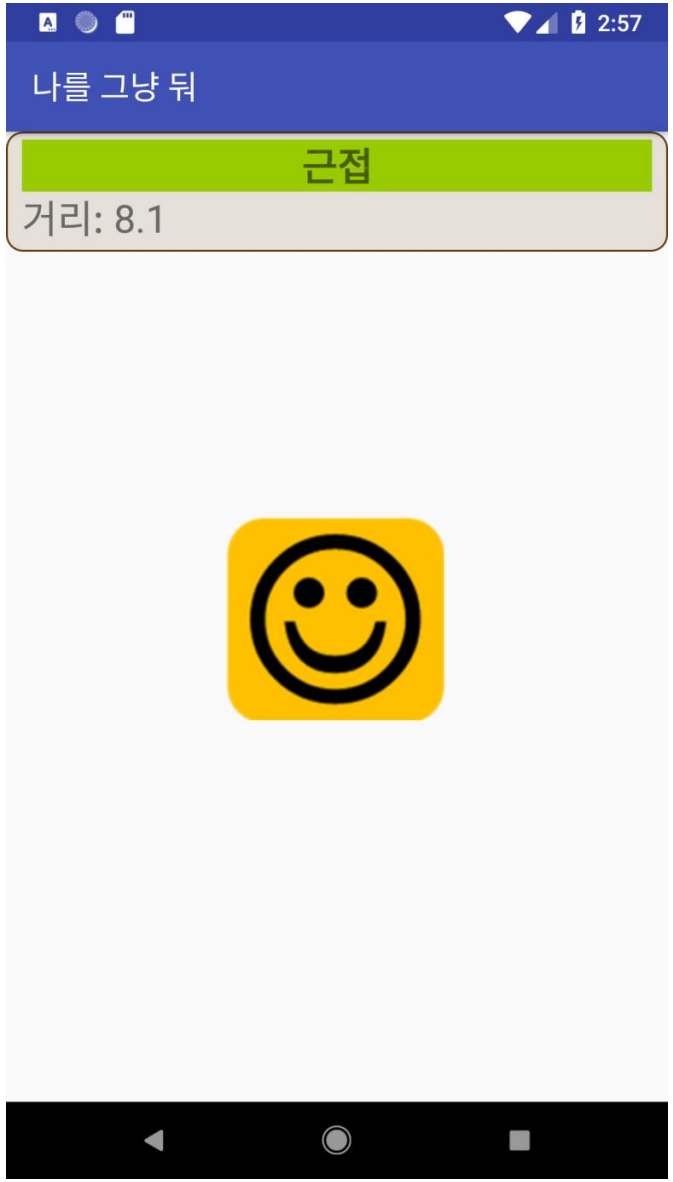


위치 센서를 이용한 앱의 예

(a) 카메라 화면과 방향



(b) 구글맵 상의 위치와 방향

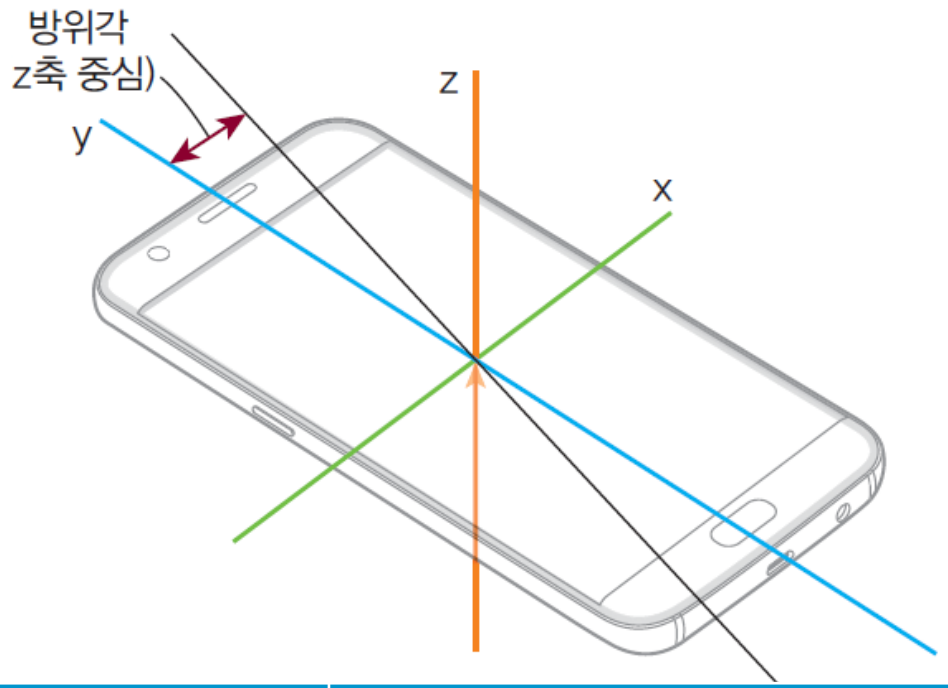


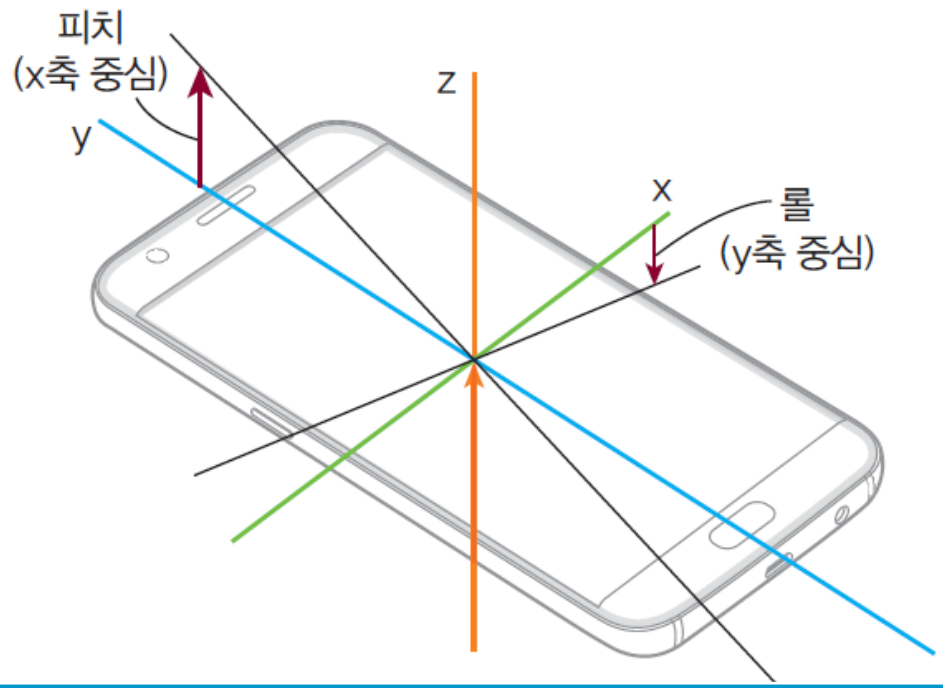
- Smart Compass 앱

모션 센서 원리


레이아웃
activity_main.xml
레이아웃
TextView
액티비티 자바 클래스
MainActivity.java
액티비티 생성
레이아웃 출력
텍스트뷰 인식
위치센서 등록
모션센서 유형 파악
모션센서 값 읽기
모션센서 값 출력
스마트폰
모션 변경
2:57
나를 그냥 둬
근접
거리: 8.1

모션 센서는 센서 값을 표현하기 위해 세가지 축 사용





속 성	설 명
롤Roll(왼쪽-오른쪽)	평평한 상태는 0도이고, 왼쪽으로 기울면 90도까지 증가한다. 반대로 오른쪽으로 기울이면 -90도까지 감소한다.
피치pitch(위-아래)	평평한 상태는 0도로, 핸드폰의 윗부분을 땅 쪽으로 기울이면 90도까지 증가하고, 더 기울여 뒤집어지면 180도까지 증가한다. 반대로 핸드폰의 아랫부분을 땅 쪽으로 기울이면 -90도까지 감소하며, 더 기울여 뒤집어지면 -180도까지 감소한다.
아지무스Azimuth(방위각)	폰의 위가 북쪽을 가리키면 0도, 동쪽을 가리키면 90도, 남쪽을 가리키면 180도, 서쪽을 가리키면 270도이다.

Step 0.프로젝트 개요

프로젝트 개요

스마트폰 센서값 읽기

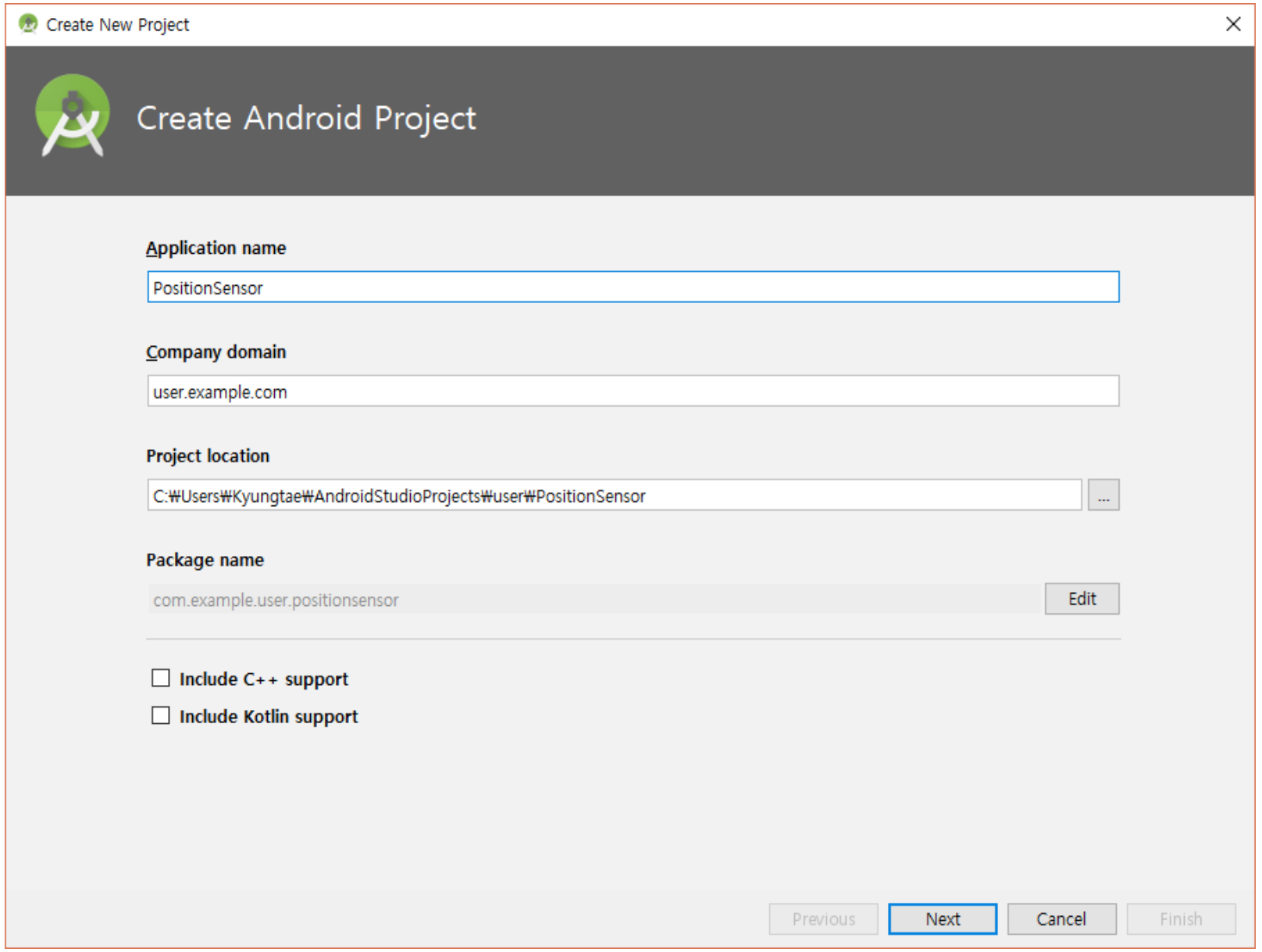
Application Name

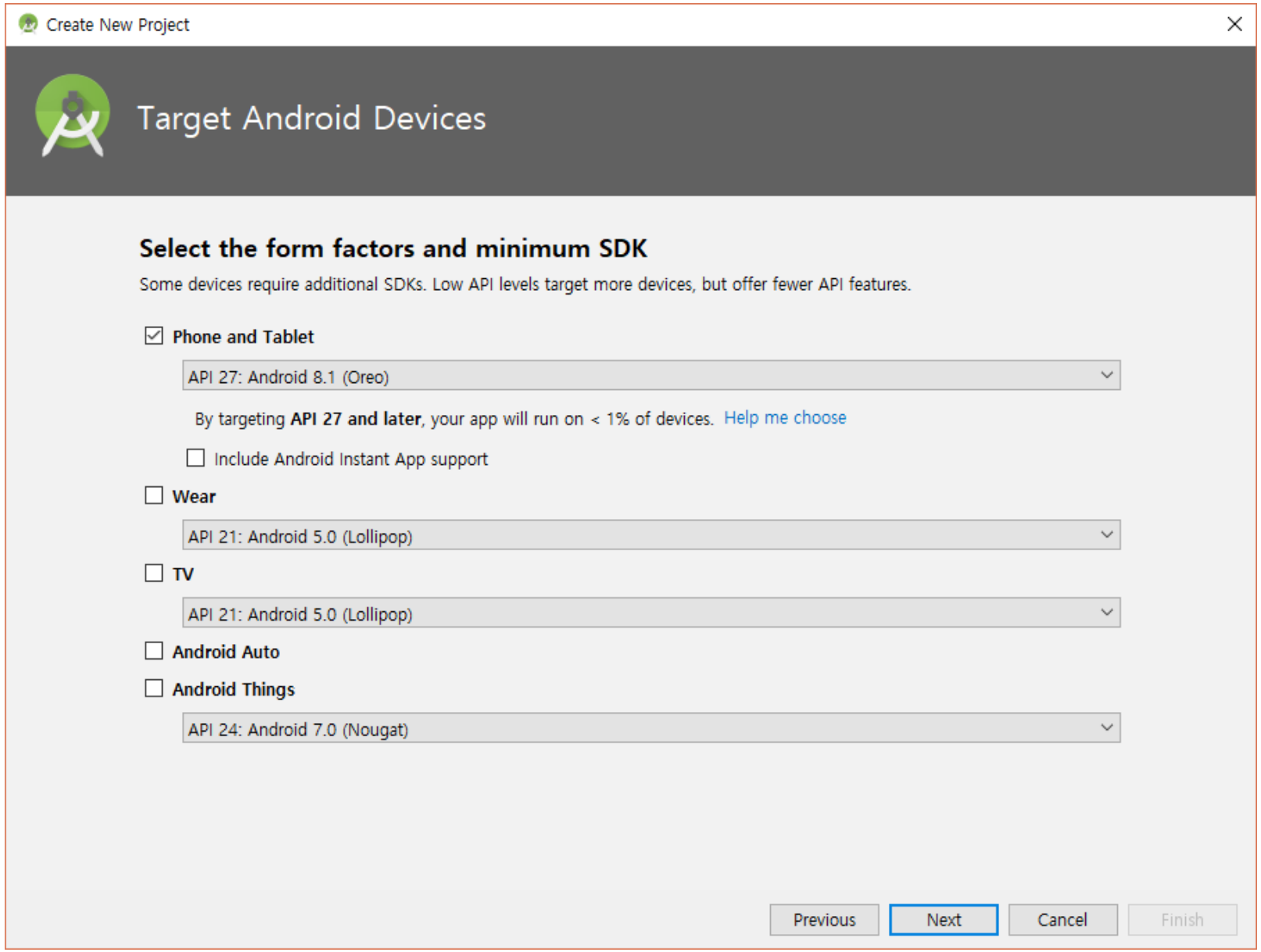
PositionSensor

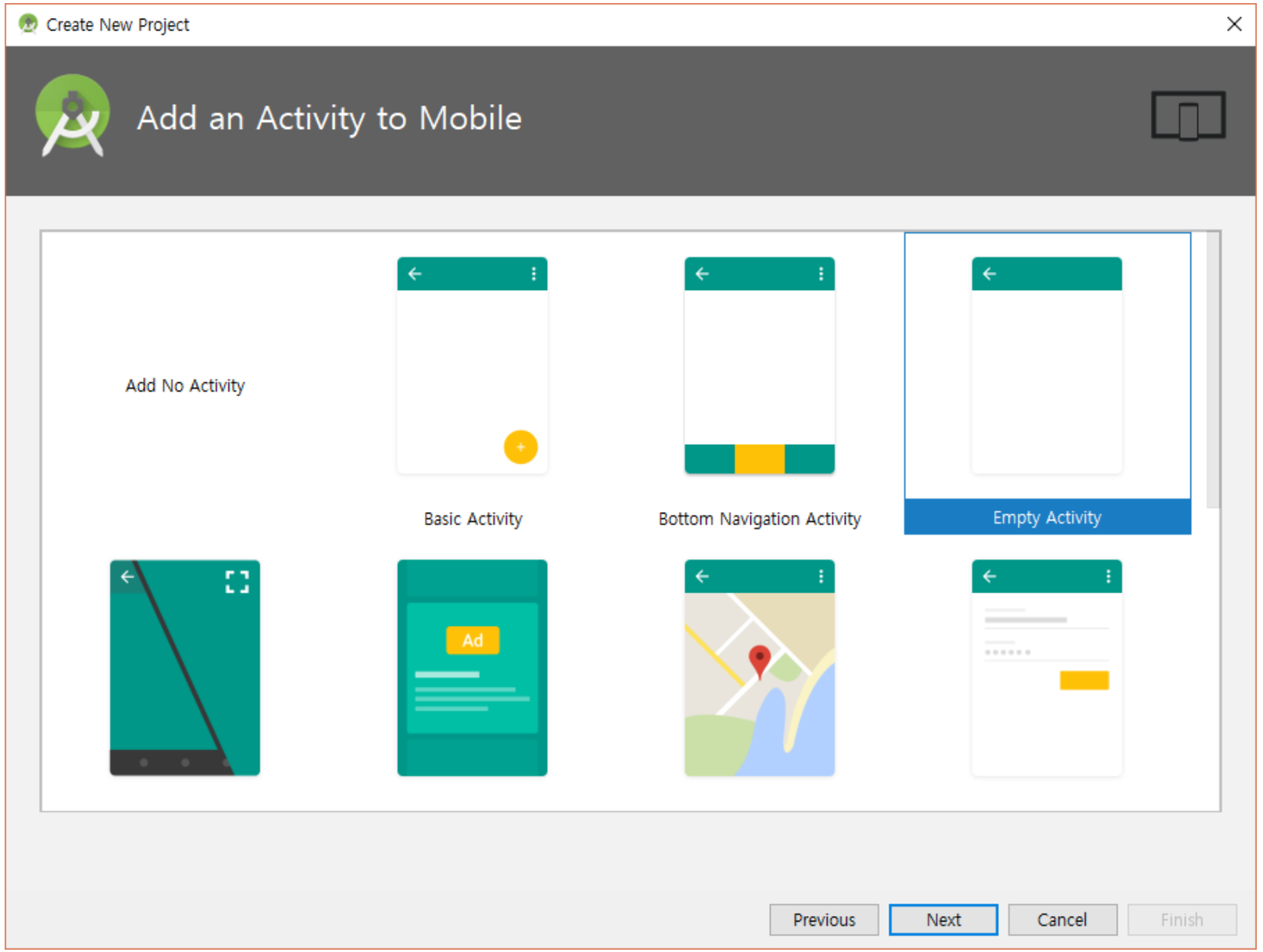
어플리케이션 라벨

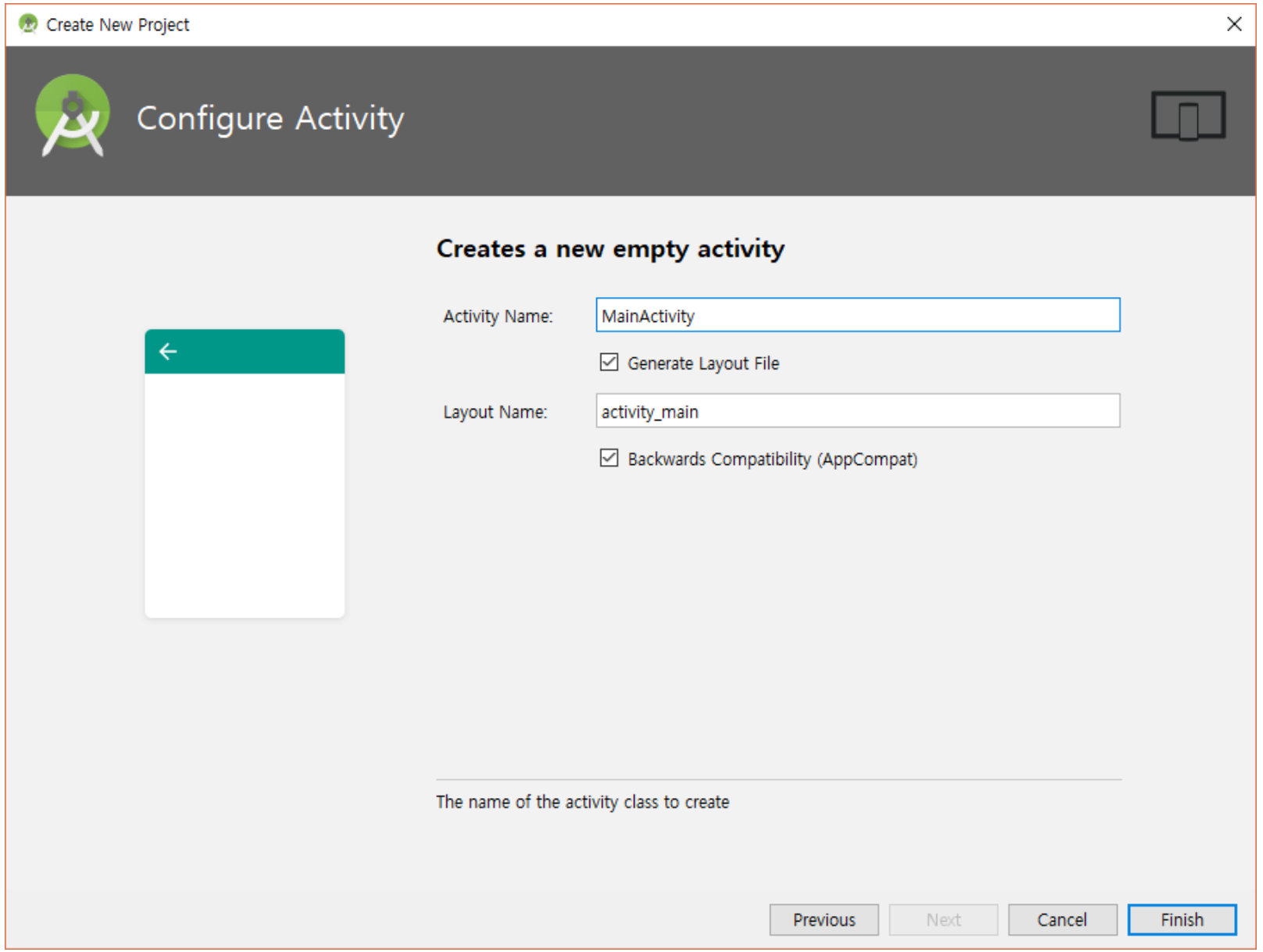
나를 그냥 둬

Create Project – PositionSensor




Create New Project
Target Android Devices
Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.
Phone and Tablet
API 27: Android 8.1 (Oreo)
By targeting API 27 and later, your app will run on < 1% of devices. Help me choose
Include Android Instant App support
Wear
API 21: Android 5.0 (Lollipop)
TV
API 21: Android 5.0 (Lollipop)
Android Auto
Android Things
API 24: Android 7.0 (Nougat)
Previous
Next
Cancel
Finish


Create New Project
Add an Activity to Mobile
Add No Activity
Basic Activity
Bottom Navigation Activity
Empty Activity
Ad
Previous
Next
Cancel
Finish


Create New Project
Configure Activity
Creates a new empty activity
Activity Name:
MainActivity
Generate Layout File
Layout Name:
activity_main
Backwards Compatibility (AppCompat)
The name of the activity class to create
Previous
Next
Cancel
Finish

Step 1. 프로젝트 생성

절차	내 용
①프로젝트 시작	메뉴에서 'File → New Project' 클릭
②프로젝트 구성	Application Name: **PositionSensor**
	Company Domain: kyungtae.example.com(디폴트 사용)
	Project Location: ~/AndroidStudioProject/ktpark/PositionSensor
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

모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.video1	MainActivity.java	• 센서 등록 • 센서 종류 확인 및 값 변경 확인 • 근접이면 이미지 변경 및 진동
res	anim	shaking.xml	• 아이콘 이미지의 진동 애니메이션
	drawable	shape_list	• 출력모양 설계(배경색)
		smile.png, angry.png	• 아이콘 이미지
	layout	activity_main.xml	• 이미지의 화면 중앙 배치
	mipmap	ic_launcher.png	
	values	colors.xml	
		dimens.xml	
		strings.xml	• 어플리케이션 라벨 수정 • "나를 그냥 둬"의 문자열 추가
		styles.xml	

이미지 리소스

smile.png

angry.png

(drawable)

화면 레이아웃

```
RelativeLayout
  ImageView
    id @+id/img
    src @drawable/smile
    ...
```

activity_main.xml (layout)

액티비티 제어

```
onCreate()
  super onCreate()
  setContentView(R.layout.activity_main)
  img = findViewById(R.id.img)
  ...
onSensorChanged()
  ...
   ani = AnimationUtil.loadAnimation
  (this, R.anim.shaking)
  img.setImageResource(R.drawable.angry);
  img.startAnimation(ani);
  mVibe. ... vibrate(1000);
  ...
```

MainActivity.java (layout)

목록 아이템의 모양

```
set
  translate
```

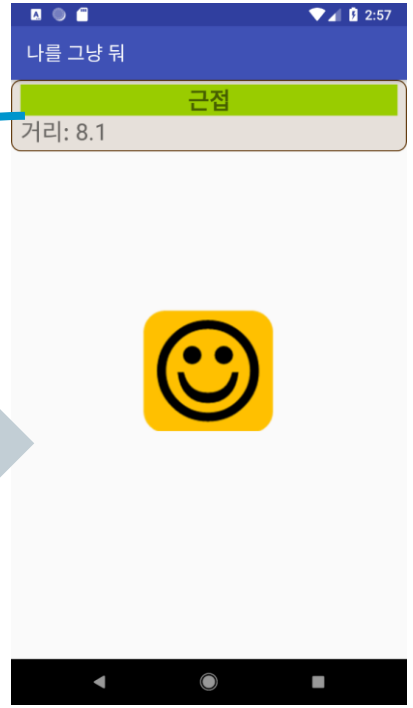
shaking.xml (drawable)

좌우로 진동

컴파일/빌더

애니메이션 설정

이미지에
애니메이션 설정



어플리케이션 구성
액티비티의 자바 클래스

텍스트 자원

```
string
  app_name 나를 버려 둬
```

strings.xml (values)

앱 라벨

어플리케이션 기본 정보

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

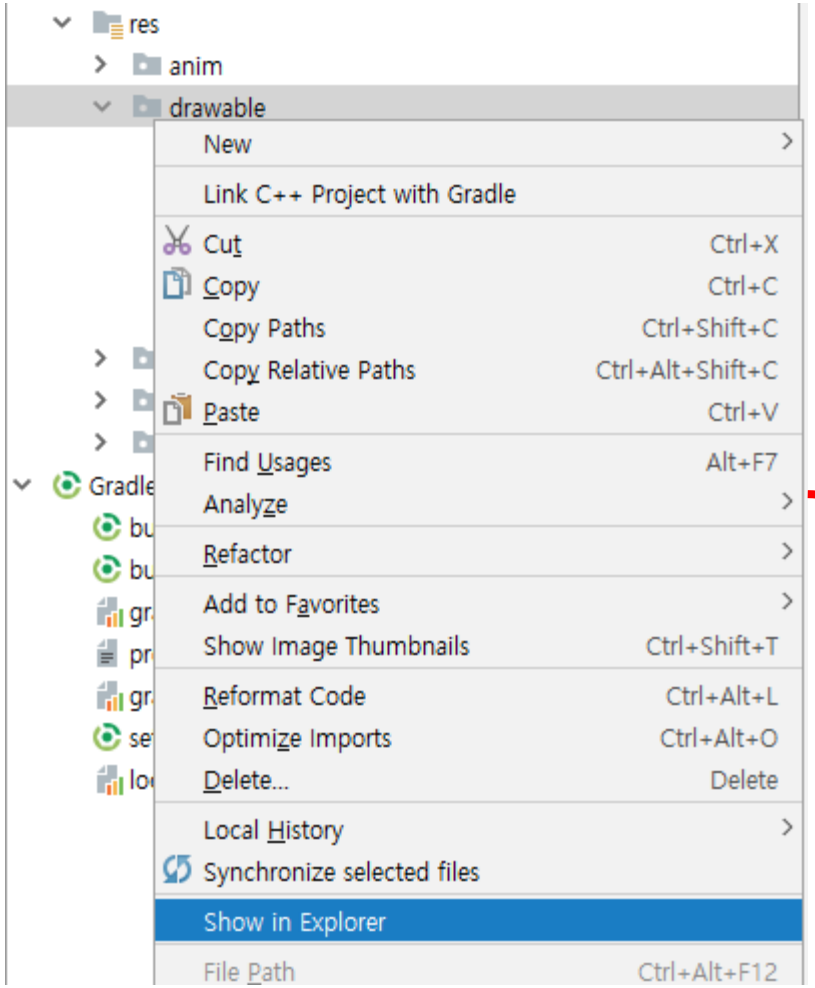
AndroidManifest.xml (manifest)

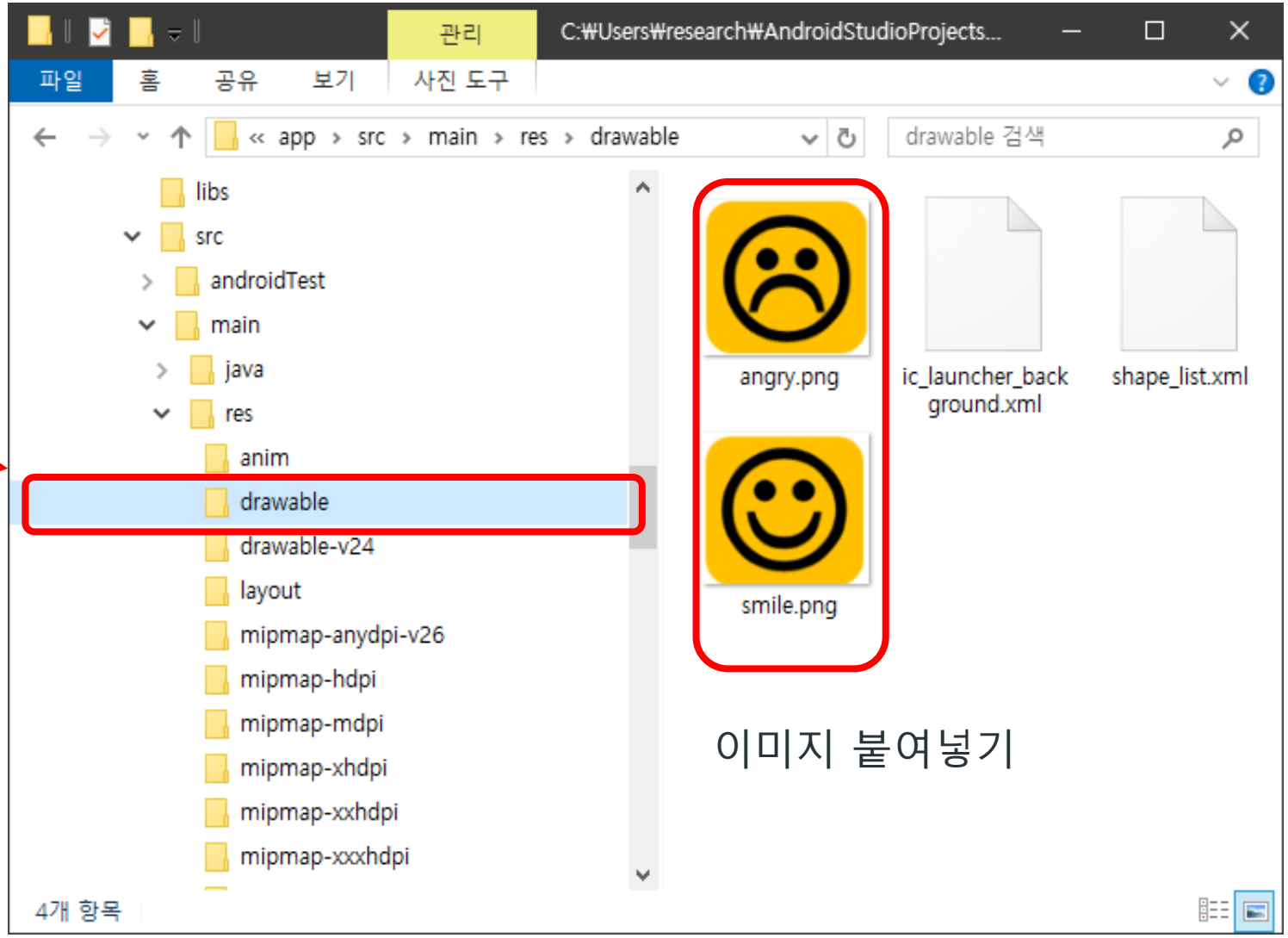
컴파일/빌더 정보

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

(Gradle Scripts)

이미지 리소스 올리기





이미지 붙여넣기

Step 2.1 텍스트 자원의 편집

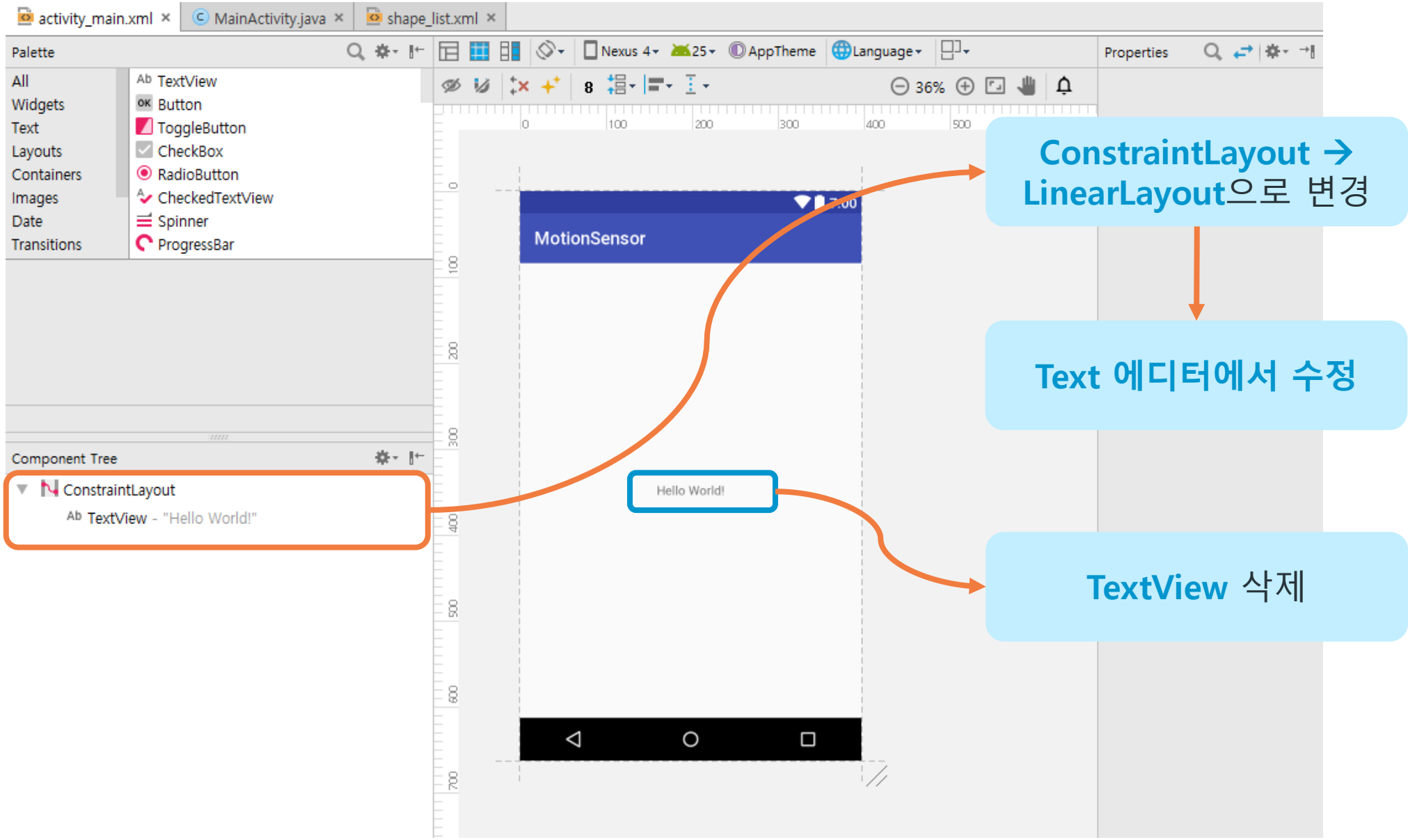
- strings.xml

activity_main.xml | shape_list.xml | MainActivity.java | strings.xml

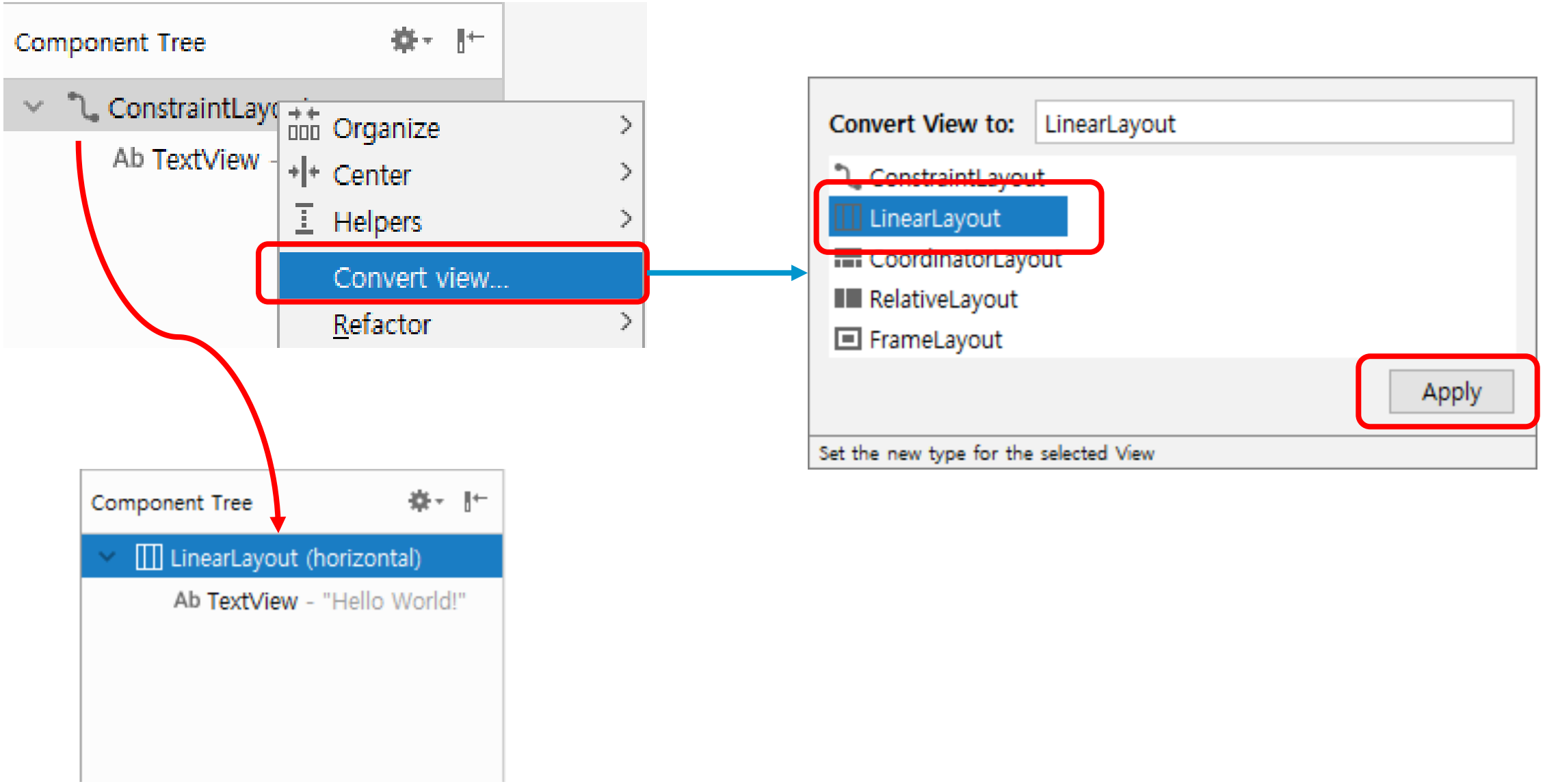
Edit translations for all locales in the translations editor.

```
1  <resources>
2      <string name="app_name">나를 그냥 둬</string>
3
4      <string name="sensor_proximity">근접</string>
5  </resources>
6
7
```

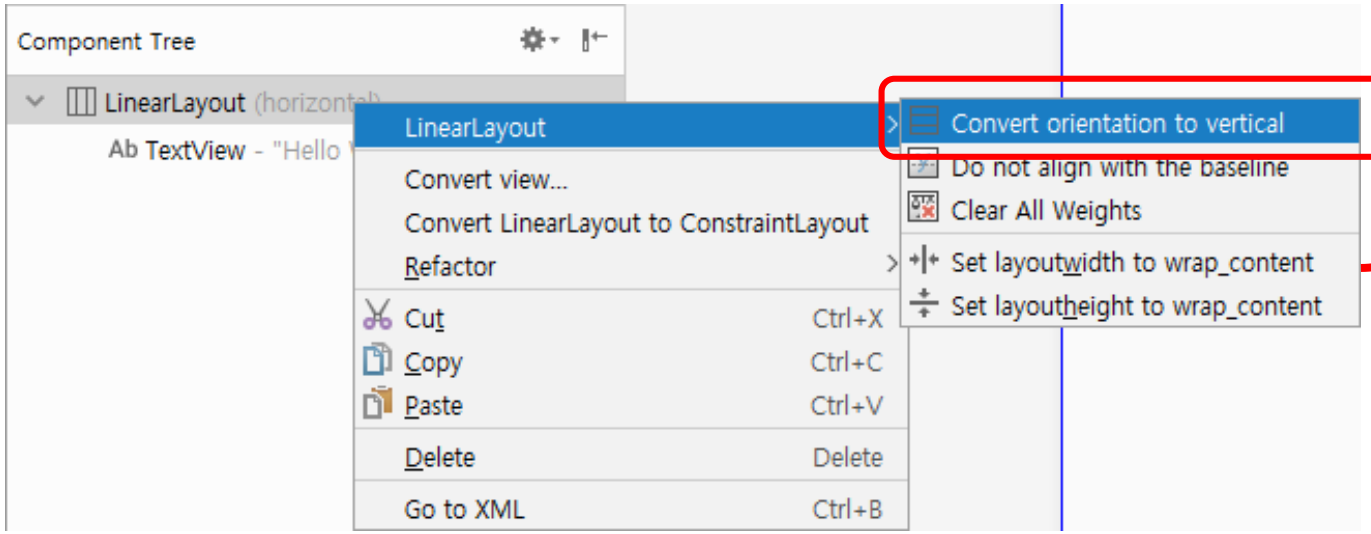
2.2 화면 설계

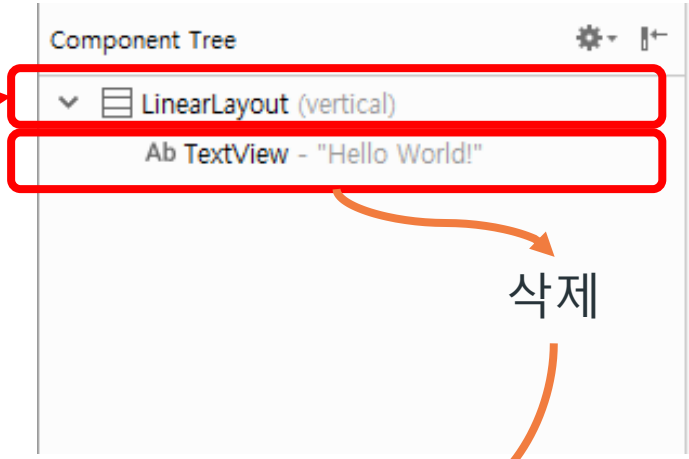

activity_main.xml
MainActivity.java
shape_list.xml
Palette
All
Widgets
Text
Layouts
Containers
Images
Date
Transitions
TextView
Button
ToggleButton
CheckBox
RadioButton
CheckedTextView
Spinner
ProgressBar
Component Tree
ConstraintLayout
TextView - "Hello World!"
Nexus 4
25
AppTheme
Language
36%
Properties
MotionSensor
Hello World!
ConstraintLayout → LinearLayout으로 변경
Text 에디터에서 수정
TextView 삭제

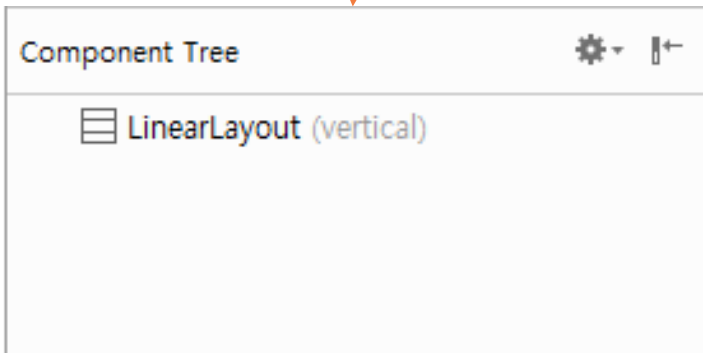
ConstraintLayout을 LinearLayout로 바꾸기



LinearLayout의 방향을 Horizontal → Vertical로 변경하기

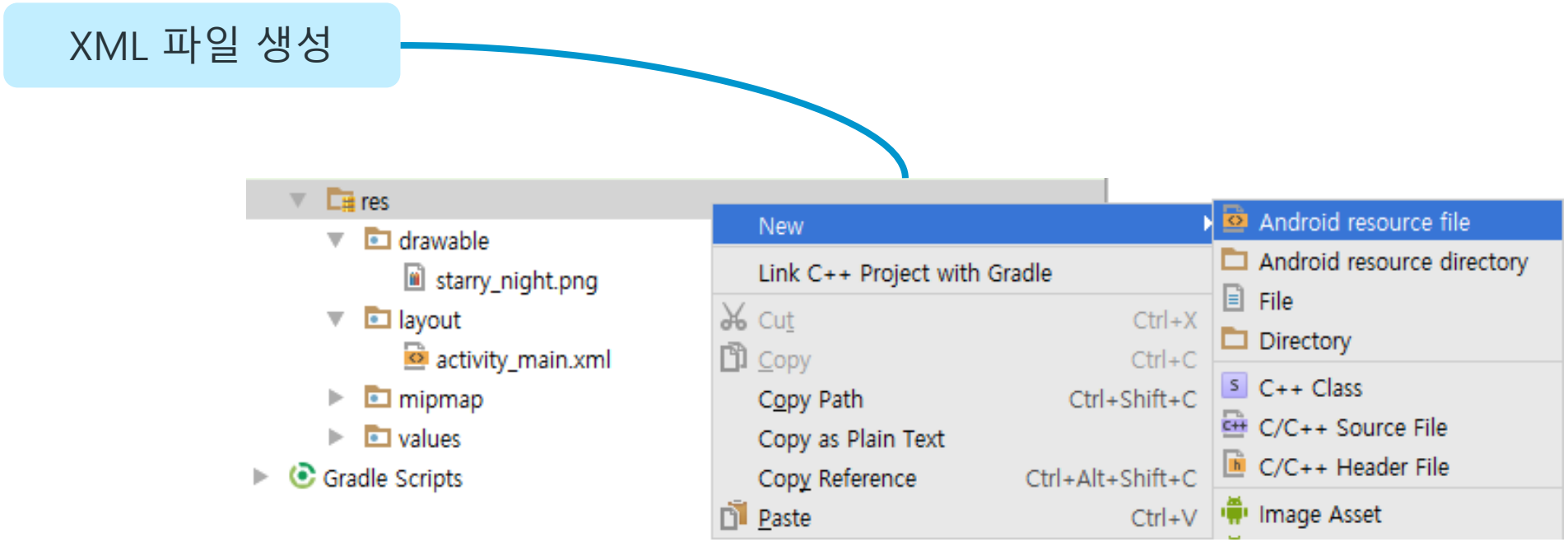






2.3 drawable 리소스 – shape_list.xml추가

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



• Set New Resource File - shape_list.xml

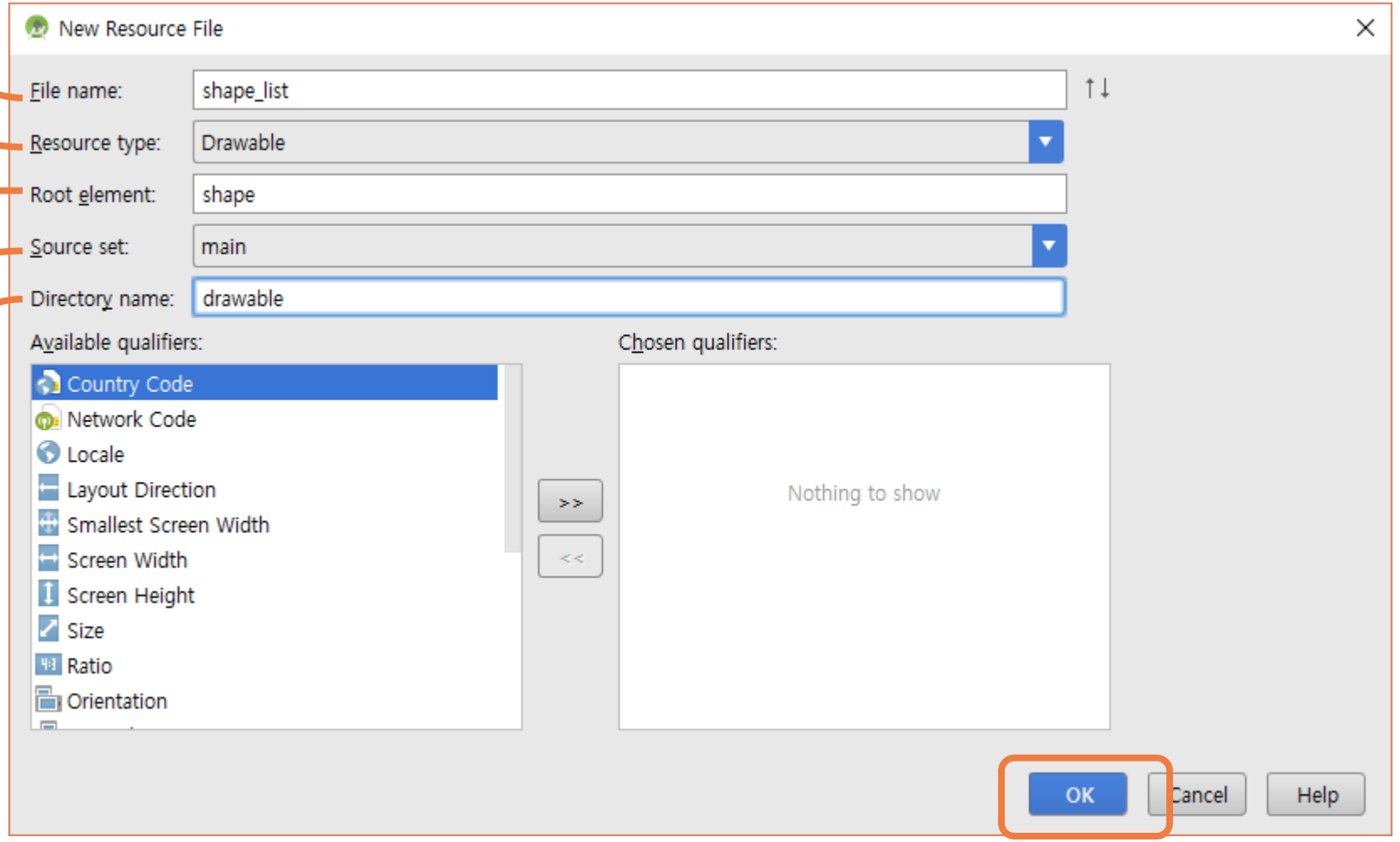
File name: shape_list

Resource type: Drawable

Root element: shape

Source set: main

Directory name: drawable



• shape_list.xml

activity_main.xml × | MainActivity.java × | shape_list.xml ×

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android"
3      android:shape="rectangle">
4
5      <solid android:color="#3061380B"/>
6
7      <stroke android:width="1dp" android:color="#61380B"/>
8
9      <padding
10         android:top="2dp"
11         android:bottom="2dp"
12         android:left="10dp"
13         android:right="10dp">
14     </padding>
15
16     <corners android:radius="5dp"></corners>
17
18 </shape>
19
20
```

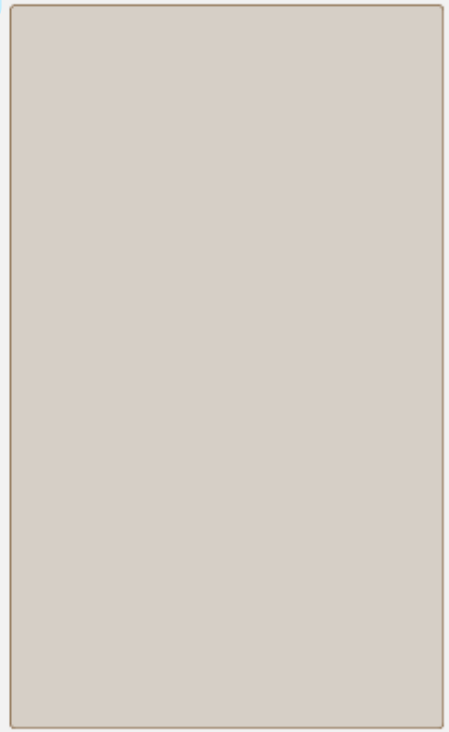
출력모양을 내부의 색

출력모양을 테두리의 색

내부 패딩 정보

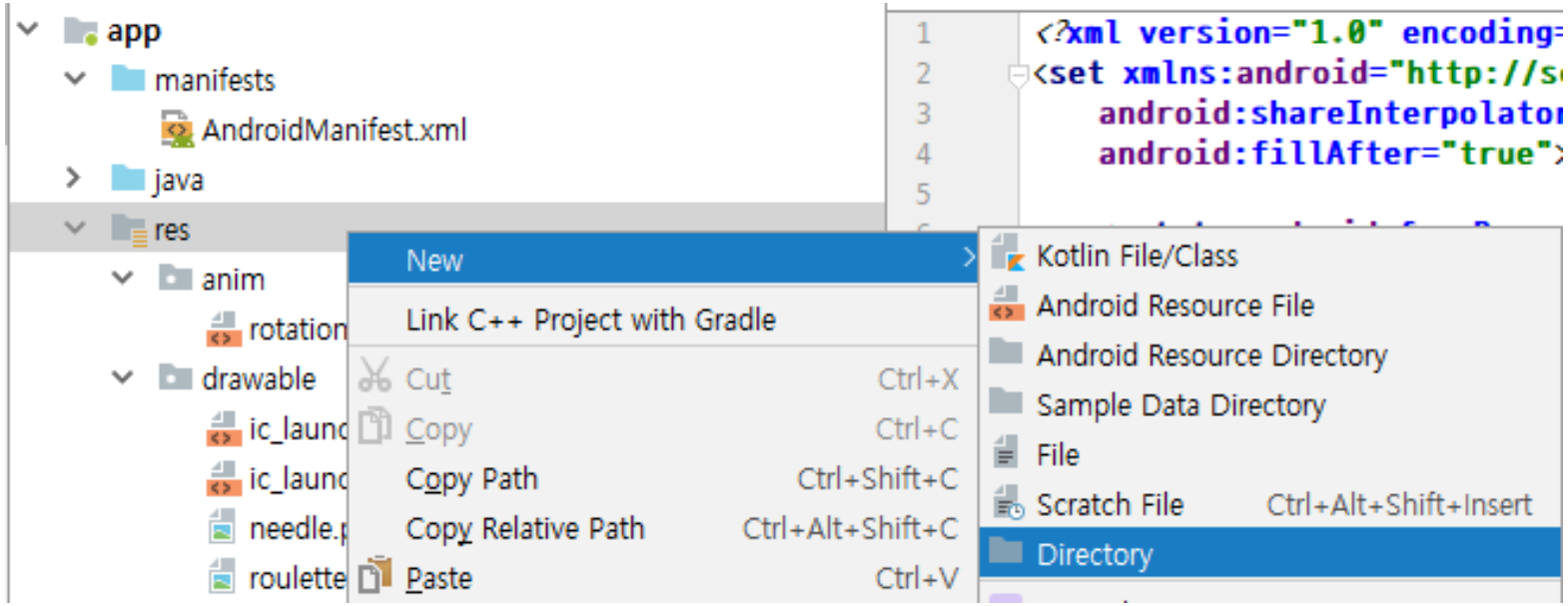
출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp)

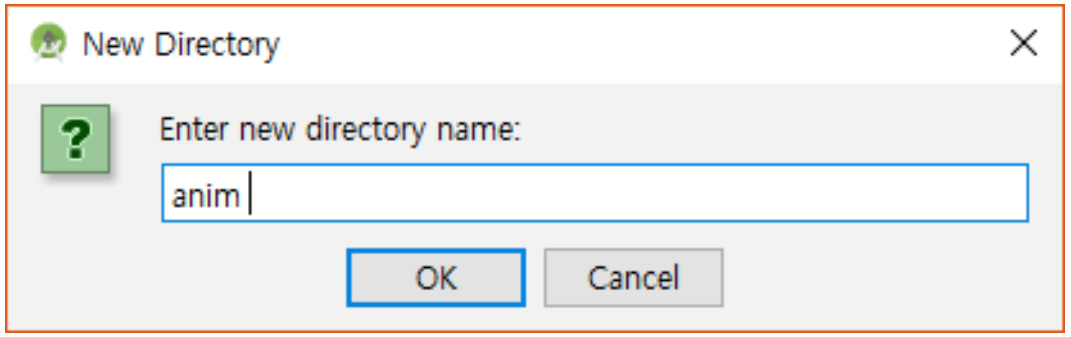
Preview



Animation 객체 추가 – res/anim 폴더 만들기

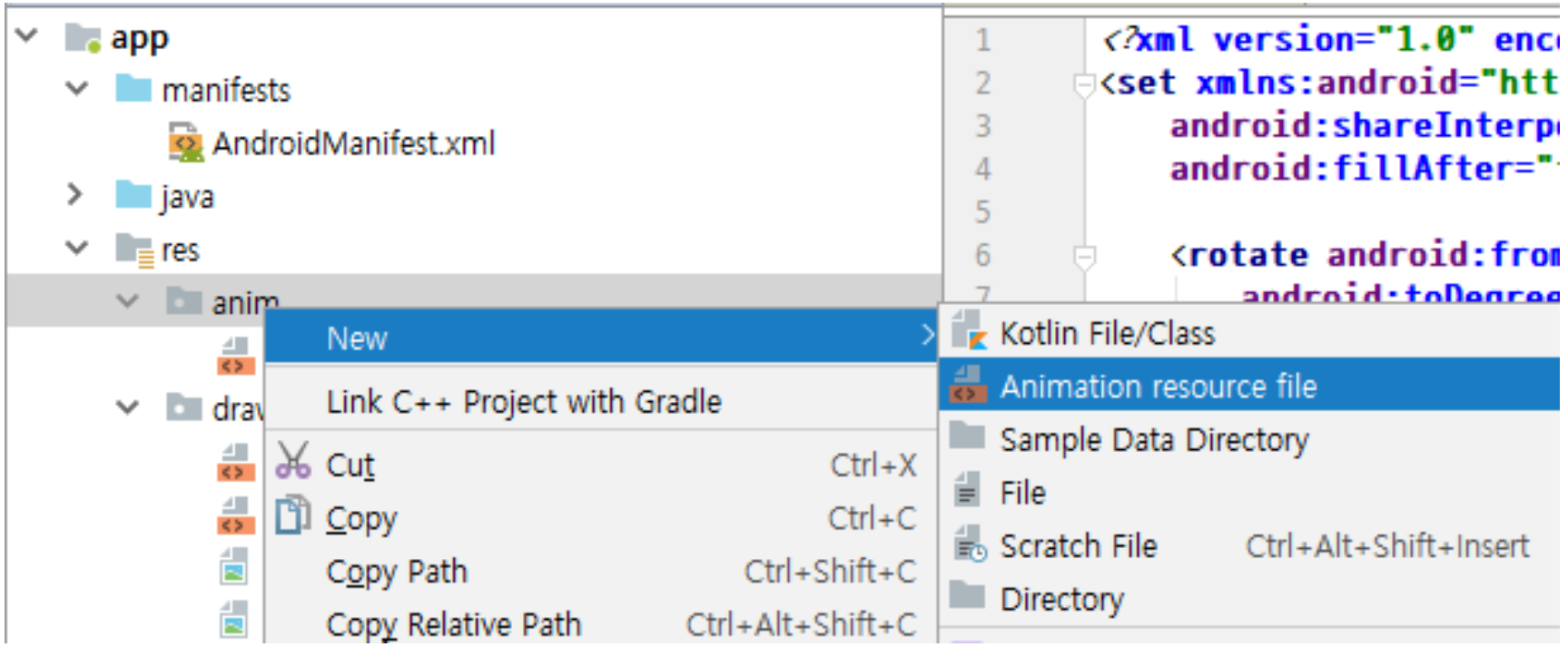
- 에니메이션 설정을 위한 xml 파일 생성



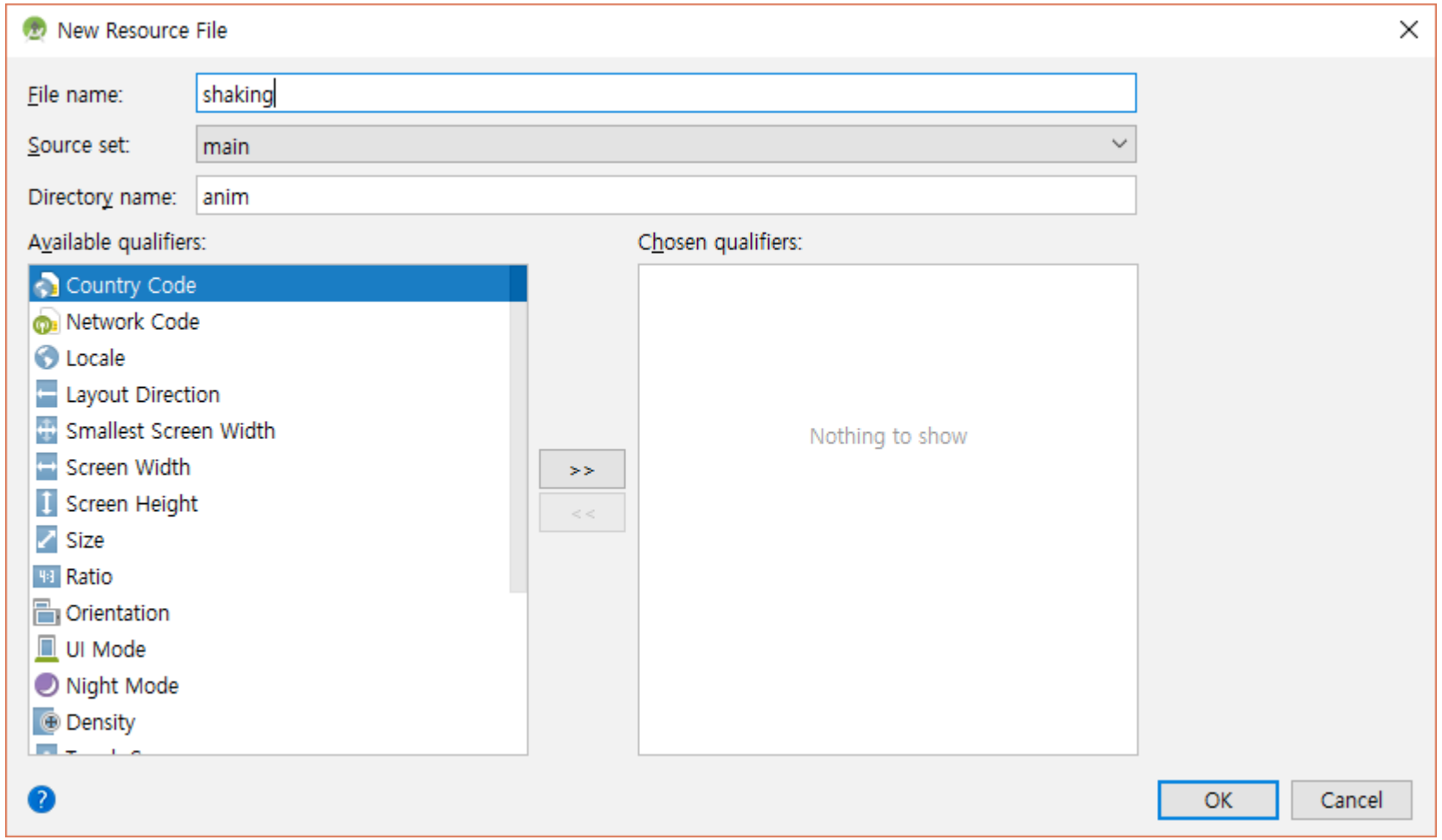


Animation 객체 추가 – rotation.xml 파일 만들기

- 에니메이션 설정을 위한 xml 파일 생성



shaking.xml 파일 만들기

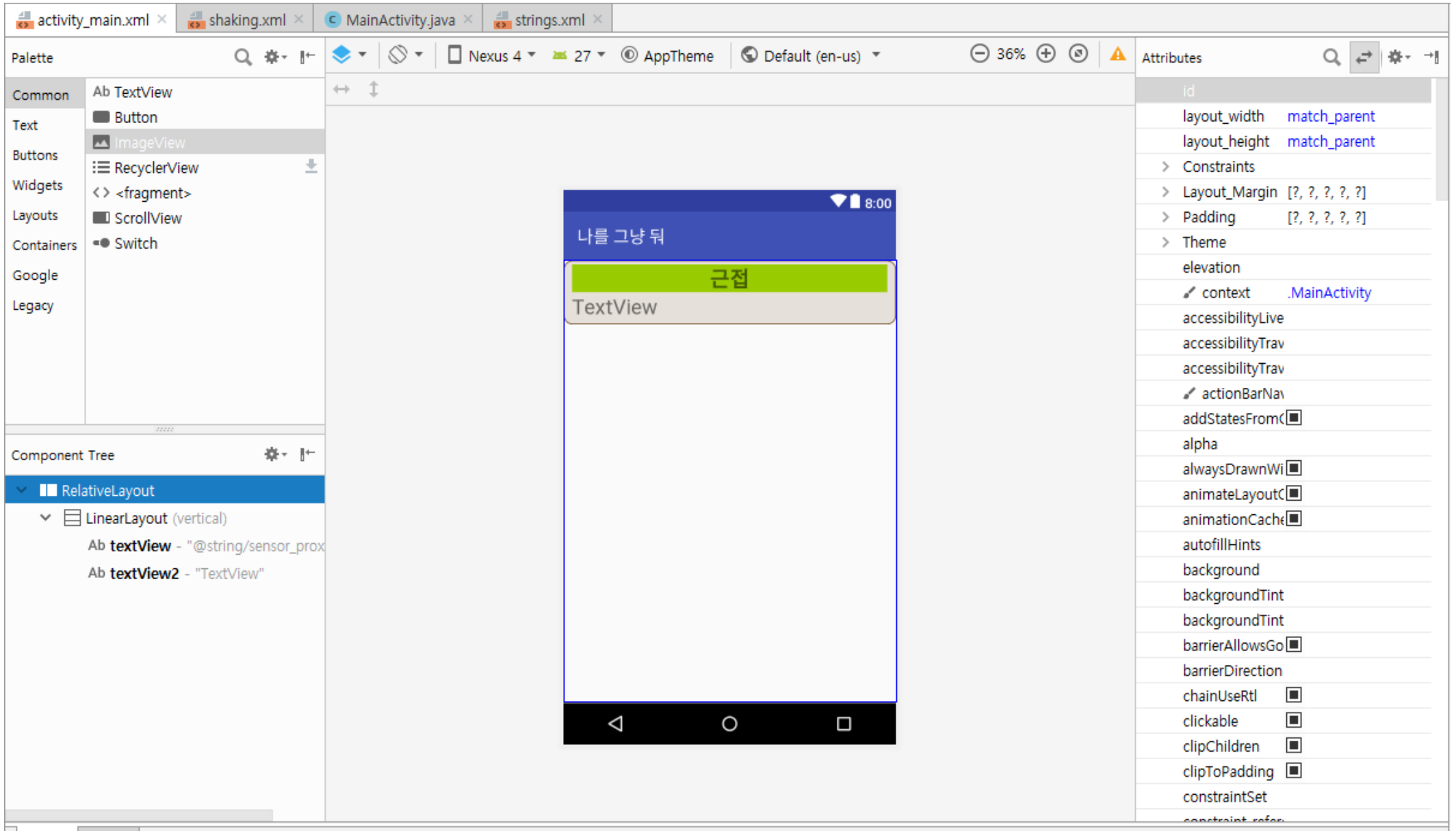


shaking.xml 파일

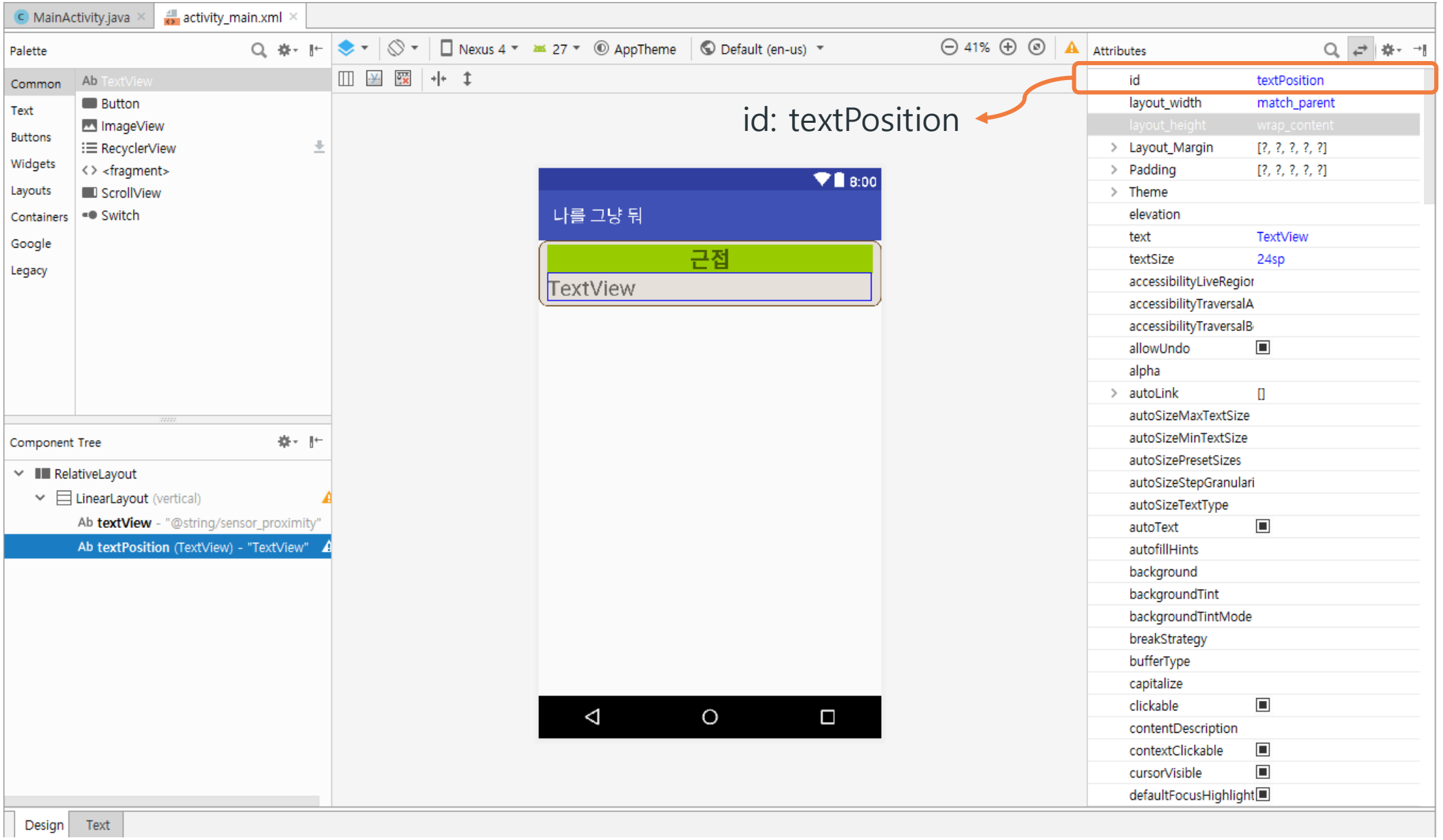
activity_main.xml | shaking.xml | MainActivity.java | strings.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- 이동위치 변화:
         x방향으로 이미지너비의 2% 크기만큼을 100msec 초 동안 10회 반복 -->
    <translate
        android:fromXDelta="-2%"
        android:toXDelta="2%"
        android:duration="100"
        android:repeatCount="10">
    </translate>

</set>

```

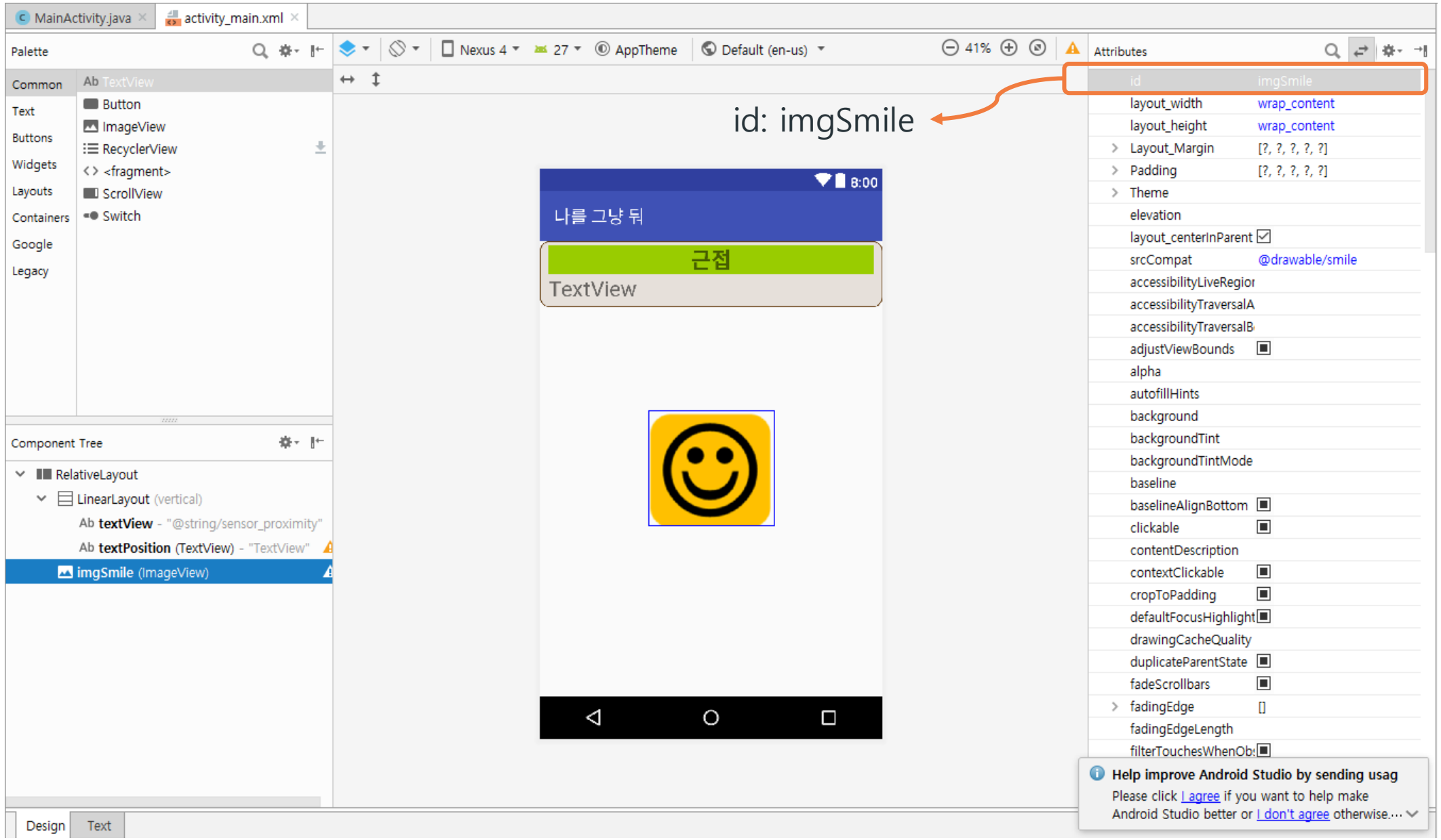
근접 센서 정보 – TextView



근접 센서 정보 – TextView



애니메이션용 이미지 – ImageView



2.5 Activity 제어(MainActivity.java)

- 센서이벤트 처리를 위한 액티비티 인터페이스 추가

activity_main.xml | shaking.xml | MainActivity.java

```
1  package com.example.user.positionsensor;
2
3  import android.support.v7.app.AppCompatActivity;
4  import android.os.Bundle;
5
6  public class MainActivity extends AppCompatActivity {
7
8      @Override
9      protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

센서값 변화에 따른 이벤트 처리를 위한 클래스

activity_main.xml | shaking.xml | MainActivity.java

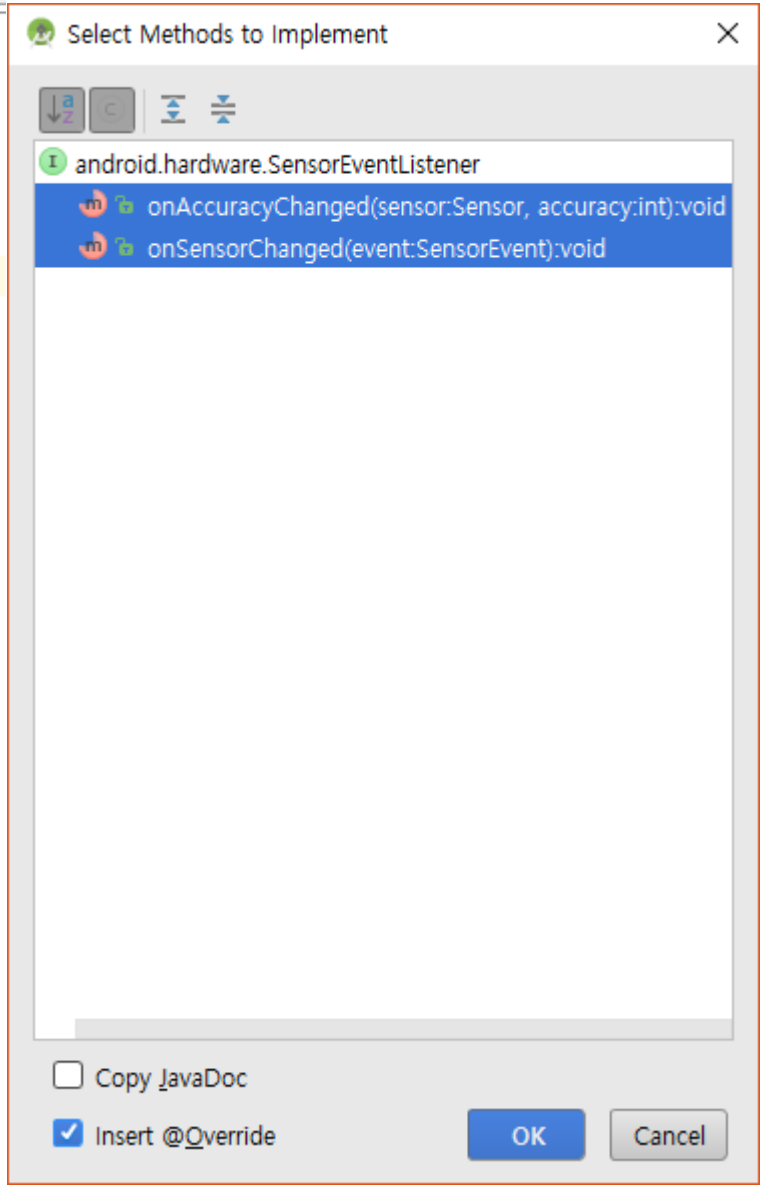
```
1  package com.example.user.positionsensor;
2
3  import android.hardware.SensorEventListener;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6
7  public class MainActivity extends AppCompatActivity implements SensorEventListener {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13     }
14 }
15
```

• 센서값 처리를 위한 매소드 구현(@override)

activity_main.xml | shaking.xml | MainActivity.java

```
1  package com.example.user.positionsensor;
2
3  import android.hardware.SensorEventListener;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6
7  public class MainActivity extends AppCompatActivity implements SensorEventListener {
8
9
10                                                    tanceState) {
11
12                                                    in);
13
14
15
16
```

- Implement methods
- Make 'MainActivity' abstract
- Create Test
- Create subclass
- Unimplement Interface
- Annotate interface 'SensorEventListener' as @Deprecated



activity_main.xml | shaking.xml | MainActivity.java

```java
package com.example.user.positionsensor;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity implements SensorEventListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {

    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }
}
```

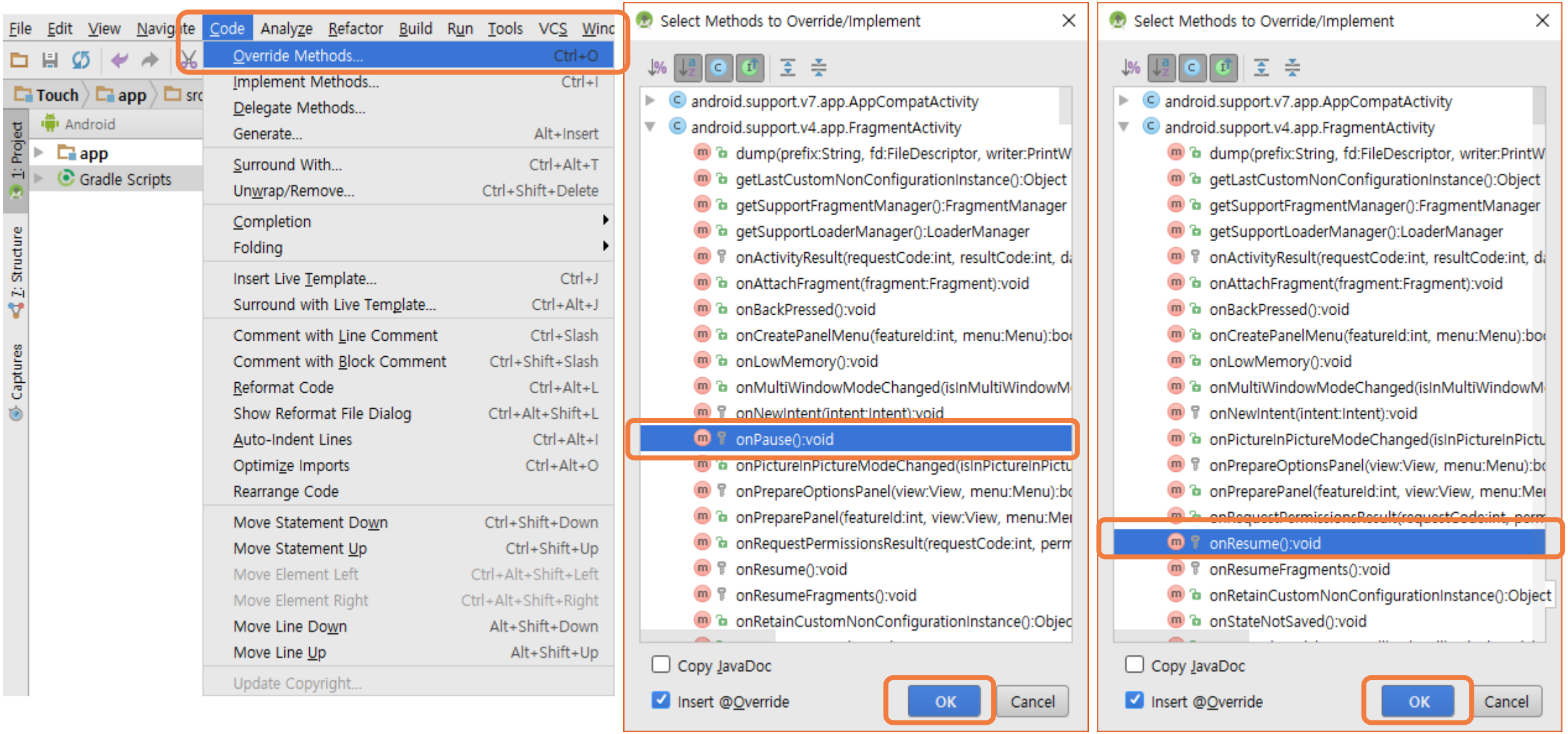
센서 값이 변할 때 호출

등록된 센서의 정확도가 변할 대 호출

- 센서와 진동을 처리하기 위한 변수 선언 **(빨간상자만 입력)**

```
17  public class MainActivity extends AppCompatActivity implements SensorEventListener{
18
19      ImageView img;
20      TextView textView;
21
22      SensorManager sm;
23      Sensor sensor_proximity;
24
25      Vibrator mVibe;
26
27      @Override
28      protected void onCreate(Bundle savedInstanceState) {
29          super.onCreate(savedInstanceState);
30          setContentView(R.layout.activity_main);
31
32          img = (ImageView) findViewById(R.id.imgSmile);
33          textView = (TextView) findViewById(R.id.textPosition);
34
35          sm = (SensorManager) getSystemService(SENSOR_SERVICE);
36          sensor_proximity = sm.getDefaultSensor(Sensor.TYPE_PROXIMITY);
37
38          mVibe = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
39      }
40
```

onPause()/onResume() 매소드 재정의(Override)



- 재정의를 위한 매소드 추가

```
40
41      @Override
42      protected void onPause() {
43          super.onPause();
44      }
45
46      @Override
47      protected void onResume() {
48          super.onResume();
49      }
```

화면에 표시되는 상태에서 사용자와 상호작용하지 않을 때

액티비티가 일시정지(pause)상태에서 복귀할 때 호출

(빨간상자만 입력)

```
45      @Override
46      protected void onPause() {
47          super.onPause();
48          sm.unregisterListener(this);
49      }
50
51      @Override
52      protected void onResume() {
53          super.onResume();
54
55          sm.registerListener( listener: this, sensor_proximity, SensorManager.SENSOR_DELAY_NORMAL);
56      }
```

- 근접 센서의 값 읽어서 처리하기**(빨간상자만 입력)**

```
41      @Override
42      public void onSensorChanged(SensorEvent event) {
43
44          if (event.sensor.getType() == Sensor.TYPE_PROXIMITY){
45
46              textView.setText("거리: " + event.values[0]);
47
48              if(event.values[0] < 5){ // min=0, max=10
49                  Animation ani = AnimationUtils.loadAnimation( context: this, R.anim.shaking);
50
51                  img.setImageResource(R.drawable.angry);
52                  img.startAnimation(ani);
53
54                  // 1000 : Vibrate for 1 sec
55                  // VibrationEffect.DEFAULT_AMPLITUDE - would perform vibration at full strength
56                  VibrationEffect effect = VibrationEffect.createOneShot( milliseconds: 1000, VibrationEffect.DEFAULT_AMPLITUDE);
57                  mVibe.vibrate(effect);
58              }else{
59                  img.setImageResource(R.drawable.smile);
60              }
61          }
62
63      }
64
```

빨간줄 처리하기
(진동 사용에 대한 허가가 필요)

- 진동 사용 허가 주기-AndoridManifest.xml **(빨간상자만 입력)**

activity_main.xml | shaking.xml | MainActivity.java | AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.user.positionsensor">
4
5      <uses-permission android:name="android.permission.VIBRATE" />
6
7      <application
8          android:allowBackup="true"
9          android:icon="@mipmap/ic_launcher"
10         android:label="나를 그냥 둬"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportsRtl="true"
13         android:theme="@style/AppTheme">
14         <activity android:name=".MainActivity">
15             <intent-filter>
16                 <action android:name="android.intent.action.MAIN" />
17
18                 <category android:name="android.intent.category.LAUNCHER" />
19             </intent-filter>
20         </activity>
21     </application>
22
23 </manifest>
24
```

Activity LifeCycle

메소드	설명	다음 메소드
onCreate()	액티비티가 생성될 때 호출되며 사용자 인터페이스 초기화에 사용됨.	onStart()
onRestart()	액티비티가 멈췄다가 다시 시작되기 바로 전에 호출됨.	onStart()
onStart()	액티비티가 사용자에게 보여지기 바로 직전에 호출됨.	onResume() 또는 onStop()
onResume()	액티비티가 사용자와 상호작용하기 바로 전에 호출됨.	onPause()
onPause()	다른 액티비티가 보여질 때 호출됨. 데이터 저장, 스레드 중지 등의 처리를 하기에 적당한 메소드.	onResume() 또는 onStop()
onStop()	액티비티가 더이상 사용자에게 보여지지 않을 때 호출됨. 메모리가 부족할 경우에는 onStop() 메소드가 호출되지 않을 수도 있음.	onRestart() 또는 onDestroy()
onDestroy()	액티비티가 소멸될 때 호출됨. finish() 메소드가 호출되거나 시스템이 메모리 확보를 위해 액티비티를 제거할 때 호출됨.	없음

클래스와 속성/메소드

- 클래스

클래스/인터페이스	설명
Sensor	센서 표현을 위한 클래스
SensorEvent	센서 이벤트를 표현하는 클래스
SensorEventListener	센서 값이 변할 때 센서 매니저로부터 공지를 받는 데 사용
SensorManager	디바이스의 센서에 접근할 수 있도록 함.

- 상수

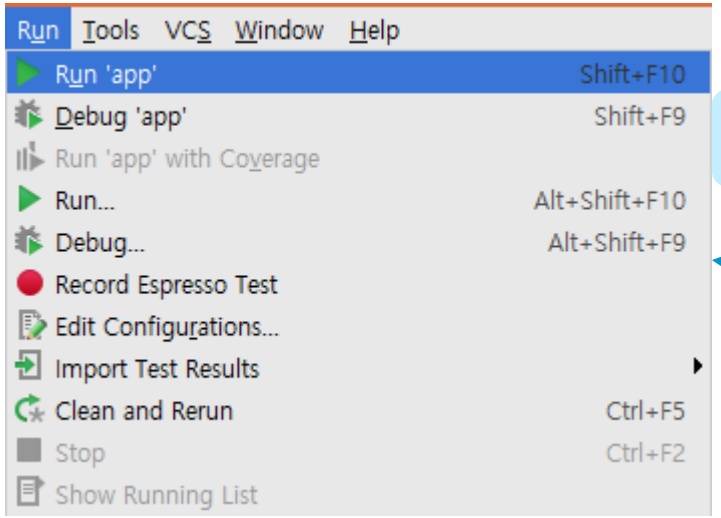
클래스	상수	설명
Context	String SENSOR_SERVICE	센서 이용을 위한 센서 매니저를 추출하기 위해 getSystemService(Class)와 함께 사용
Sensor	int TYPE_GRAVITY	중력센서 타입을 기술하는 상수
	int TYPE_ACCELEROMETER	가속도계 타입을 기술하는 상수
	int TYPE_LINEAR_ACCELERATION	직선 가속도 센서 타입을 기술하는 상수
	int TYPE_GYROSCOPE	자이로스코프 센서 타입을 기술하는 상수
SensorManager	int SENSOR_DELAY_NORMAL	스크린 방향 변화에 적당한 비율

클래스와 속성/메소드

- 메소드

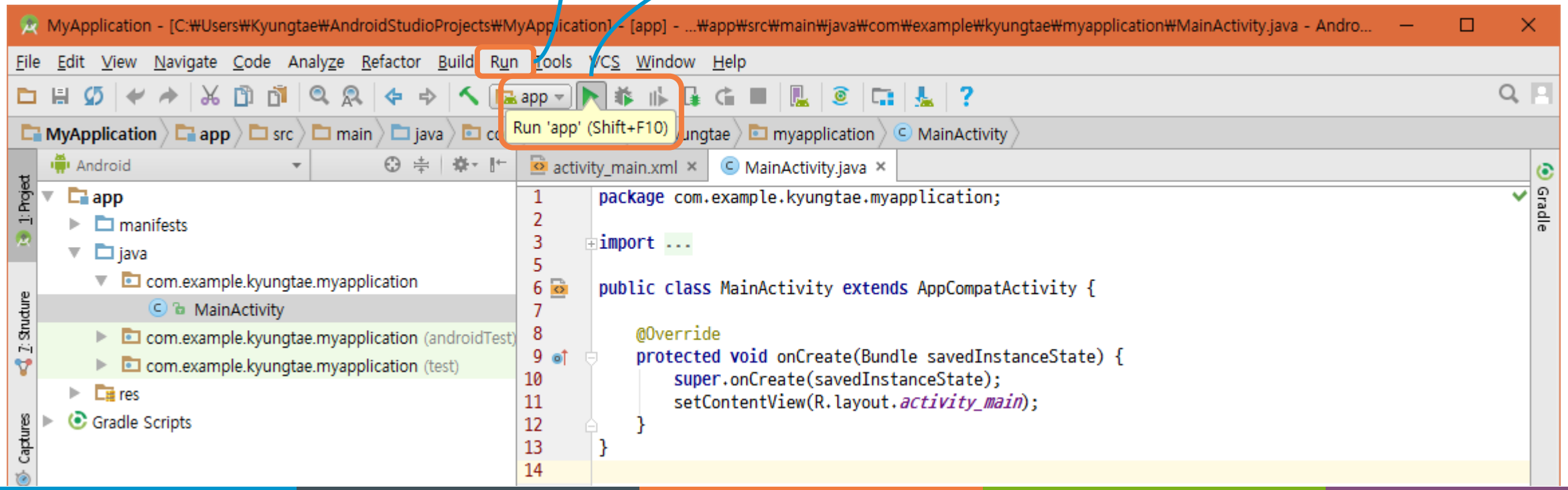
클래스	매소드	설명
Sensor	int **getType**()	센서 타입을 반환함
SensorEvent	public final float[] **values**	센서가 인식한 값들을 저장하는 배열
SensorEventListener	abstract void **onAccuracyChanged**(Sensor, int accuracy)	등록된 센서의 정확도가 변할 때 호출됨
	abstract void **onSensorChanged**(SenseorEvent event)	센서 값이 변할 때 호출됨
SensorManager	Sensor **getDefaultSensor**(int type)	주어진 type을 위한 디폴트 센서를 얻기 위해 사용
	Boolean **registerListener**(SensorEventListener listener, Sensor sensor, int samplingPeriodUs)	주어진 샘플링 주파수에서 주어진 센서를 위한 SensorEventListener를 등록함
	void **unregisterListener**(SensorEventListener listener)	모든 센서에 대한 리스너를 해제함

Step 3. 프로젝트 실행

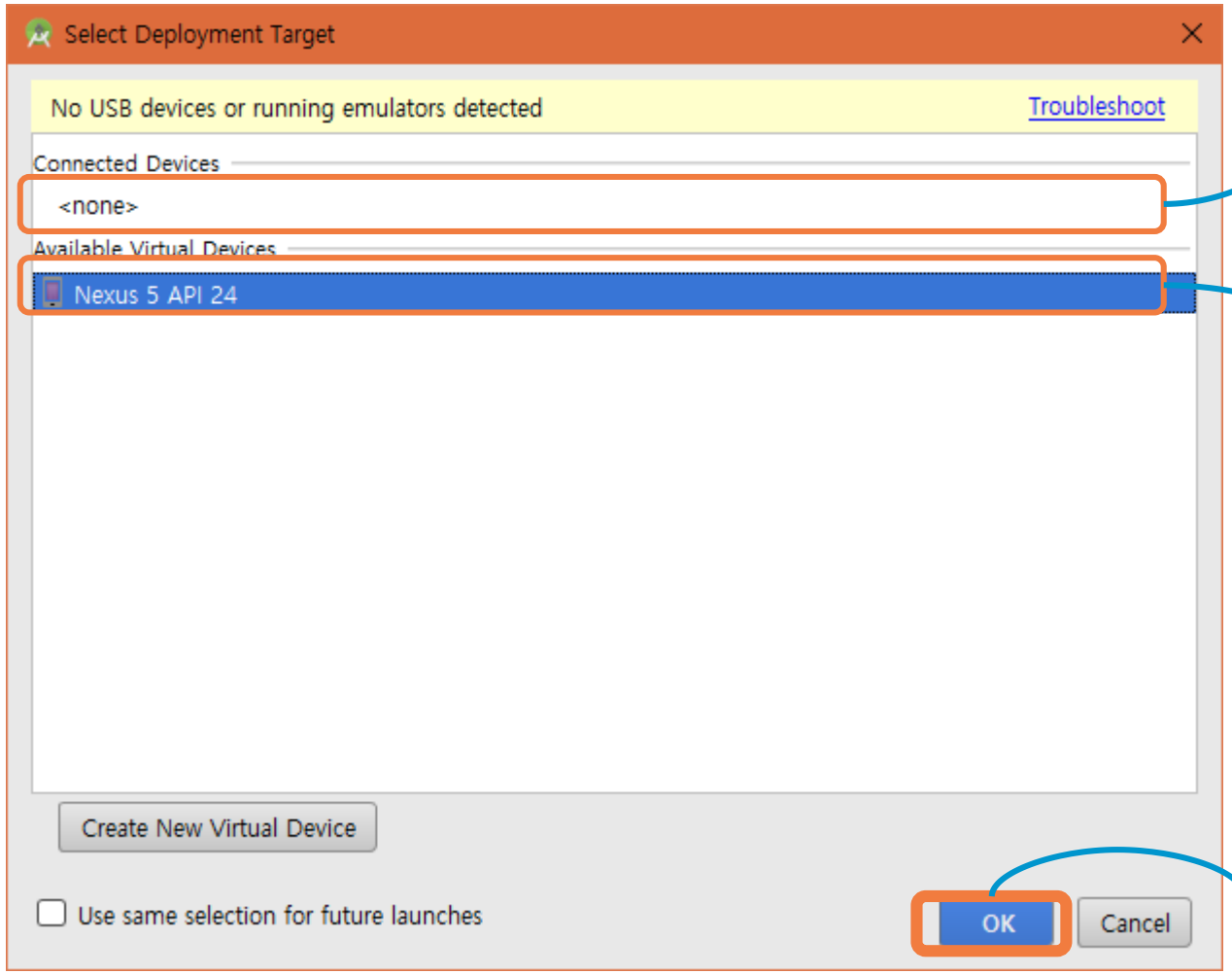


Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기

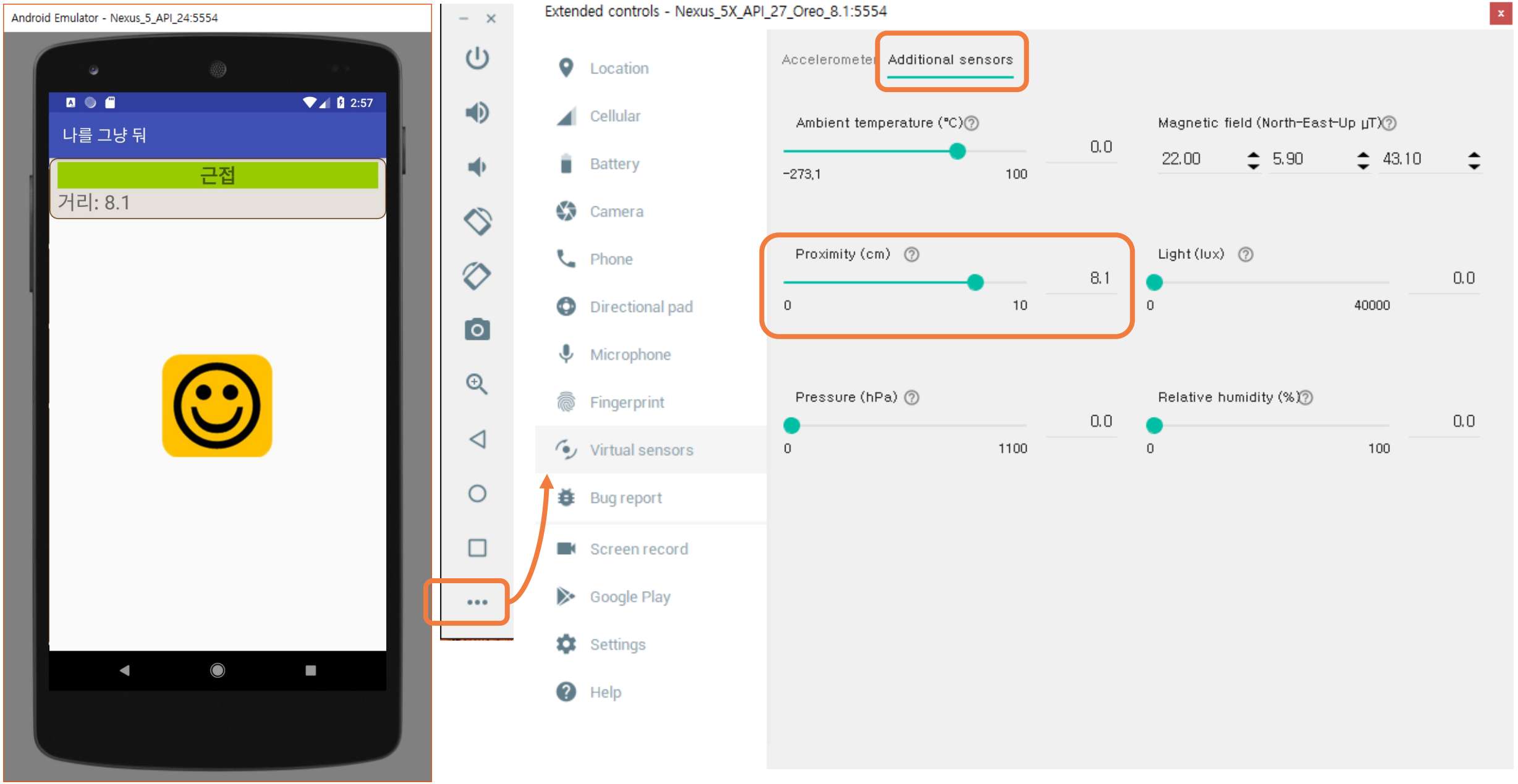


데이터 케이블로 연결된 스마트폰

AVD

스마트폰 또는 AVD를 선택하고 'OK' 버튼을 클릭

• 실행 결과


Android Emulator - Nexus_5_API_24:5554
2:57
나를 그냥 둬
근접
거리: 8.1
Extended controls - Nexus_5X_API_27_Oreo_8.1:5554
Location
Cellular
Battery
Camera
Phone
Directional pad
Microphone
Fingerprint
Virtual sensors
Bug report
Screen record
Google Play
Settings
Help
Accelerometer
Additional sensors
Ambient temperature (°C)
0.0
-273.1
100
Magnetic field (North-East-Up µT)
22.00
5.90
43.10
Proximity (cm)
8.1
0
10
Light (lux)
0.0
0
40000
Pressure (hPa)
0.0
0
1100
Relative humidity (%)
0.0
0
100

Outputs