

# Week09.

# 터치센서



# 개발환경 구축 절차

2

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



<https://github.com/hopypark>

# 터치 센서를 활용 앱의 예

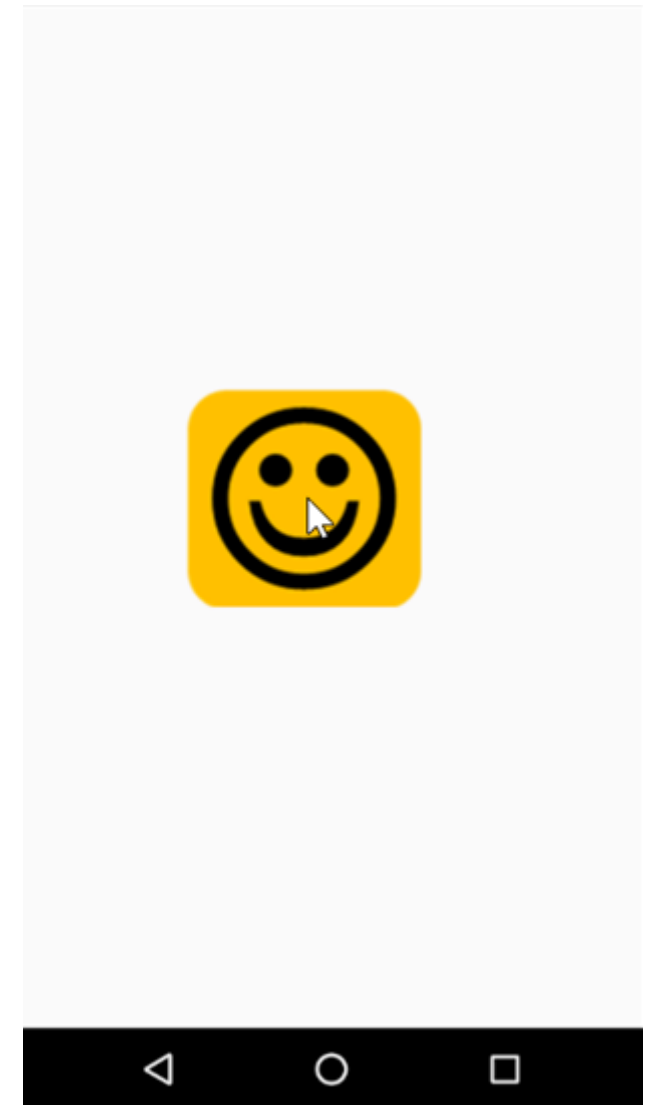
4



(a) 게임레벨 선택



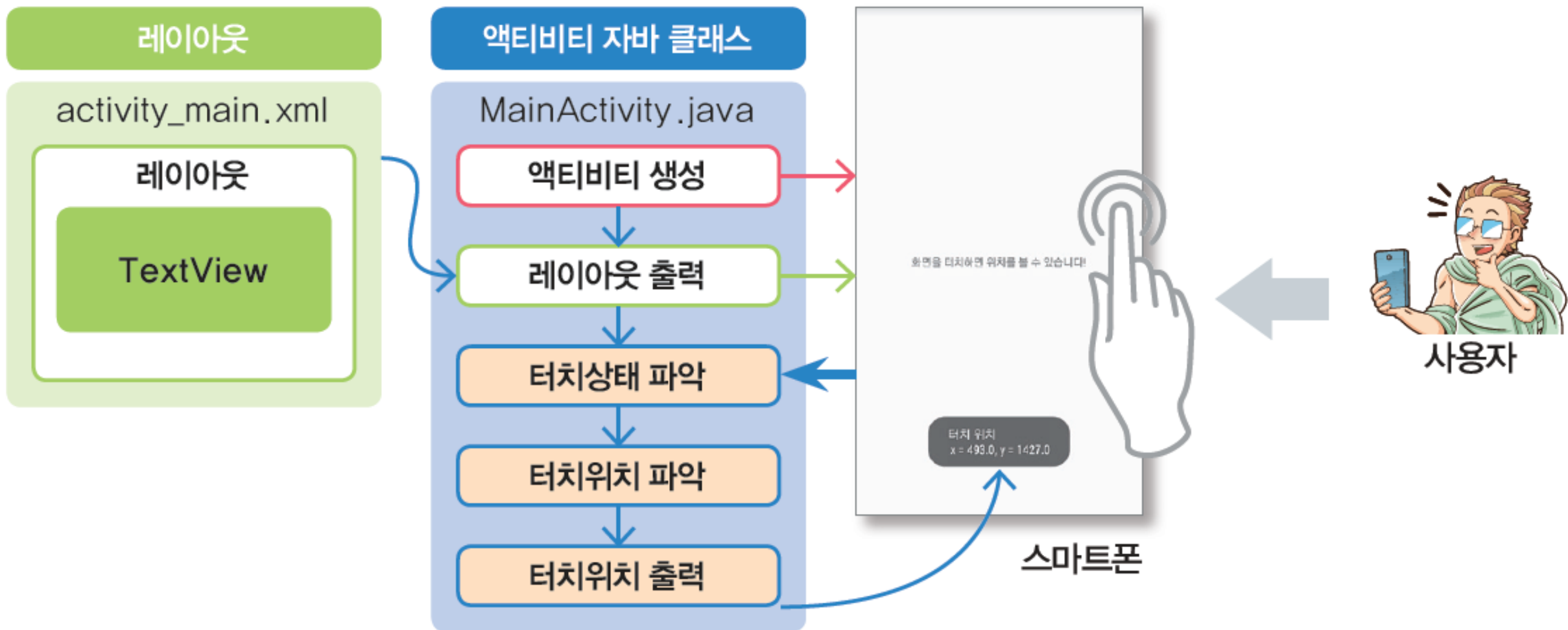
(b) 카카오프렌즈 일렬 배치하기



Follow Me

# 터치 센서 원리

5

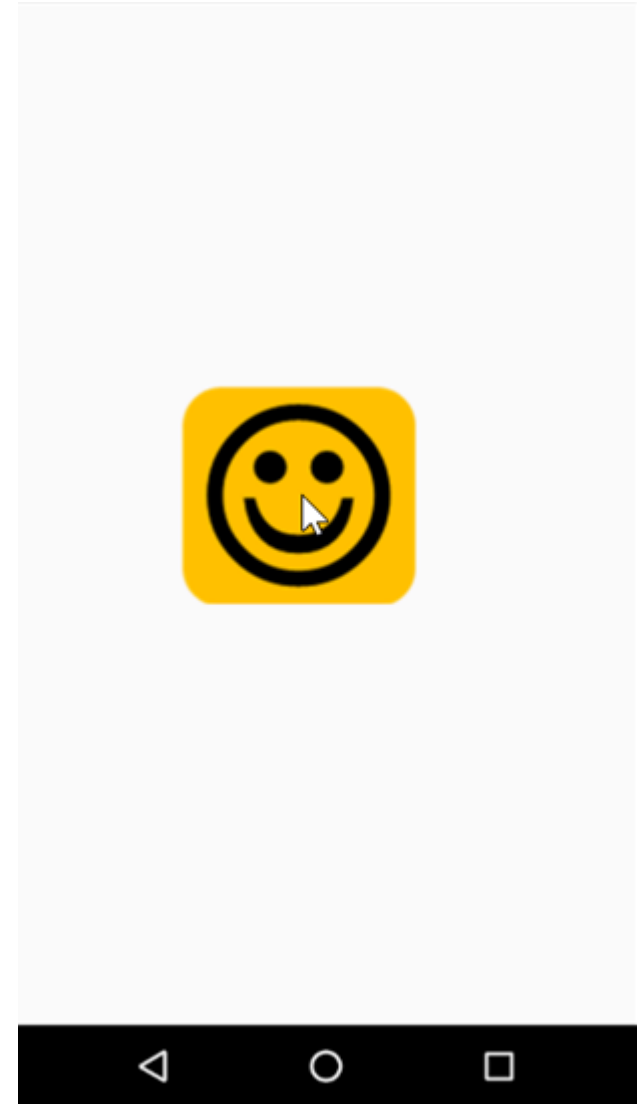
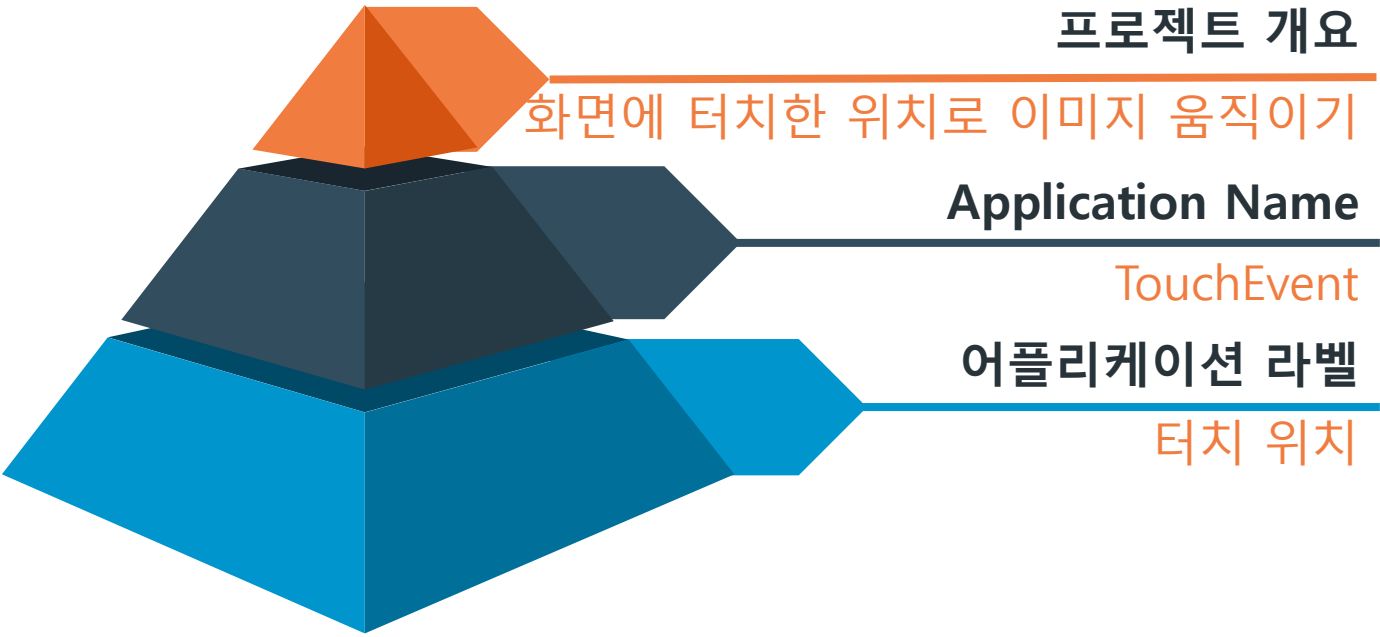


# onTouchEvent() 메소드의 매개변수

터치 이벤트	내용
<code>ACTION_CANCEL</code>	제스처가 중지될 때
<code>ACTION_DOWN</code>	누르는 제스처가 시작될 때
<code>ACTION_MOVE</code>	누르는 동안 이동 등의 변화가 일어남( <code>ACTION_DOWN</code> 과 <code>ACTION_UP</code> 사이)
<code>ACTION_UP</code>	누르는 제스처가 끝남
<code>ACTION_OUTSIDE</code>	제스처가 화면을 벗어남

# Step 0. 프로젝트 개요

7



# Step 1. 프로젝트 생성

8

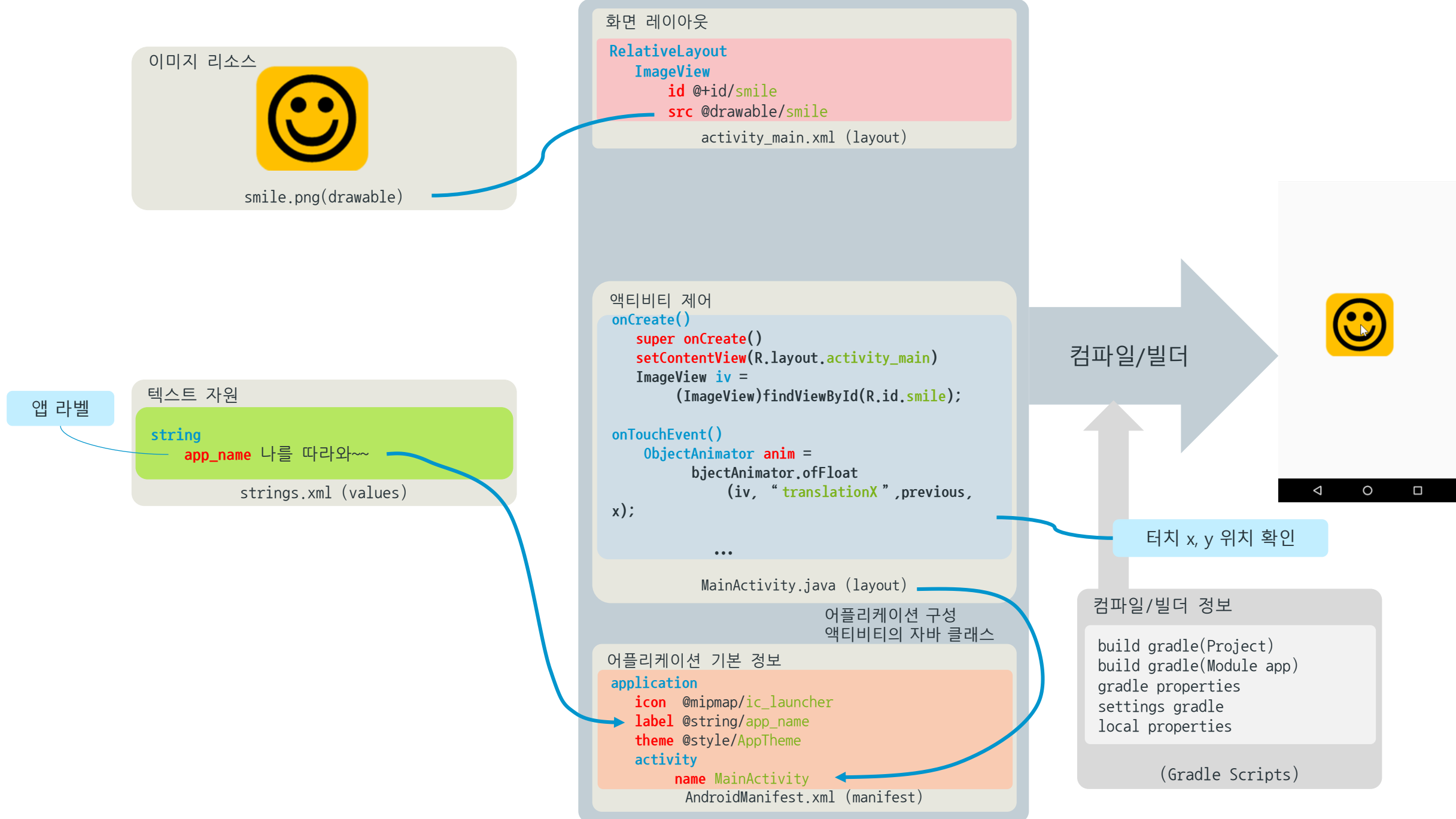
절차	내 용
①프로젝트 시작	메뉴에서 'File → New Project' 클릭
②프로젝트 구성	Application Name: <b>TouchEvent</b>
	Company Domain: <b>kyungtae.example.com</b> (디폴트 사용)
	Project Location: <b>~/AndroidStudioProject/ktpark/TouchEvent</b>
③제품형태	<b>Phone and Tablet</b> (사용할 안드로이드 버전 지정: <b>8.1 Oreo</b> )
④액티비티 유형	<b>Empty Activity</b>
⑤파일 옵션	Activity Name: <b>MainActivity</b> (디폴트 사용)
	Layout Name: <b>activity_main</b> (디폴트 사용)



# Step 2. 파일 편집

9

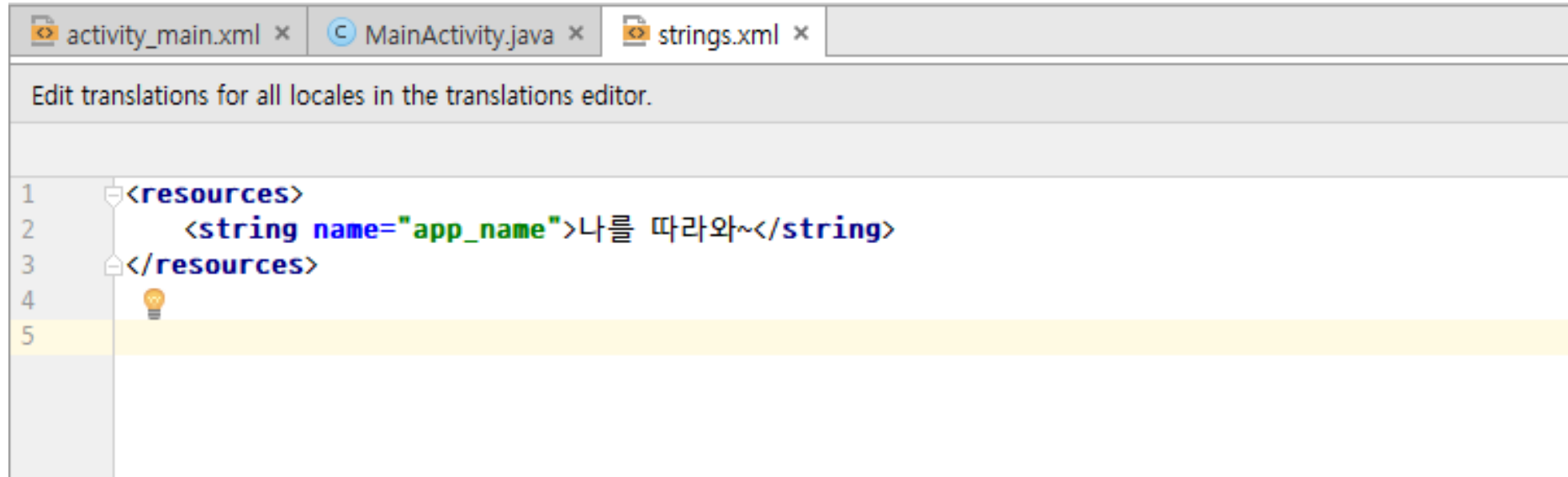
모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.video1	MainActivity.java	• 터치 위치 출력
res	drawable	smile.png	• 스마일 이미지
	layout	activity_main.xml	• 이미지 리소스 출력
	mipmap	ic_launcher.png	
	values	colors.xml	
		dimens.xml	
		styles.xml	



# Step 2.1 텍스트 자원의 편집

11

- strings.xml

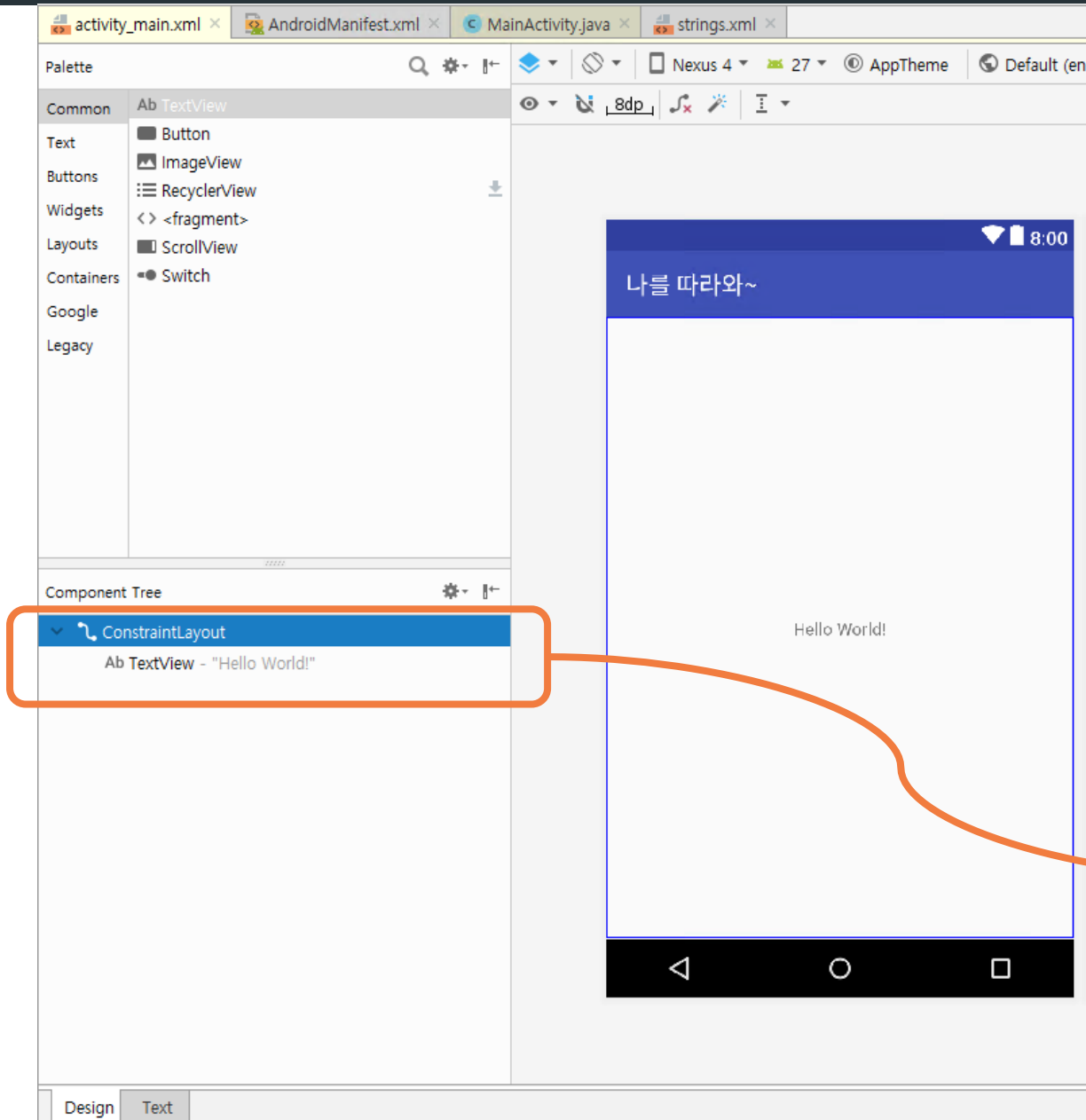


The screenshot shows an IDE window with three tabs: activity\_main.xml, MainActivity.java, and strings.xml. The strings.xml tab is active, displaying the XML content for editing translations. The text "Edit translations for all locales in the translations editor." is shown at the top. The XML code is as follows:

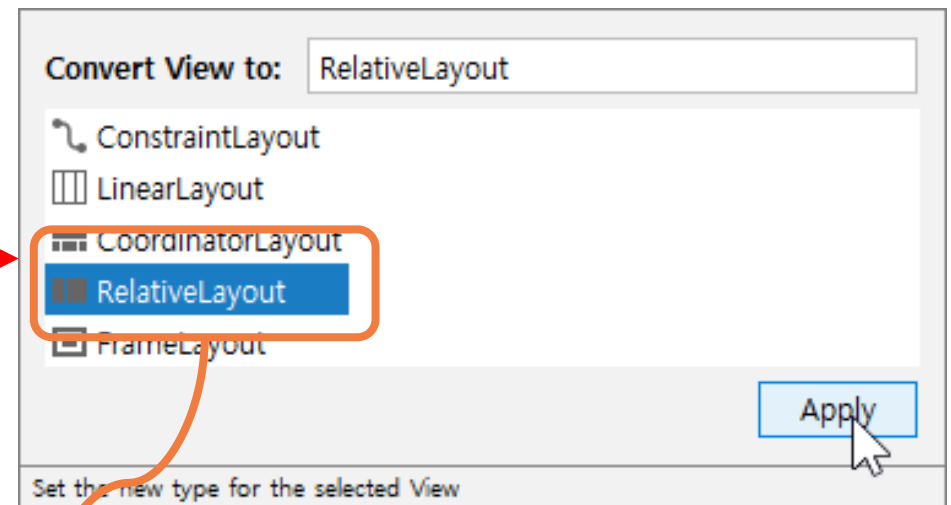
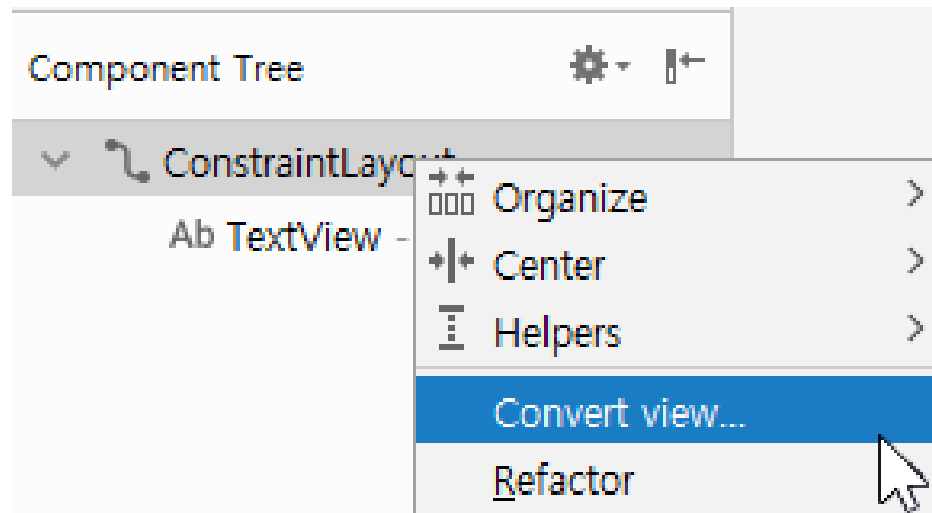
```
1 <resources>
2     <string name="app_name">나를 따라와~</string>
3 </resources>
```

Line 4 is highlighted in yellow, and a lightbulb icon is visible next to it, indicating a suggestion or tip.

## 2.2 화면 설계

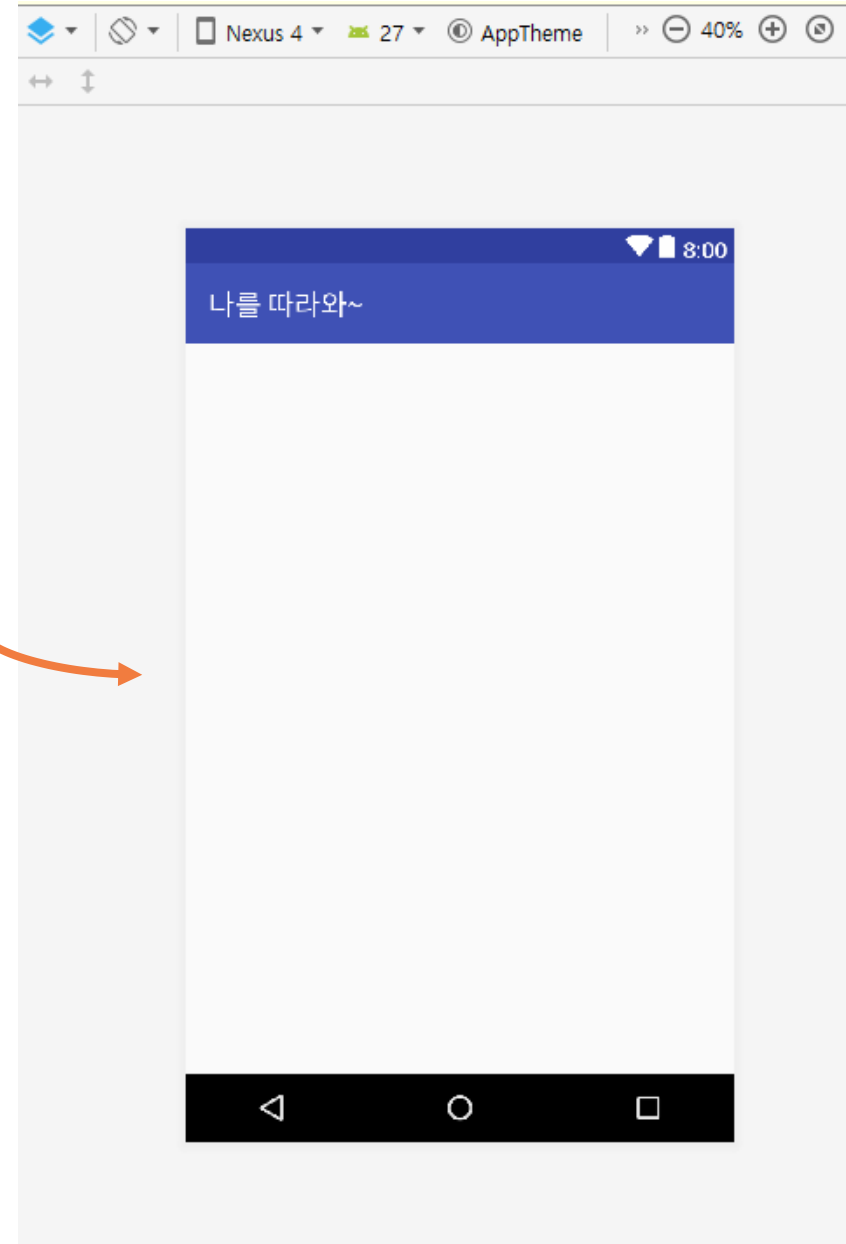
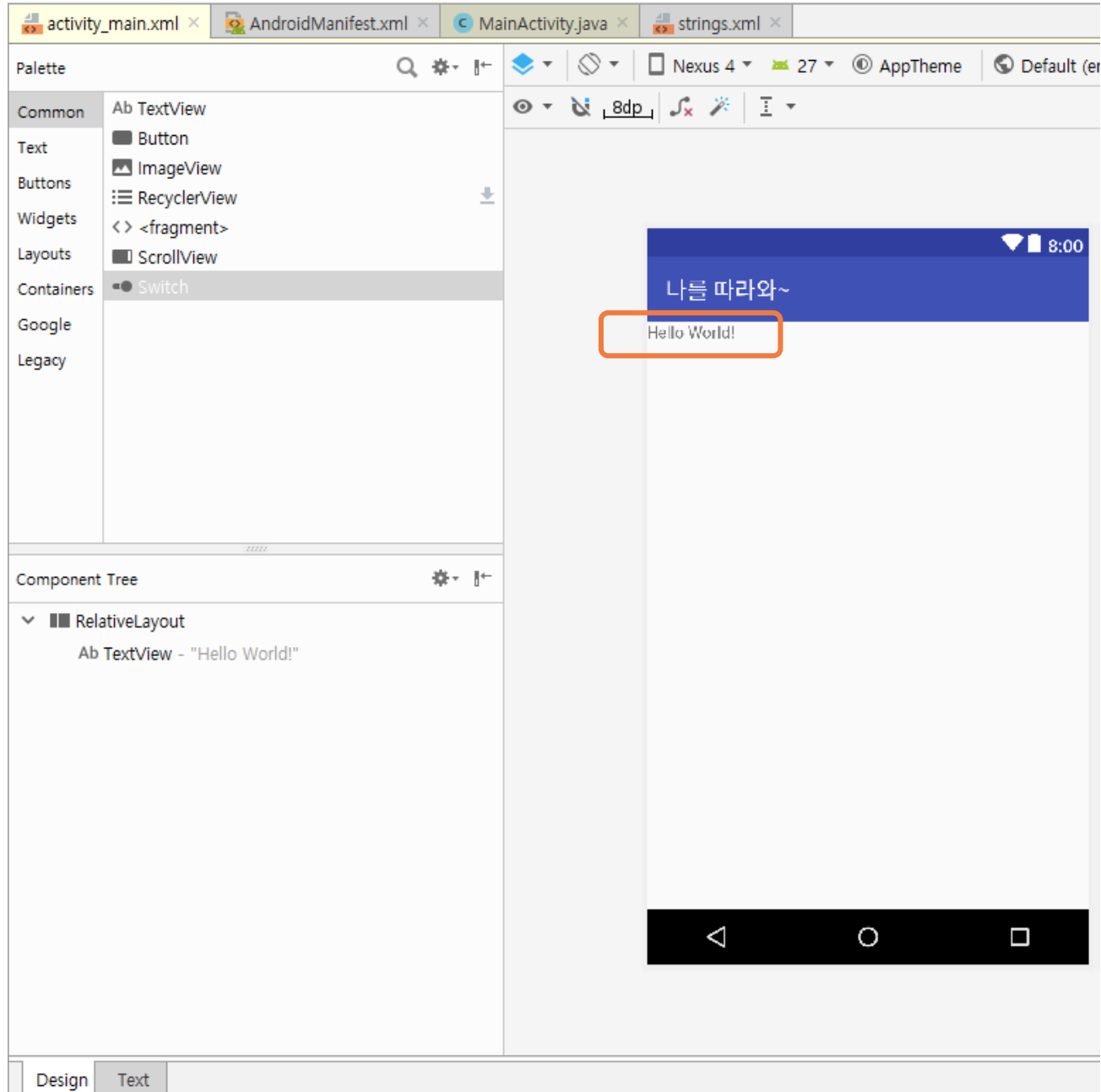


ConstraintLayout →  
RelativeLayout으로 변경

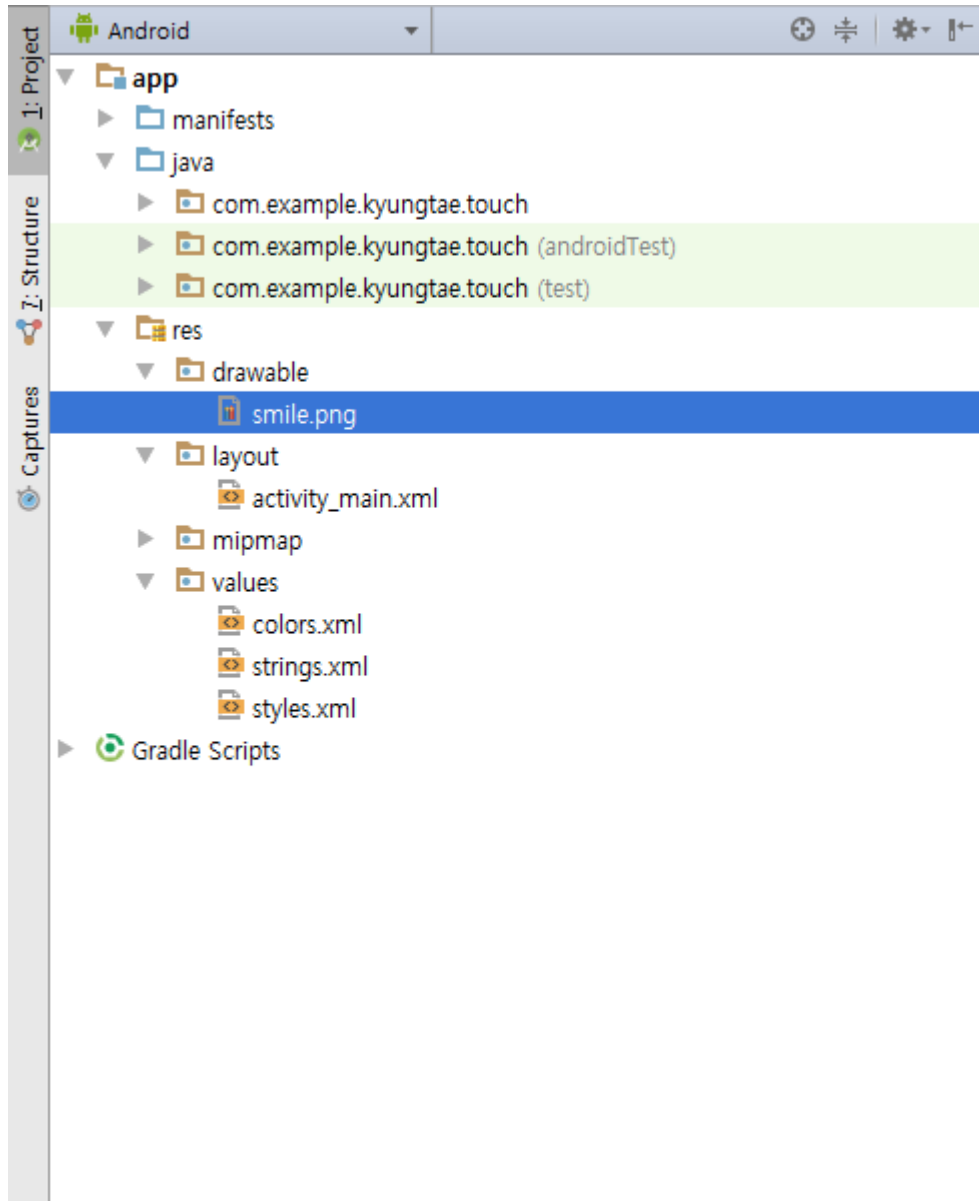


RelativeLayout 선택

# • Layout 초기화 설정 - 기본 TextView 삭제

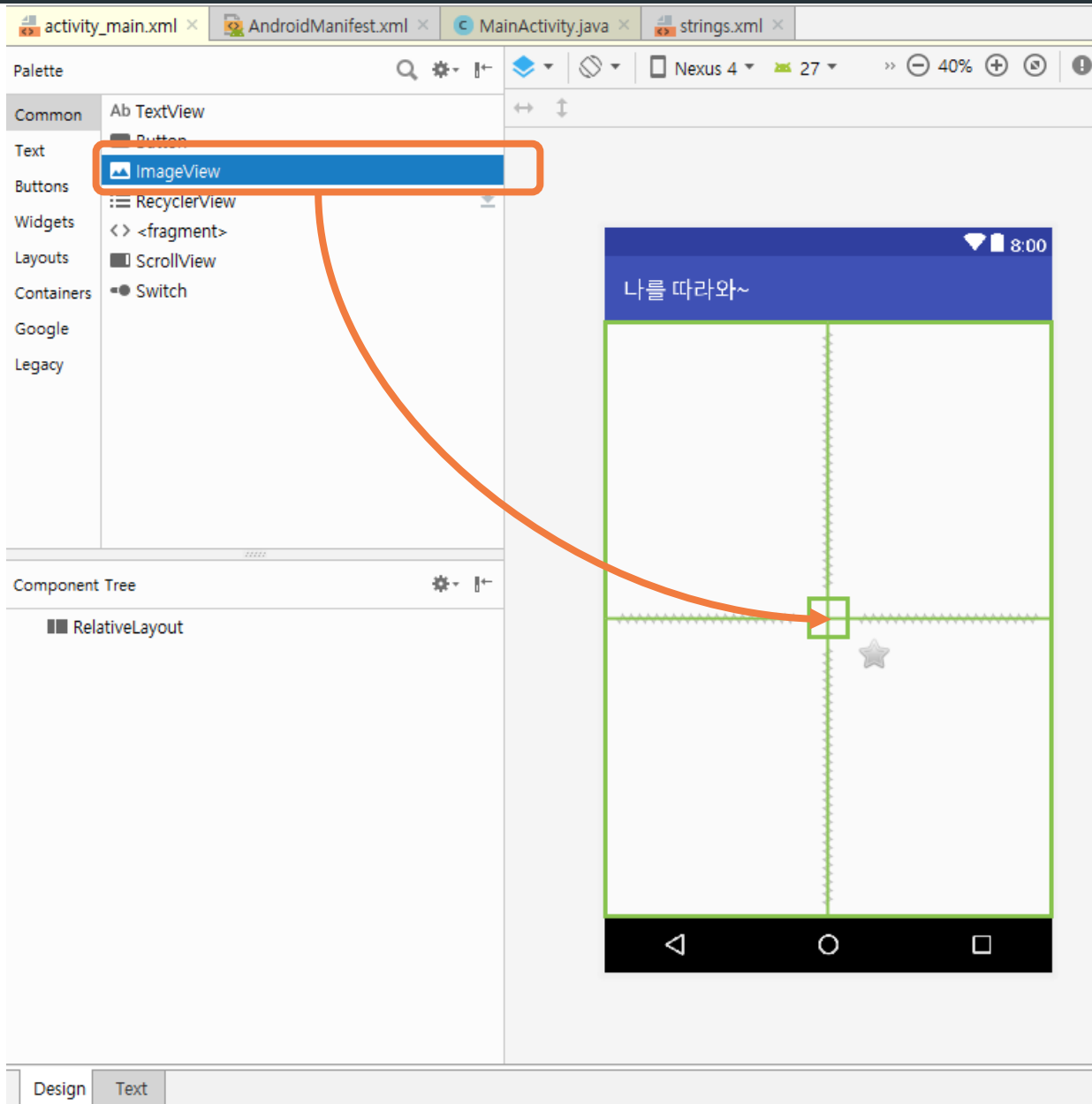


## 2.3 이미지 리소스 – res/drawable 에 smile.png 추가

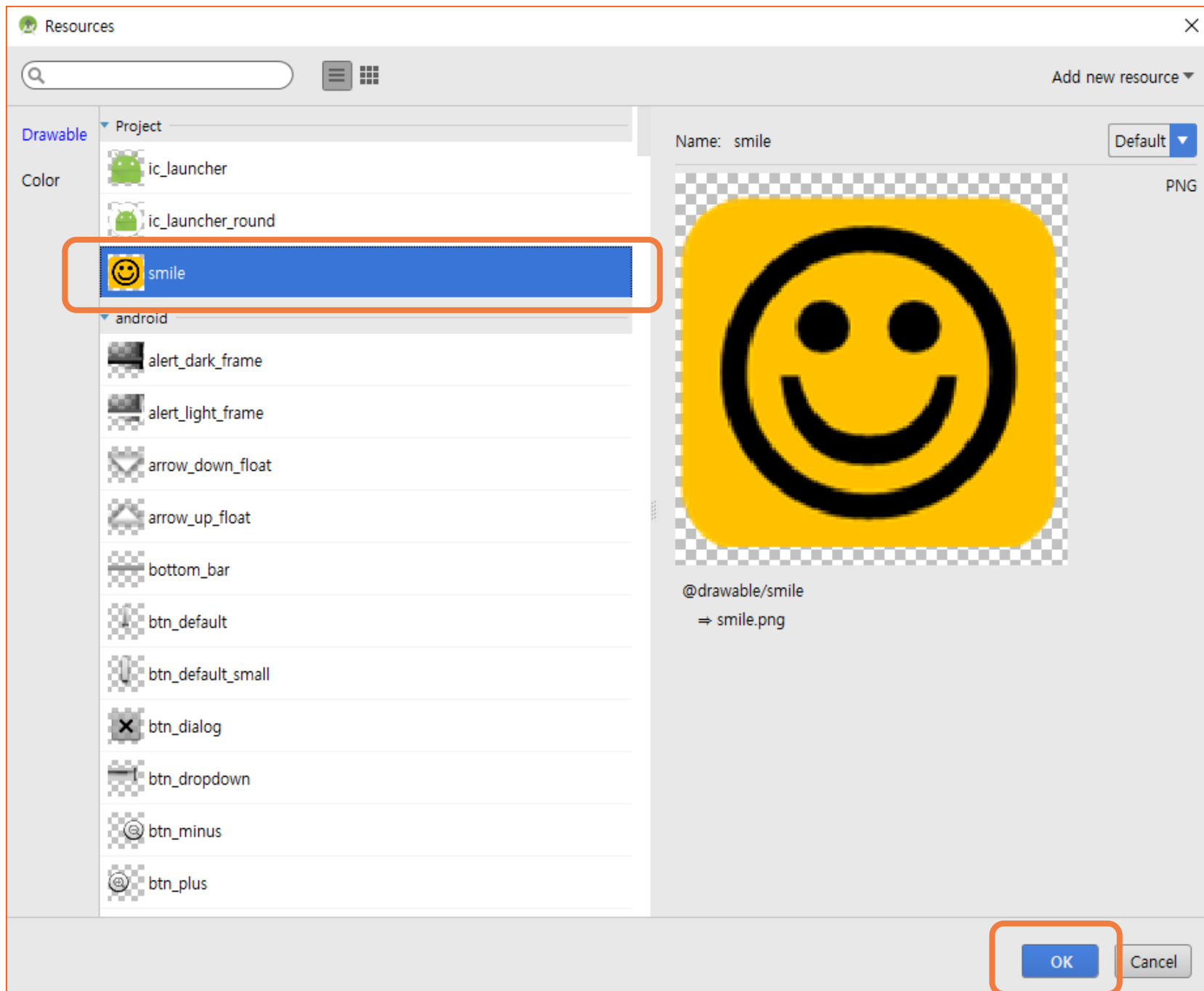


smile.png

## 2.4 화면 설계







- Drawable 항목의 smile을 선택

activity\_main.xml x AndroidManifest.xml x MainActivity.java x strings.xml x

Palette

Common

Ab TextView

Text

Buttons

Widgets

Layouts

Containers

Google

Legacy

Button

ImageView

RecyclerView

<fragment>

ScrollView

Switch

Component Tree

RelativeLayout

smile (ImageView)

Nexus 4

27

AppTheme

40%

Attributes

id

smile

layout\_width

wrap\_content

layout\_height

wrap\_content

Layout\_Margin

[?, ?, ?, ?]

Padding

[?, ?, ?, ?]

Theme

elevation

layout\_centerInParent

☒

srcCompat

@drawable/smile

accessibilityLiveRegion

accessibilityTraversalAfter

accessibilityTraversalBefore

adjustViewBounds

☐

alpha

autofillHints

background

backgroundTint

backgroundTintMode

baseline

baselineAlignBottom

☐

clickable

☐

contentDescription

contextClickable

☐

cropToPadding

☐

defaultFocusHighlightEnabled

☐

drawingCacheQuality

duplicateParentState

☐

fadeScrollbars

☐

fadingEdge

☐

fadingEdgeLength

filterTouchesWhenObscured

☐

fitsSystemWindows

☐

Design

Text

## • ImageView의 src 속성 변경

```
activity_main.xml × AndroidManifest.xml × MainActivity.java × strings.xml ×
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout 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=".MainActivity">
8
9   <ImageView
10     android:id="@+id/smile"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:layout_centerInParent="true"
14     app:srcCompat="@drawable/smile" />
15 </RelativeLayout>
16
17
```

```
nActivity.java × strings.xml ×
f-8"?>
2 <RelativeLayout 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=".MainActivity">
8
9   <ImageView
10     android:id="@+id/smile"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:layout_centerInParent="true"
14     android:src="@drawable/smile" />
15 </RelativeLayout>
16
17
```

# 터치한 위치에 아이콘 이미지 위치시키기

- 이미지의 원점과 터치 위치 사이의 오차처리

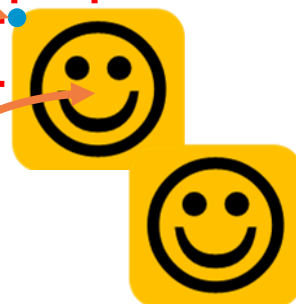
이미지  $x = x - \text{이미지 폭}/2$ ;

이미지  $y = y - \text{이미지 높이}/2$ ;

원점(0,0)

터치 위치:  
(x, y)

이미지 원점 위치:  
(x-이미지폭/2, y-이미지높이/2)



## 2.5 Activity 제어(MainActivity.java)

21

- 화면을 전체화면 크기로 만들기 위한 액티비티 상속 클래스 변경

```
activity_main.xml x MainActivity.java x strings.xml x
1 package com.example.kyungtae.touch;
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
```

```
activity_main.xml x MainActivity.java x strings.xml x
1 package com.example.kyungtae.touch;
2
3 import android.support.v4.app.FragmentActivity;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6
7 public class MainActivity extends FragmentActivity {
8
9     @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13     }
14 }
15
```

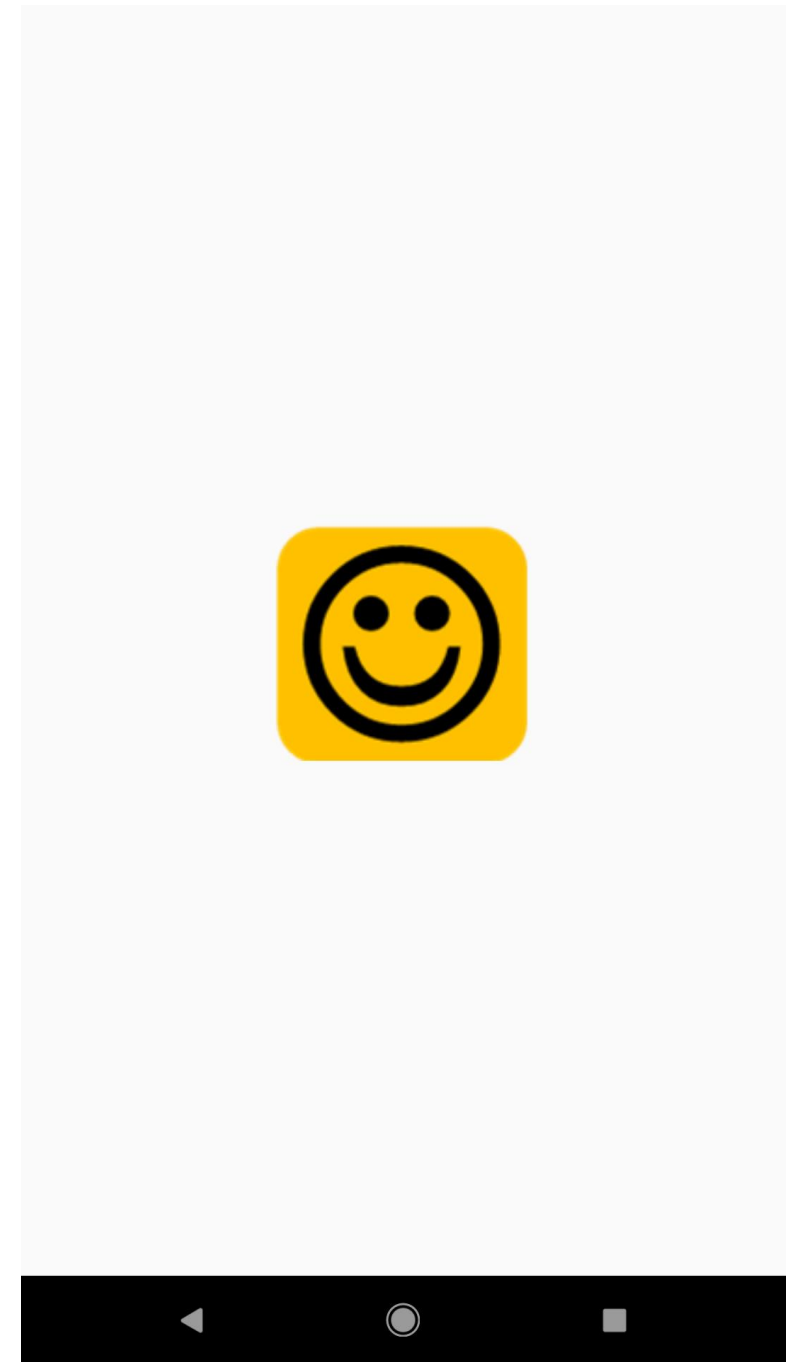
화면 상단의 액션 바(제목창) 나타나지 않는 전체화면 크기로 만들기

```
activity_main.xml × MainActivity.java ×  
1 package com.example.user.touchevent;  
2  
3 import android.animation.ObjectAnimator;  
4 import android.support.v4.app.FragmentActivity;  
5 import android.os.Bundle;  
6 import android.util.DisplayMetrics;  
7 import android.view.MotionEvent;  
8 import android.view.WindowManager;  
9 import android.widget.ImageView;  
10  
11 public class MainActivity extends FragmentActivity {  
12  
13     ImageView iv_Smile; // smile 이미지용 객체  
14     float previousX = 0f; // 이전 x 위치  
15     float previousY = 0f; // 이전 y 위치  
16  
17     float screen_Width = 0f; // 화면의 폭  
18     float screen_Height = 0f; // 화면의 높이  
19  
20     float iv_Width = 0f; // 이미지의 폭  
21     float iv_Height = 0f; // 이미지의 높이  
22  
23     float x, y; // 아이콘의 임시 x,y 위치  
24
```



타이틀 - 안테나, 배터리 정보  
표시 장 없애기

화면 중앙에 이미지가 위치



@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

*// 윈도우를 전체 화면 크기로 설정*

```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
    WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```
setContentView(R.layout.activity_main);
```

```
iv_Smile = (ImageView) findViewById(R.id.smile);
```

*// 화면 전체 크기 얻기*

```
DisplayMetrics displayMetrics = getApplicationContext().getResources().getDisplayMetrics();
```

```
screen_Width = displayMetrics.widthPixels;
```

```
screen_Height = displayMetrics.heightPixels;
```

*// 이미지 크기 얻기*

```
iv_Smile.measure(iv_Smile.getMeasuredWidth(), iv_Smile.getMeasuredHeight());
```

```
iv_Height = iv_Smile.getMeasuredHeight();
```

```
iv_Width = iv_Smile.getMeasuredWidth();
```

*// 화면의 중앙위치 계산*

```
x = screen_Width / 2 - iv_Width / 2;
```

```
y = screen_Height / 2 - iv_Height / 2;
```

*// 화면의 중앙으로 이미지를 위치시킴*

```
iv_Smile.setX(x);
```

```
iv_Smile.setY(y);
```

*// 이전 위치로 중앙을 설정*

```
previousX = x;
```

```
previousY = y;
```

```
}
```

안테나와 배터리 정보창 제거

이미지가 위치할 중앙 위치



# onTouchEvent() 매소드 재정의(Override)

25

The image shows a sequence of three screenshots from an IDE, illustrating the steps to override the `onTouchEvent()` method in an Android application.

**Left Screenshot:** The IDE's menu bar is visible, with the **Code** menu open. The **Override Methods...** option is highlighted, which has the keyboard shortcut **Ctrl+O**. The project structure on the left shows a project named **Touch** with a sub-project **app**.

**Middle Screenshot:** The **Select Methods to Override/Implement** dialog box is open. It displays a list of methods from the **android.app.Activity** class. The **android.app.Activity** class is selected in the hierarchy, and the **onTouchEvent(event: MotionEvent): boolean** method is highlighted in the list. At the bottom, the **Insert @Override** checkbox is checked.

**Right Screenshot:** The same **Select Methods to Override/Implement** dialog box is shown, but now the **onTouchEvent(event: MotionEvent): boolean** method is selected in the list. The **Insert @Override** checkbox remains checked.

- 추가된 onTouchEvent() 매소드

```
@Override  
public boolean onTouchEvent(MotionEvent event) {  
    return super.onTouchEvent(event);  
}
```

➔ 수정은 다음 페이지

```
58 @Override
59 public boolean onTouchEvent(MotionEvent event) {
60
61     switch (event.getAction()){
62         case MotionEvent.ACTION_DOWN:
63             break;
64         case MotionEvent.ACTION_MOVE:
65             int touch_x = (int) event.getX();
66             int touch_y = (int) event.getY();
67
68             x = touch_x - iv_Width / 2;
69             y = touch_y - iv_Height / 2;
70
71             ObjectAnimator smileX = ObjectAnimator.ofFloat(iv_Smile, propertyName: "translationX", previousX, x);
72             smileX.start();
73             ObjectAnimator smileY = ObjectAnimator.ofFloat(iv_Smile, propertyName: "translationY", previousY, y);
74             smileY.start();
75
76             previousX = x;
77             previousY = y;
78
79             break;
80         case MotionEvent.ACTION_UP:
81             break;
82     }
83
84     return false;
85 }
86 }
```

# 클래스와 속성/메소드

- 클래스

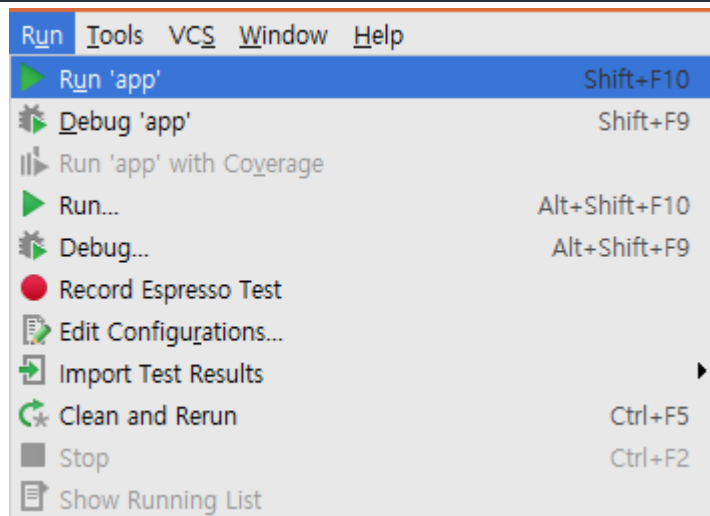
클래스	설명
ObjectAnimator	목표 객체에 대한 애니메이션 특성을 설정

- 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									

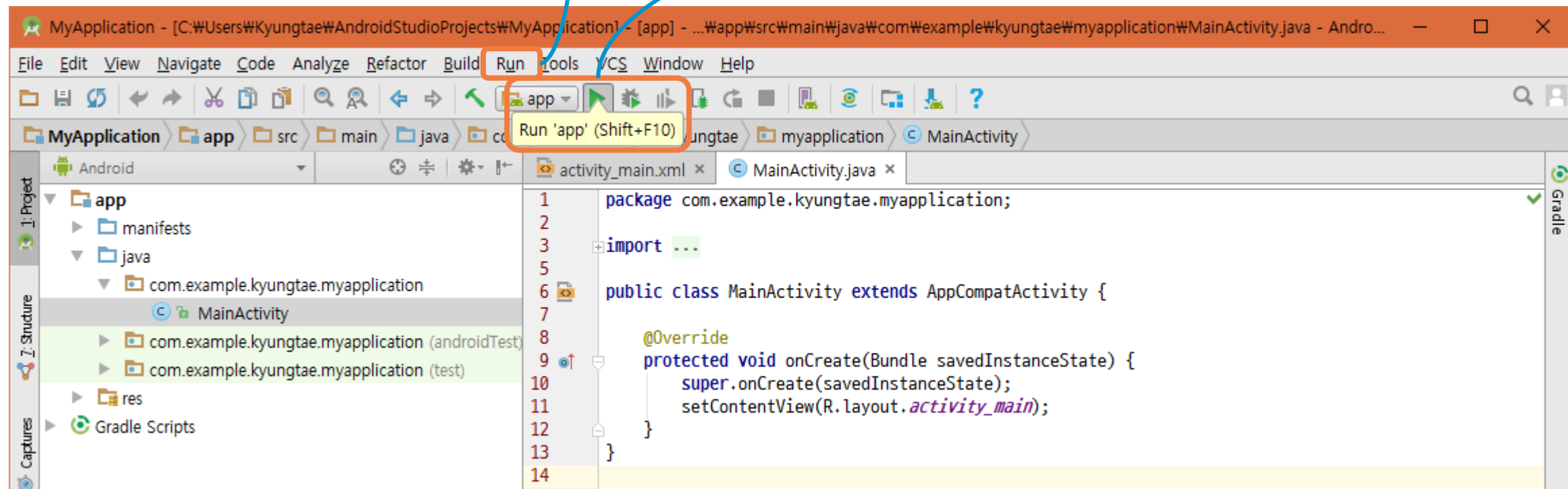
# Step 3. 프로젝트 실행

29



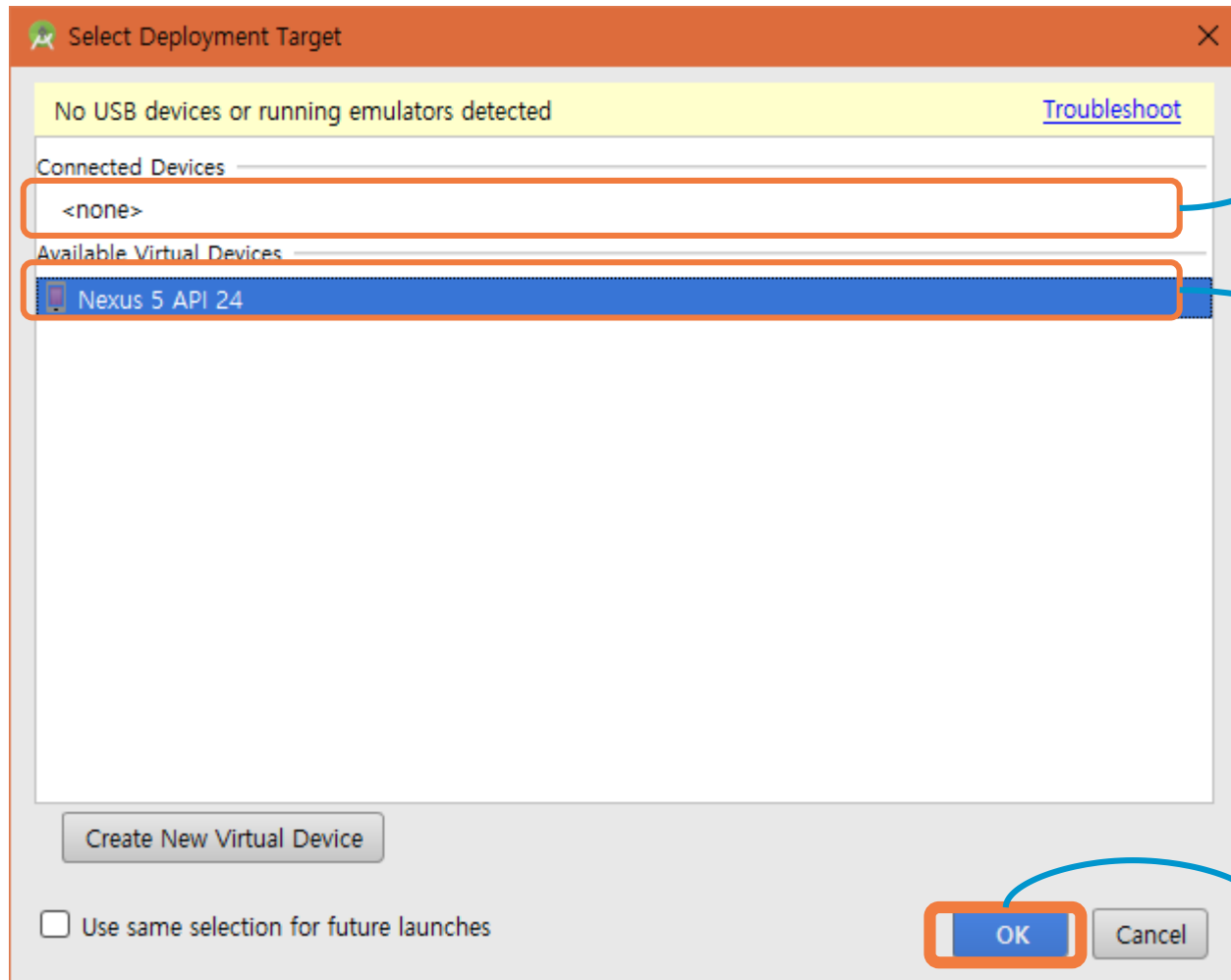
Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



## • AVD 장비 선택하기

30



데이터 케이블로 연결된  
스마트폰

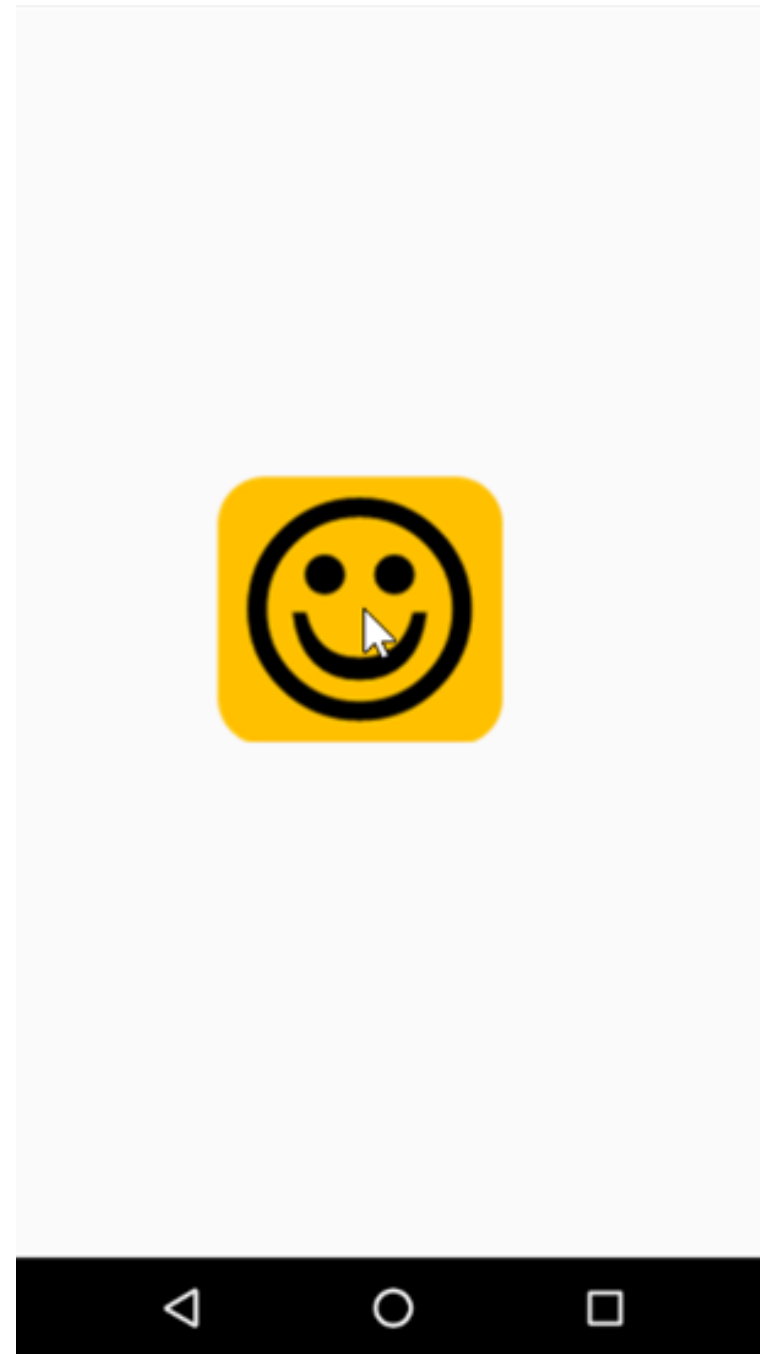
AVD

스마트폰 또는 AVD를 선택하고  
'OK' 버튼을 클릭

- 실행 결과



# O utputs





# Q & A uestion nswer

35

