Week09.

터치센서

개발환경 구축 절차

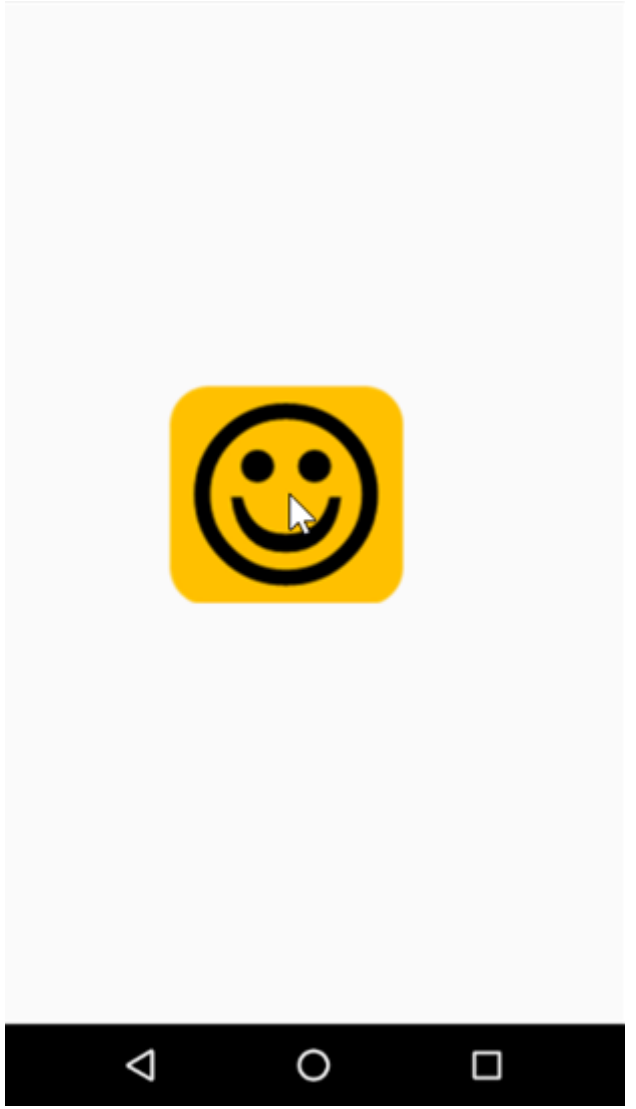
주 차	수 업 내 용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	이미지의 출력
5	이벤트 처리와 액티비티 간 이동
6	오디오 재생
7	비디오 재생
8	**중간고사**
9	애니메이션
10	**사물인터넷과 센서 – 터치 센서,** 모션 센서
11	사물인터넷과 센서 – 위치 센서, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활요
14	구글 맵과 위치 추적
15	**기말 고사**

http://github.com/hopypark

터치 센서를 활용 앱의 예

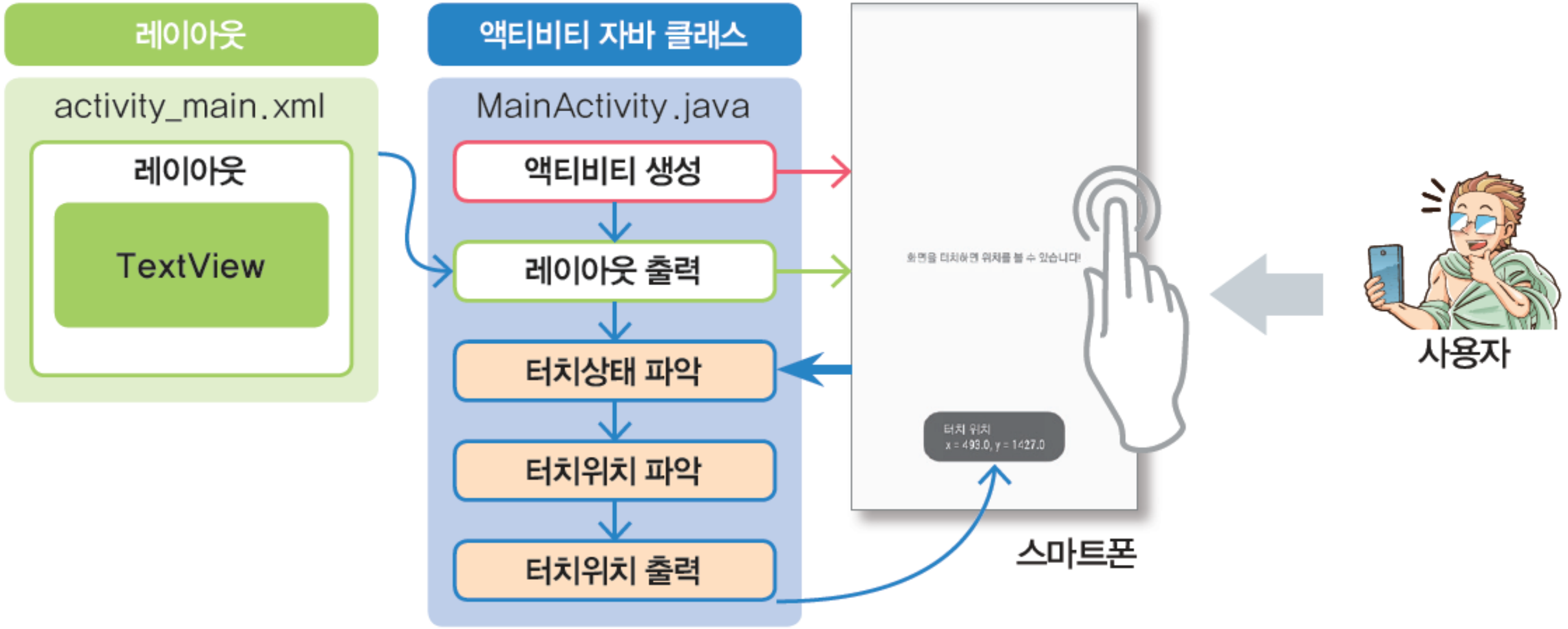
(a) 게임레벨 선택

(b) 카카오프렌즈 일렬 배치하기



Follow Me

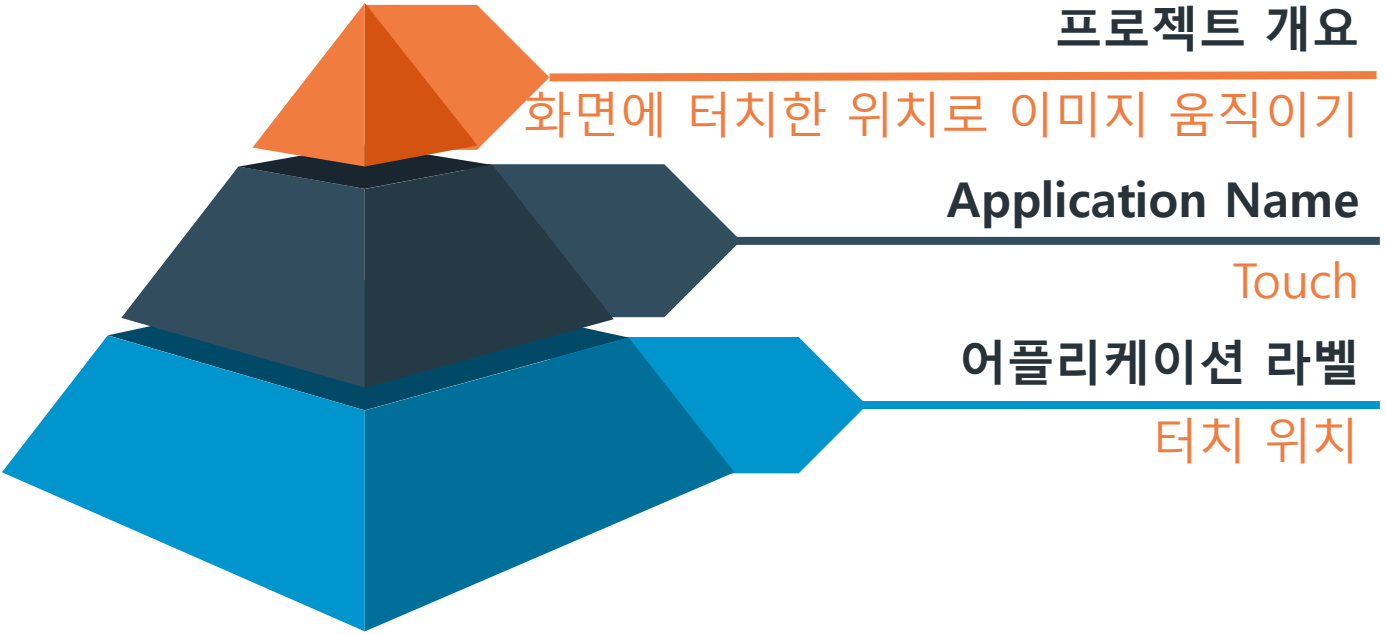
터치 센서 원리


레이아웃
activity_main.xml
레이아웃
TextView
액티비티 자바 클래스
MainActivity.java
액티비티 생성
레이아웃 출력
터치상태 파악
터치위치 파악
터치위치 출력
화면을 터치하면 위치를 볼 수 있습니다!
터치 위치
x = 493.0, y = 1427.0
스마트폰
사용자

onTouchEvent() 메소드의 매개변수

터치 이벤트	내용
ACTION_CANCEL	제스처가 중지될 때
ACTION_DOWN	화면을 터치할 때
ACTION_MOVE	화면을 터치한 상태에서 움직일 때(ACTION_DOWN과 ACTION_UP 사이)
ACTION_UP	화면 터치가 종료될 때
ACTION_OUTSIDE	터치가 화면을 벗어날 때

Step 0.프로젝트 개요



프로젝트 개요

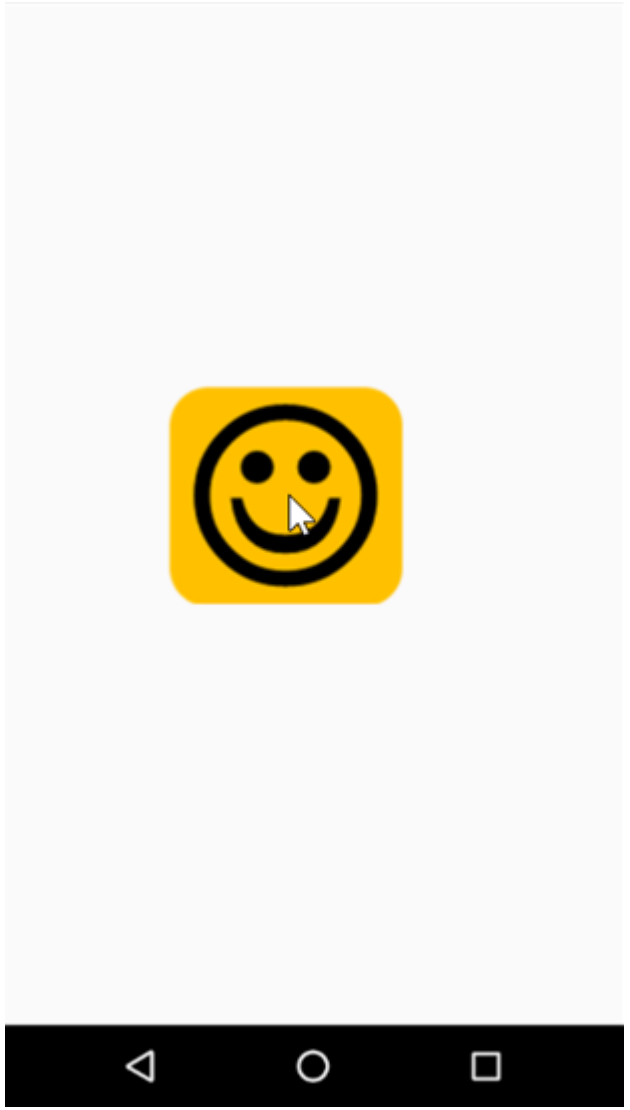
화면에 터치한 위치로 이미지 움직이기

Application Name

Touch

어플리케이션 라벨

터치 위치



Step 1. 프로젝트 생성

절차	내 용
①프로젝트 시작	메뉴에서 ‘File → New Project’ 클릭
②프로젝트 구성	Application Name: **Touch**
	Company Domain: kyungtae.example.com(디폴트 사용)
	Project Location: ~/AndroidStudioProject/ktpark/Touch
③제품형태	Phone and Tablet(사용할 안드로이드 버전 지정: 7.0 Nougat)
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	• 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	

이미지 리소스

smile.png(drawable)

화면 레이아웃

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

activity_main.xml (layout)

액티비티 제어

```
onCreate()
    super onCreate()
    setContentView(R.layout.activity_main)
    ImageView iv =
        (ImageView)findViewById(R.id.smile);

onTouchEvent()
    ObjectAnimator anim =
        bjectAnimator.ofFloat
            (iv, " transitionX ",previous, x);

            ...
```

MainActivity.java (layout)

앱 라벨

텍스트 자원

```
string
    app_name 터치 위치
```

strings.xml (values)

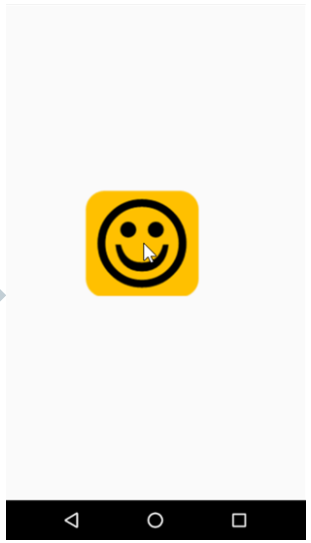
어플리케이션 구성
액티비티의 자바 클래스

어플리케이션 기본 정보

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

AndroidManifest.xml (manifest)

컴파일/빌더



터치 x, y 위치 확인

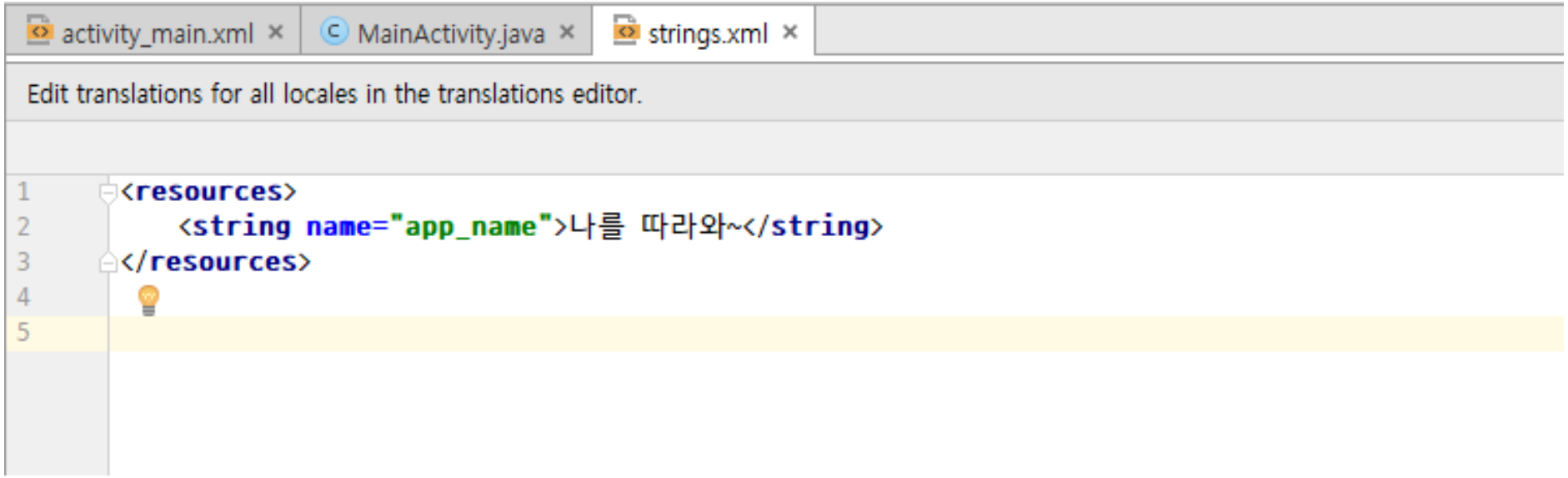
컴파일/빌더 정보

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

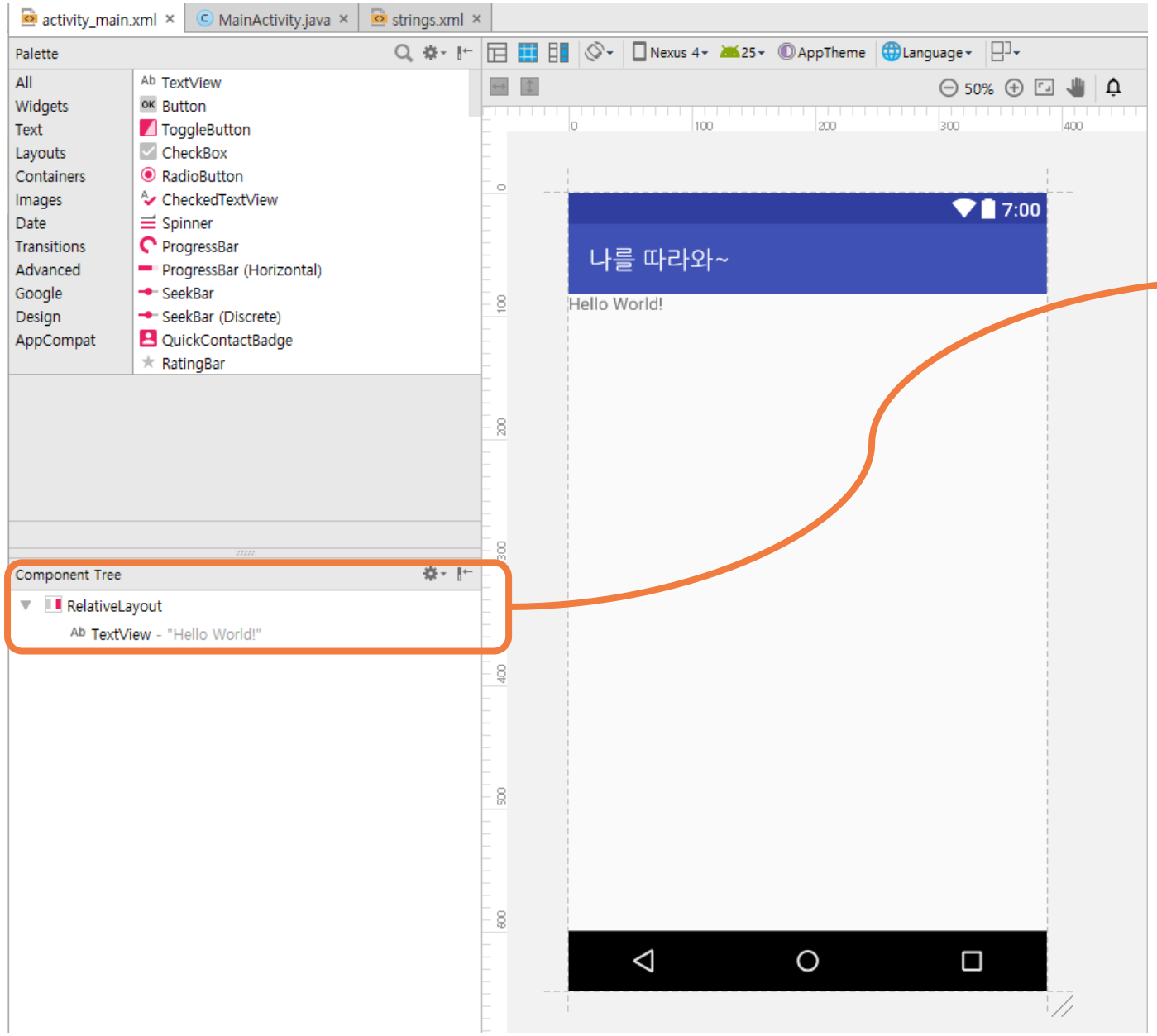
(Gradle Scripts)

Step 2.1 텍스트 자원의 편집

- strings.xml

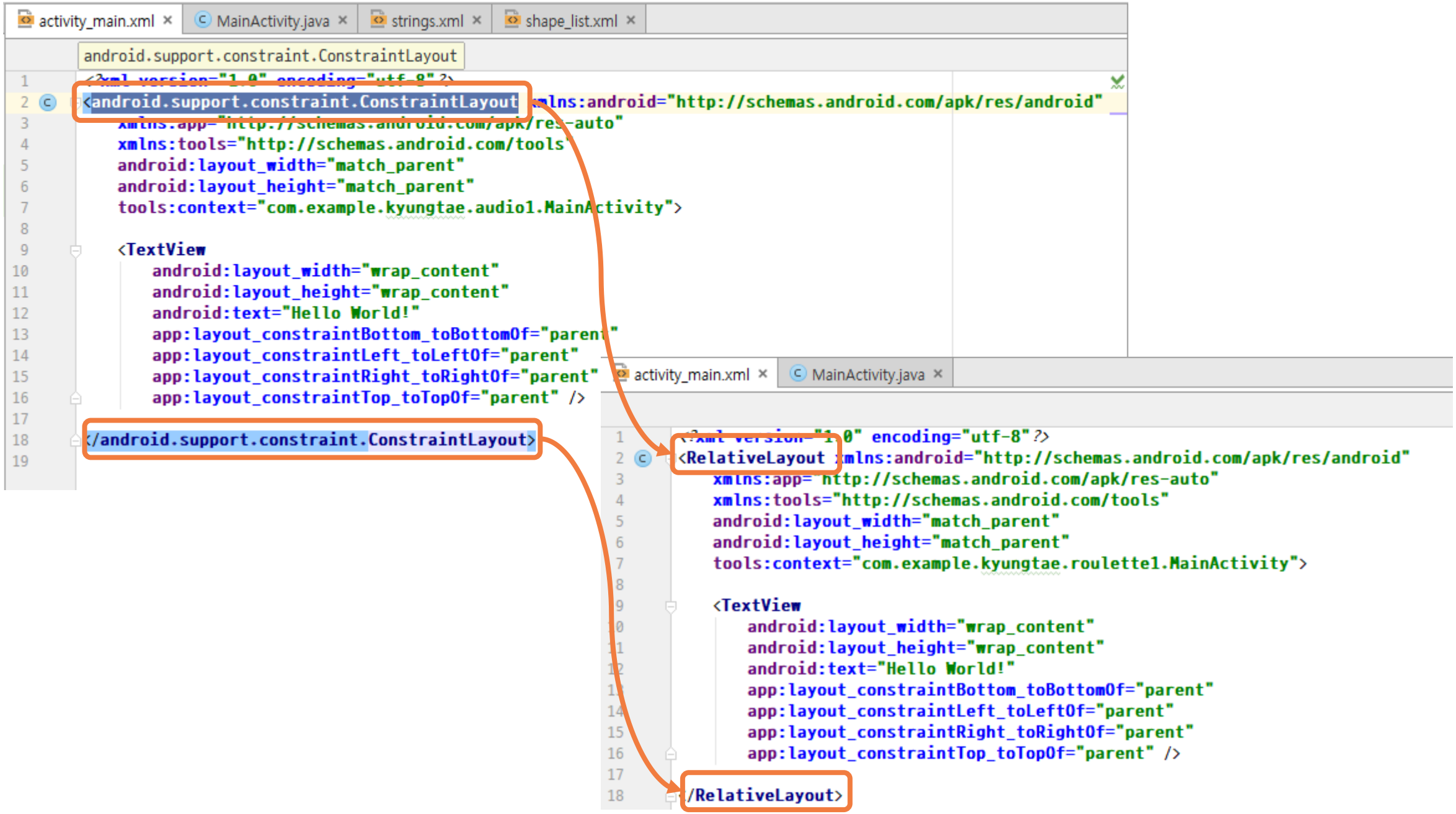


2.2 화면 설계

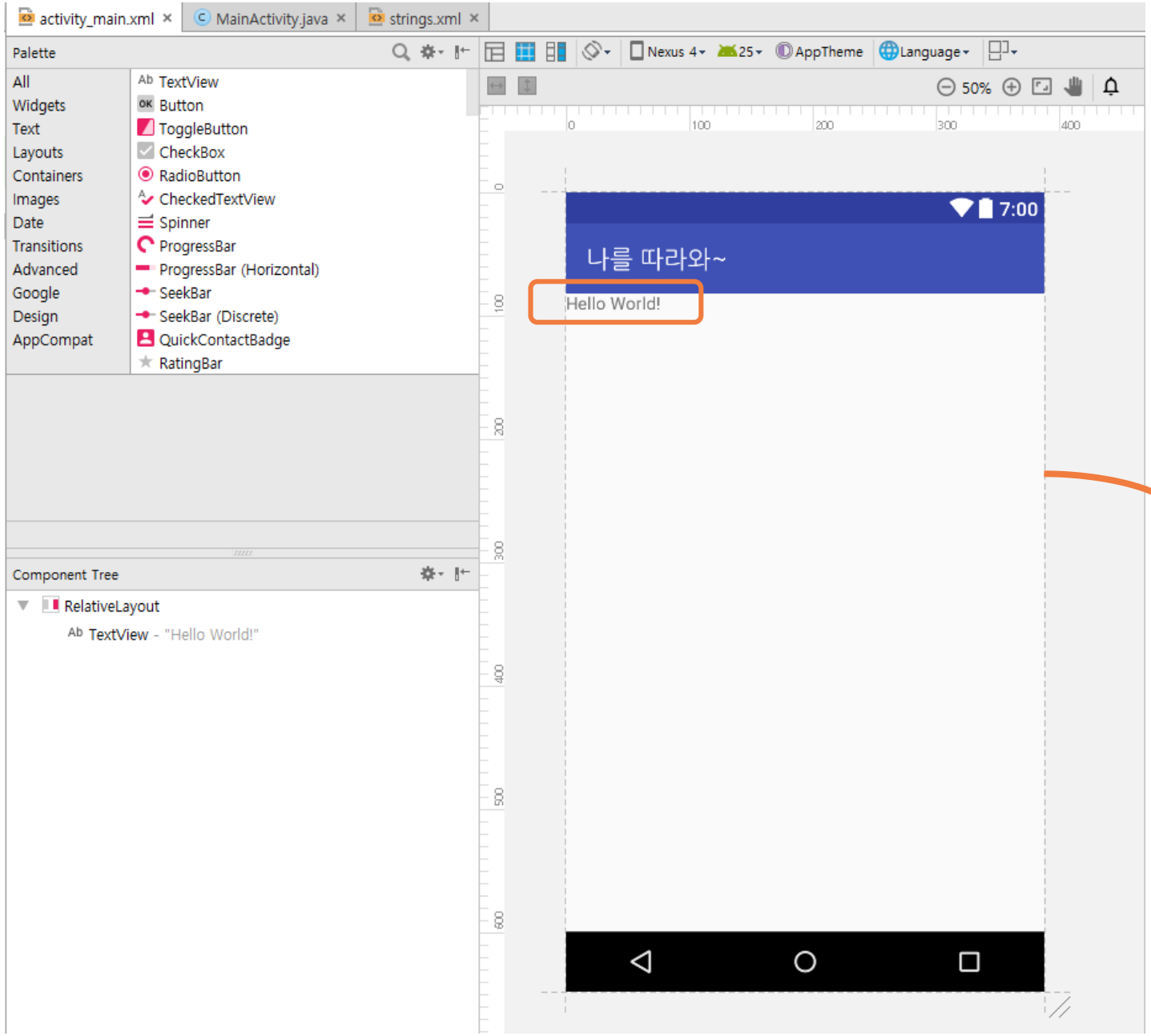


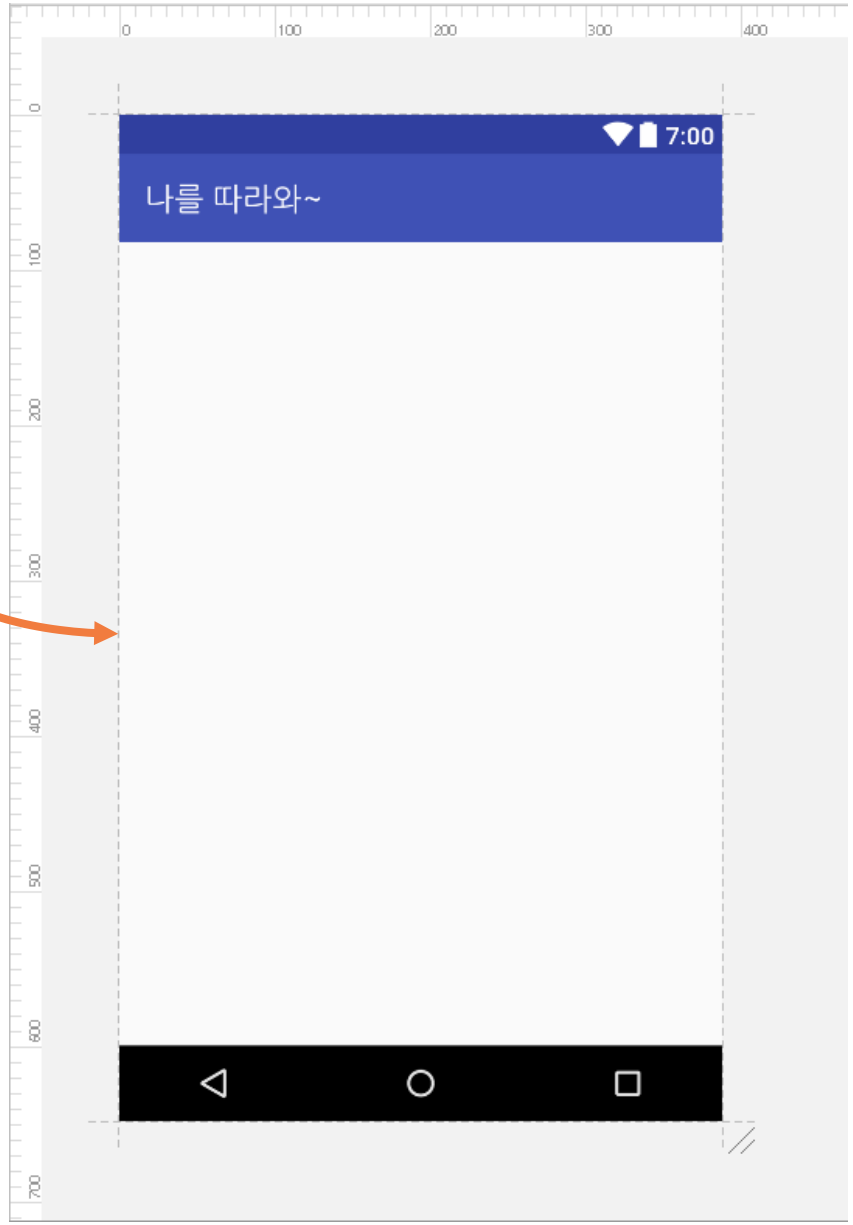
ConstraintLayout → RelativeLayout으로 변경

Text 에디터에서 수정

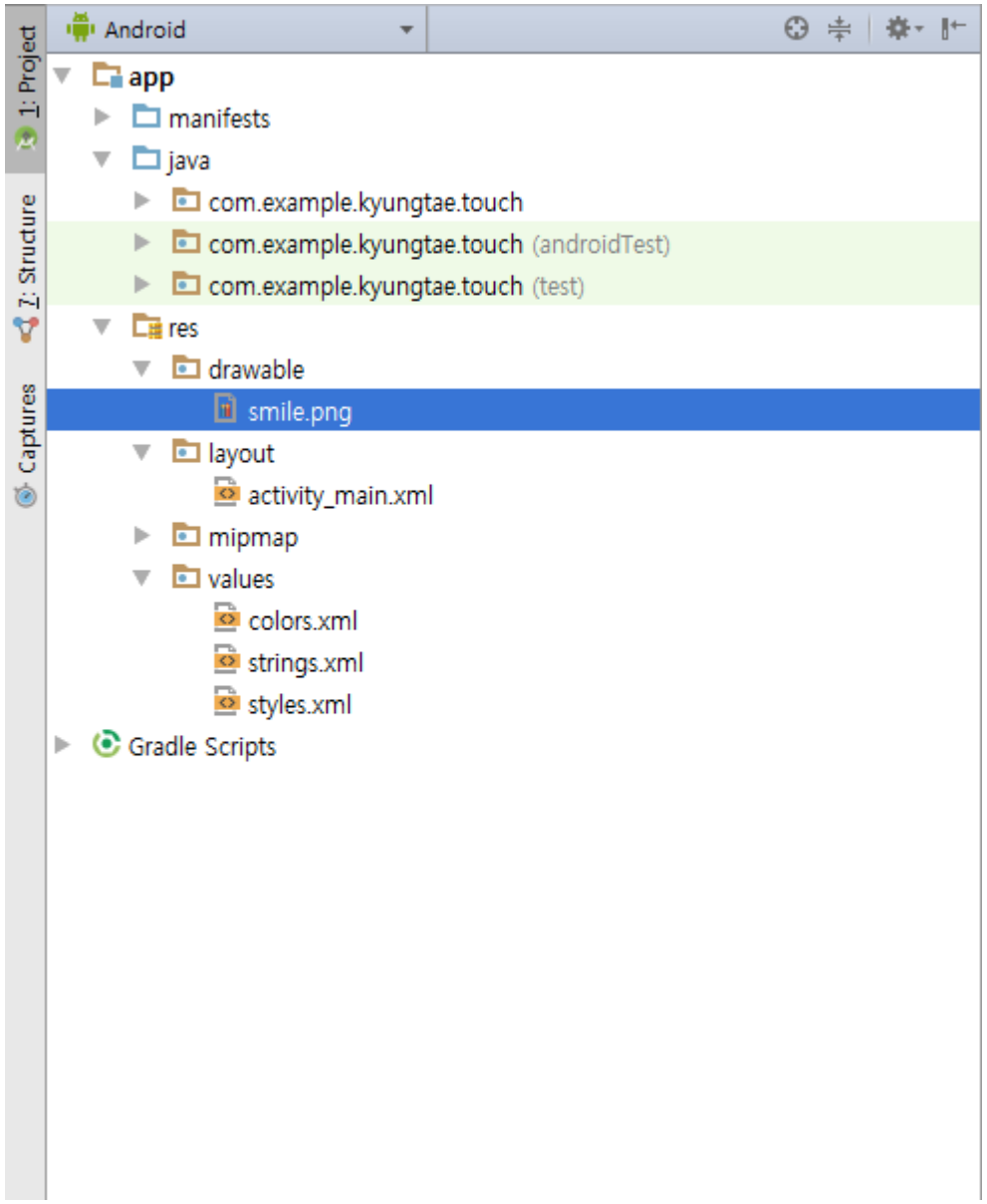

activity_main.xml
MainActivity.java
strings.xml
shape_list.xml
android.support.constraint.ConstraintLayout
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.kyungtae.audio1.MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
activity_main.xml
MainActivity.java
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.kyungtae.roulette1.MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</RelativeLayout>

- Layout 초기화 설정 - 기본 TextView 삭제



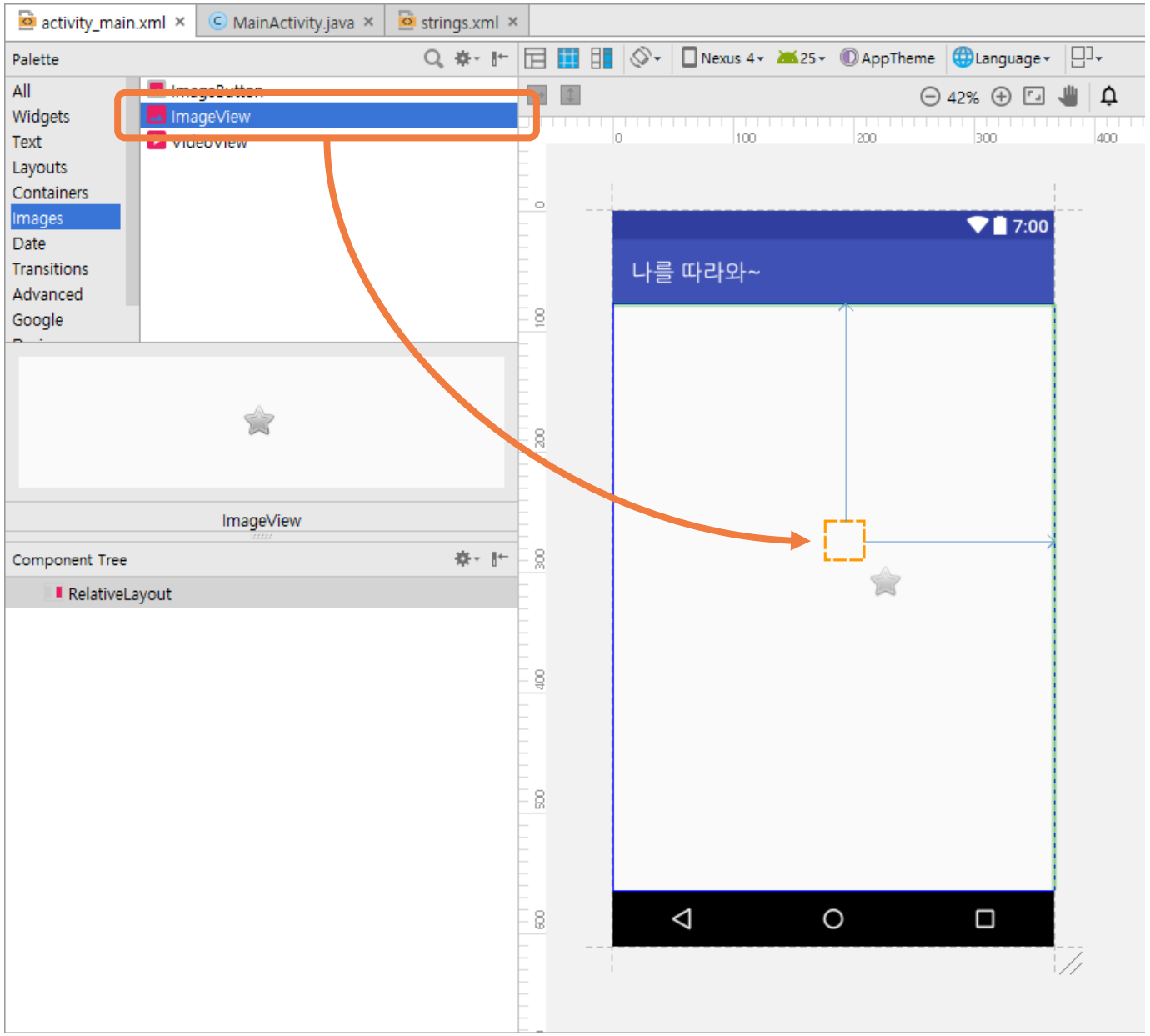


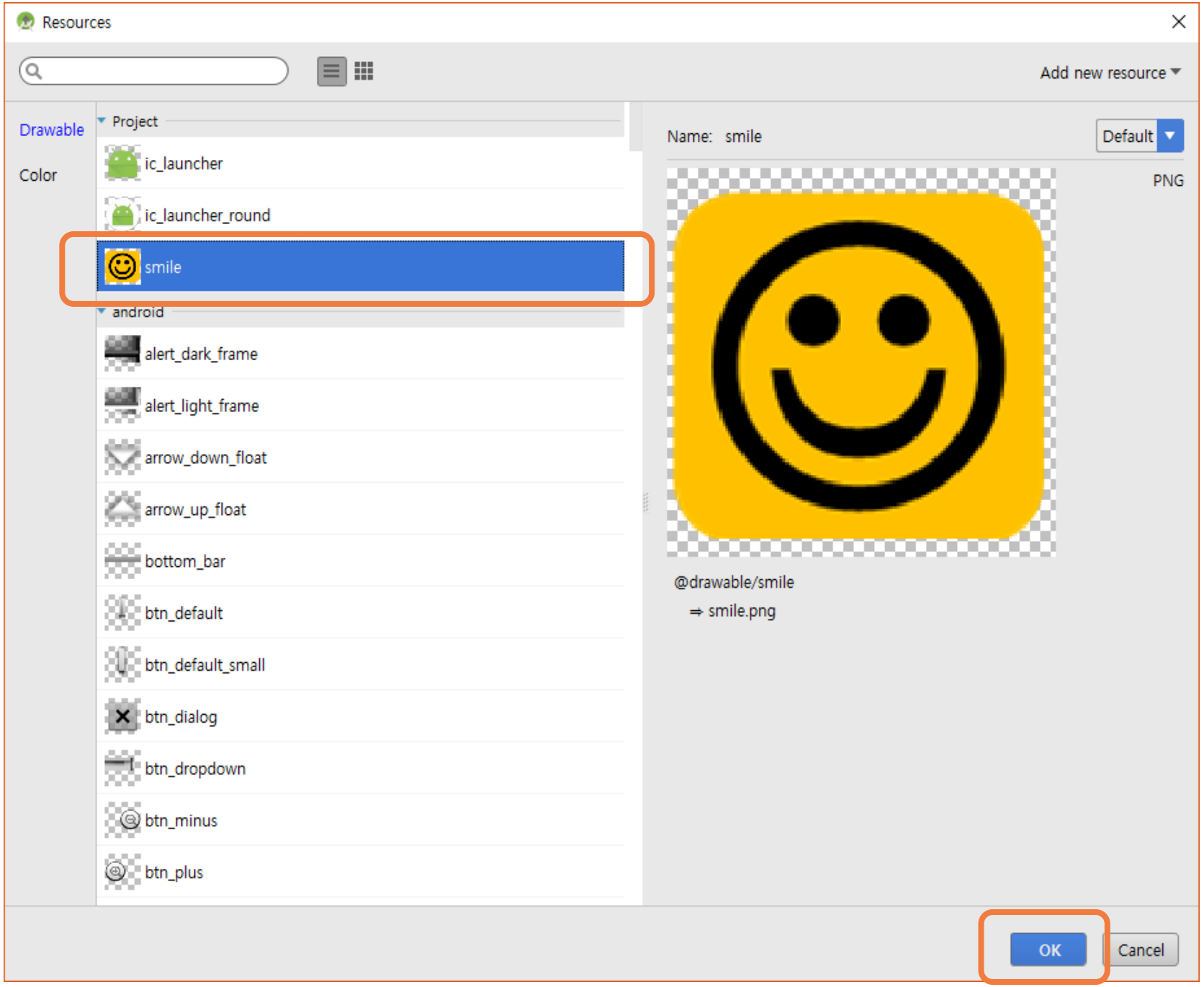
2.3 이미지 리소스 – res/drawable 에 smile.png 추가



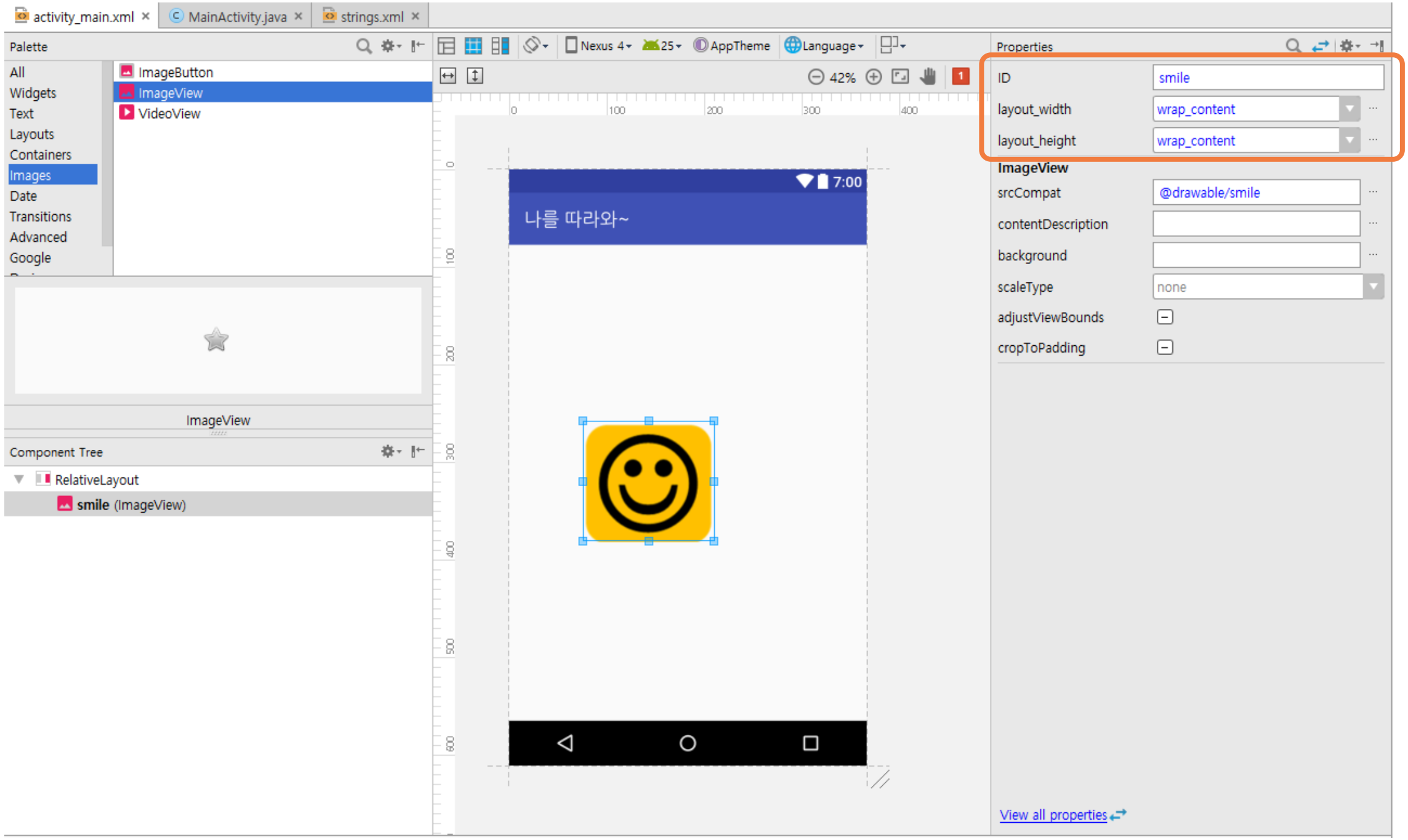
smile.png

2.4 화면 설계


activity_main.xml
MainActivity.java
strings.xml
Palette
All
Widgets
Text
Layouts
Containers
Images
Date
Transitions
Advanced
Google
ImageView
ImageView
Component Tree
RelativeLayout
Nexus 4
25
AppTheme
Language
42%
7:00
나를 따라와~



- Drawable 항목의 smile을 선택


activity_main.xml
MainActivity.java
strings.xml
Palette
All
Widgets
Text
Layouts
Containers
Images
Date
Transitions
Advanced
Google
ImageButton
ImageView
VideoView
ImageView
Component Tree
RelativeLayout
smile (ImageView)
Nexus 4
25
AppTheme
Language
42%
7:00
나를 따라와~
Properties
ID
smile
layout_width
wrap_content
layout_height
wrap_content
ImageView
srcCompat
@drawable/smile
contentDescription
background
scaleType
none
adjustViewBounds
cropToPadding
View all properties

2.5 Activity 제어(MainActivity.java)

- 화면을 전체화면 크기로 만들기 위한 액티비티 상속 클래스 변경

activity_main.xml × | MainActivity.java × | strings.xml ×

```
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 × | MainActivity.java × | strings.xml ×

```
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 상태 바와 앱 바가 나타나지 않는 full screen화면

- ImageView의 src 속성 변경

activity_main.xml × | dimens.xml × | MainActivity.java ×

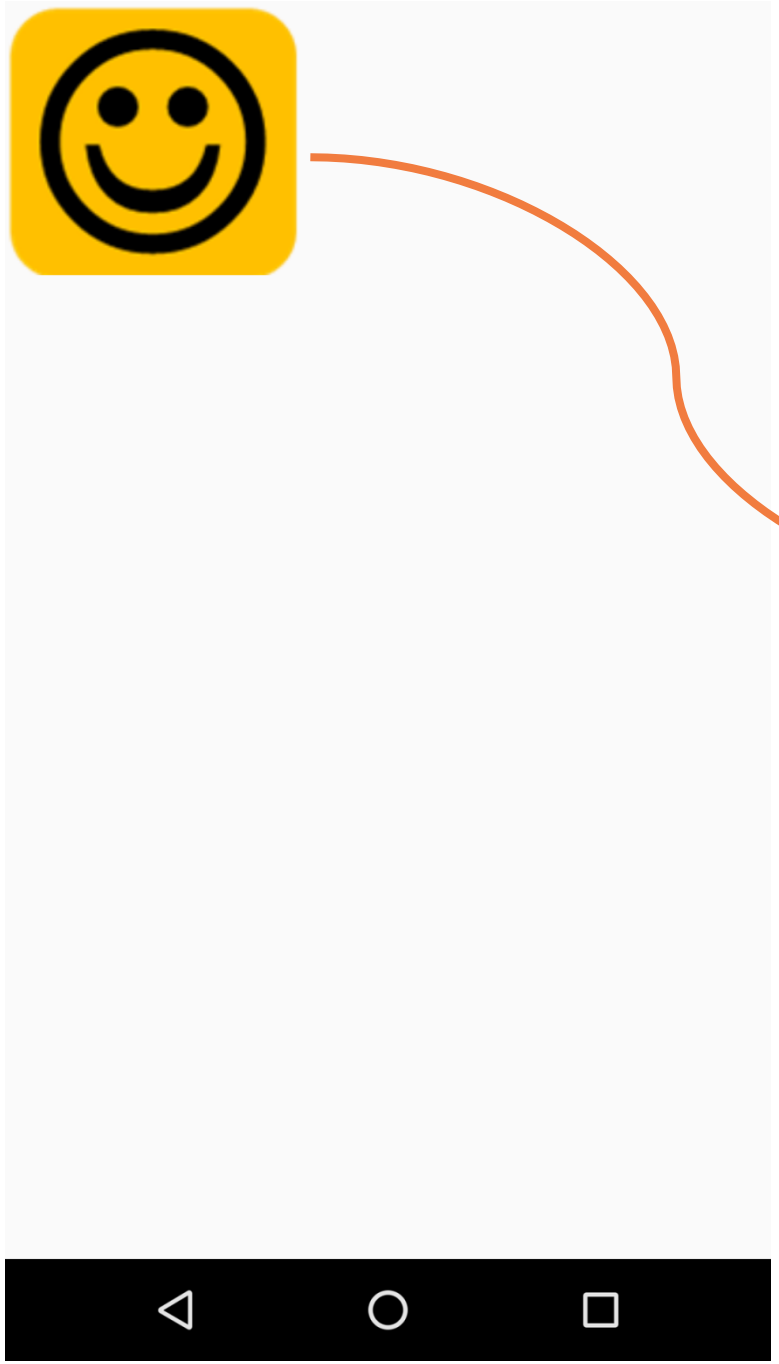
RelativeLayout | ImageView

```
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="com.example.kyungtae.touch.MainActivity">
8
9      <ImageView
10         android:id="@+id/smile"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         app:srcCompat="@drawable/smile" />
14
15 </RelativeLayout>
16
```

제거

activity_main.xml × | dimens.xml × | MainActivity.java ×

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context="com.example.kyungtae.touch.MainActivity">
7
8      <ImageView
9          android:id="@+id/smile"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:src="@drawable/smile" />
13
14 </RelativeLayout>
15
```



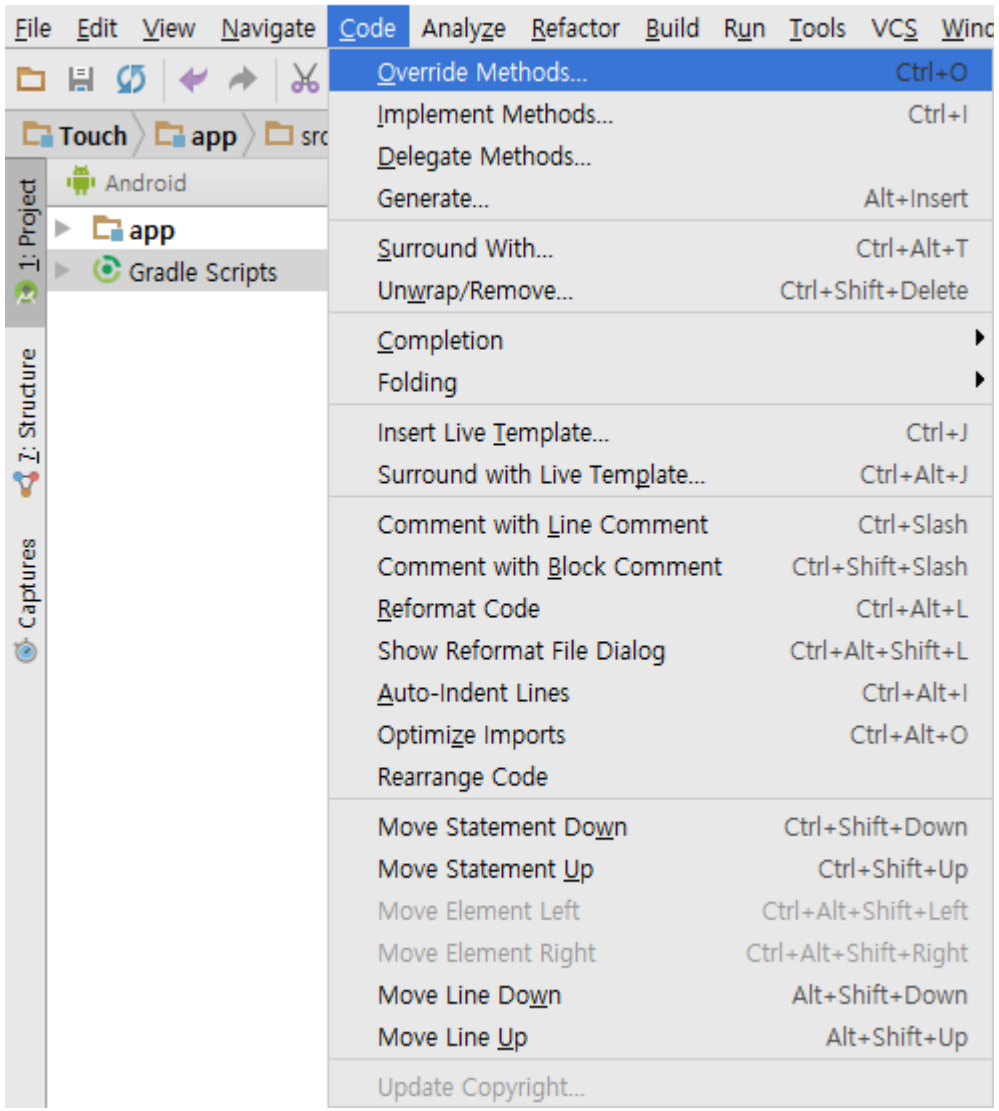
좌측 상단에 이미지가 위치

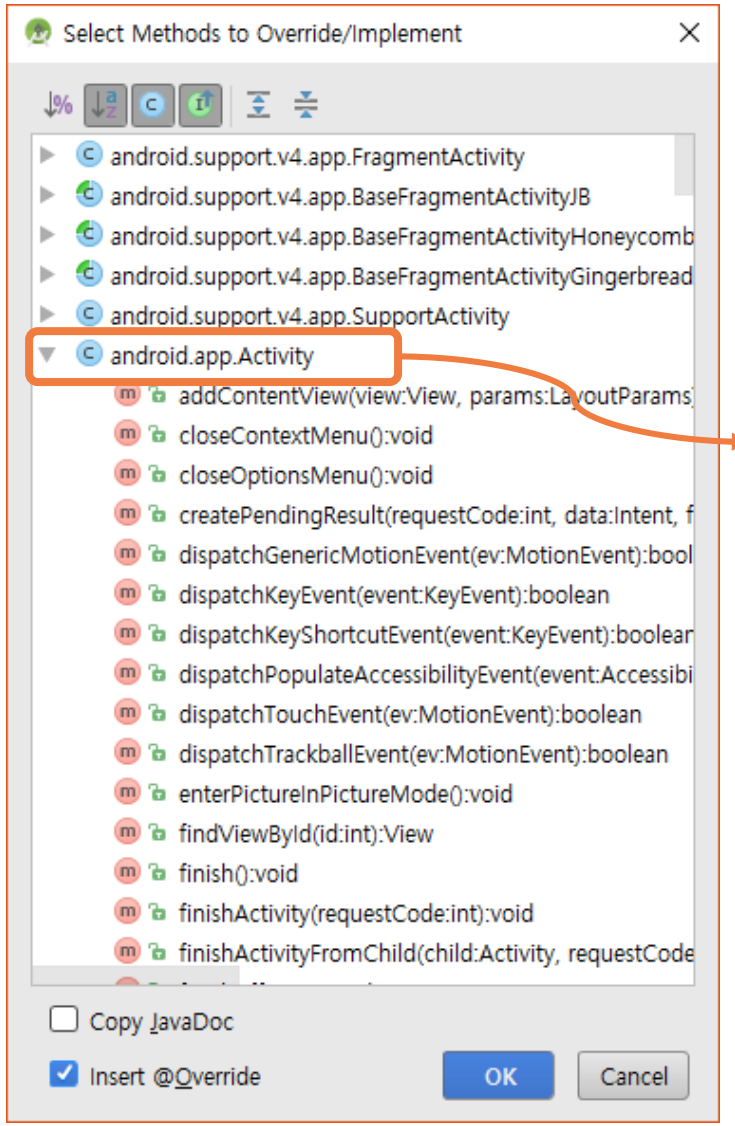
activity_main.xml × | dimens.xml × | MainActivity.java ×

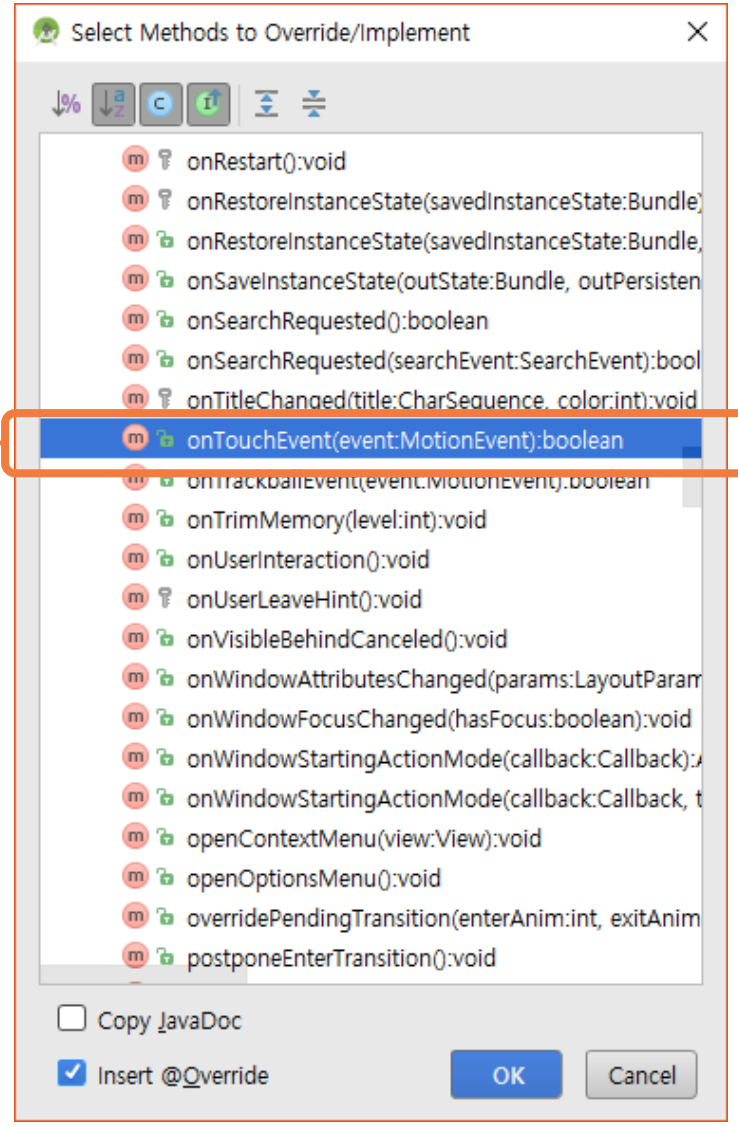
MainActivity

```java
package com.example.kyungtae.touch;

import android.animation.ObjectAnimator;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.WindowManager;
import android.widget.ImageView;

public class MainActivity extends FragmentActivity {

    ImageView ivSmile;
    float previousX = 0;
    float previousY = 0;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // 화면을 Full Screen 모드로 만듦
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);
        // 레이아웃을 액티비티에 출력
        setContentView(R.layout.activity_main);
        // ID가 'smile'인 이미지를 인식
        ivSmile = (ImageView) findViewById(R.id.smile);
    }

```

onTouchEvent() 메소드 재정의(Override)







- 추가된 onTouchEvent() 매소드

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

➔ 수정은 다음 페이지

```
30
31      @Override
32      public boolean onTouchEvent(MotionEvent event) {
33          switch (event.getAction()){ // 터치 유형에 따른 실행
34              case MotionEvent.ACTION_DOWN:   //  누르기 시작하는 상태일 때
35                  break;
36
37              case MotionEvent.ACTION_MOVE: // 화면을 누른 상태에서 움직일 때, 드래그
38                  // 터치 지점의 위치
39                  int touch_x = (int) event.getX();
40                  int touch_y = (int) event.getY();
41
42                  // 아이콘 이미지를 터치 지점으로 이동하는 애니메이션 실행
43                  ObjectAnimator smileX = ObjectAnimator.ofFloat(ivSmile, "translationX", previousX, touch_x);
44                  smileX.start();
45                  ObjectAnimator smileY = ObjectAnimator.ofFloat(ivSmile, "translationY", previousY, touch_y);
46                  smileY.start();
47                  // 현재 위치를 이전 위치로 설정
48                  previousX = touch_x;
49                  previousY = touch_y;
50                  break;
51
52              case MotionEvent.ACTION_UP: // 터치 후 손을 떼는 상태일 때
53                  break;
54          }
55
56          return false;
57      }
58  }
59
```

클래스와 속성/메소드

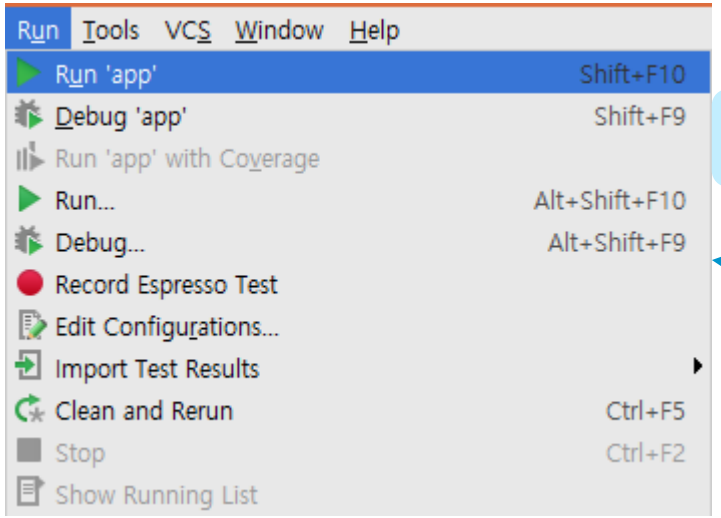
- 클래스

클래스	설명
ObjectAnimator	목표 객체에 대한 애니메이션 특성을 설정

- 메소드

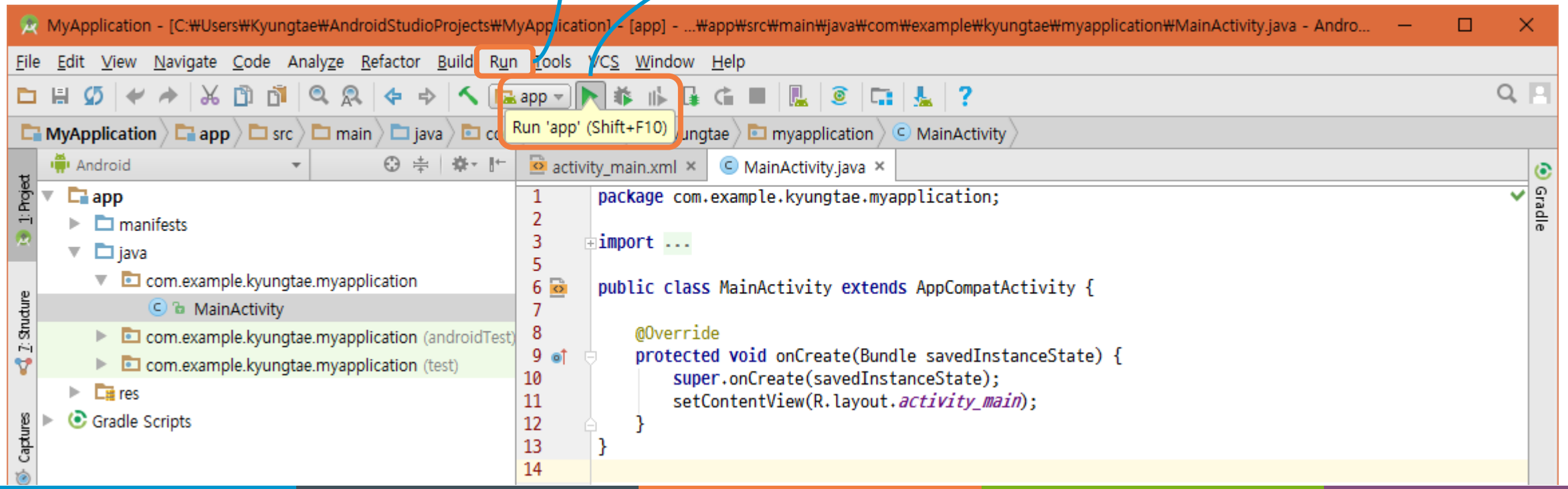
<table>
<tr><th>클래스</th><th>메소드</th><th>설명</th></tr>
<tr><td rowspan="3">ObjectAnimator</td><td>static ObjectAnimator ofFloat(Object target, String propertyName, float ...values)</td><td>values 사이의 애니메이션을 만들고 ObjectAnimator를 반환함
<table>
<tr><th>매개변수</th><th>설명</th></tr>
<tr><td>target</td><td>애니메이션 대상</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></td></tr>
<tr><td>ObjectAnimator setDuration(long duration)</td><td>애니메이션 시간 설정, 밀리초 단위이며, 기본값은 300밀리초로 설정됨</td></tr>
<tr><td>void start()</td><td>애니메이션 시작</td></tr>
</table>

Step 3. 프로젝트 실행

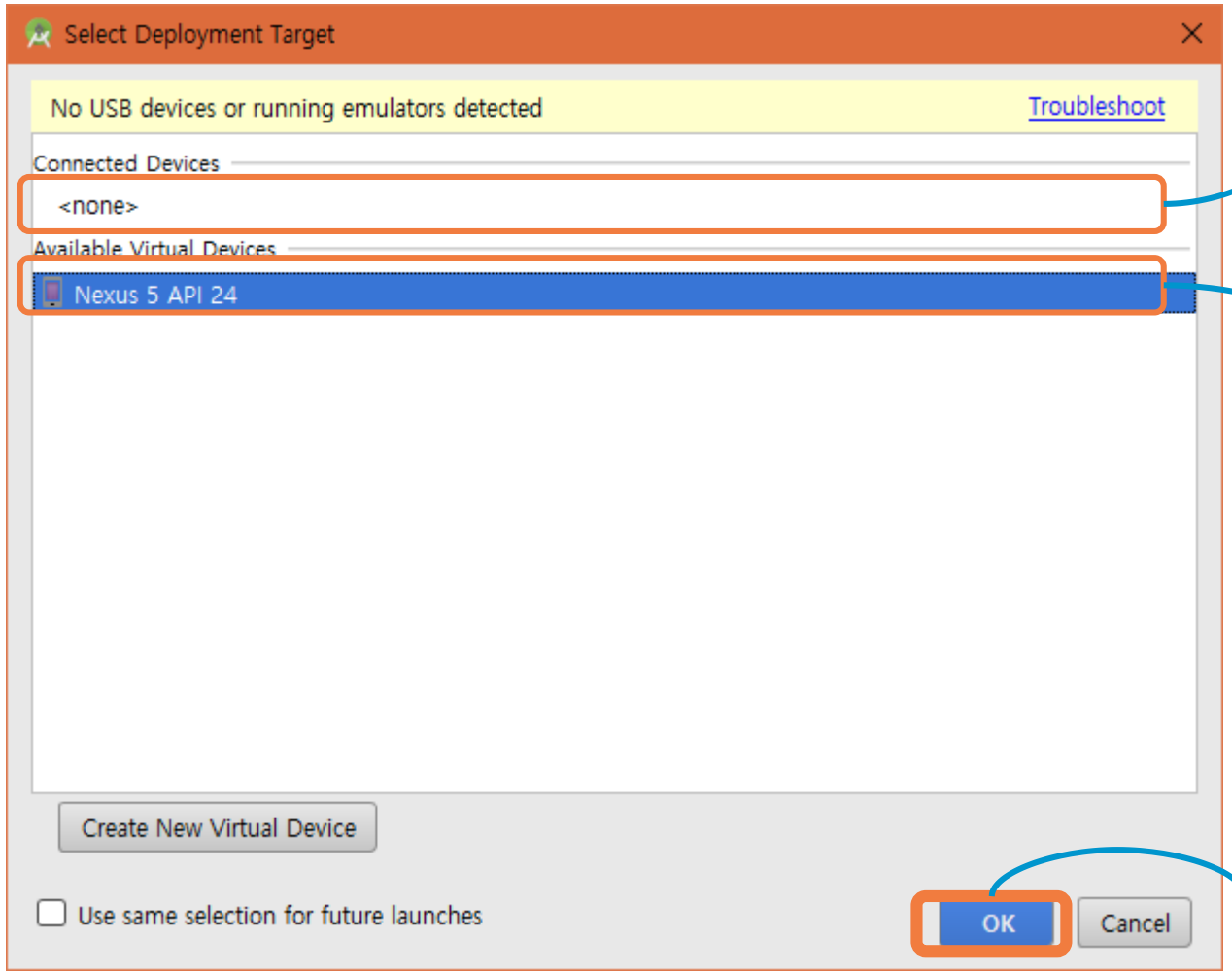


Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기



데이터 케이블로 연결된 스마트폰

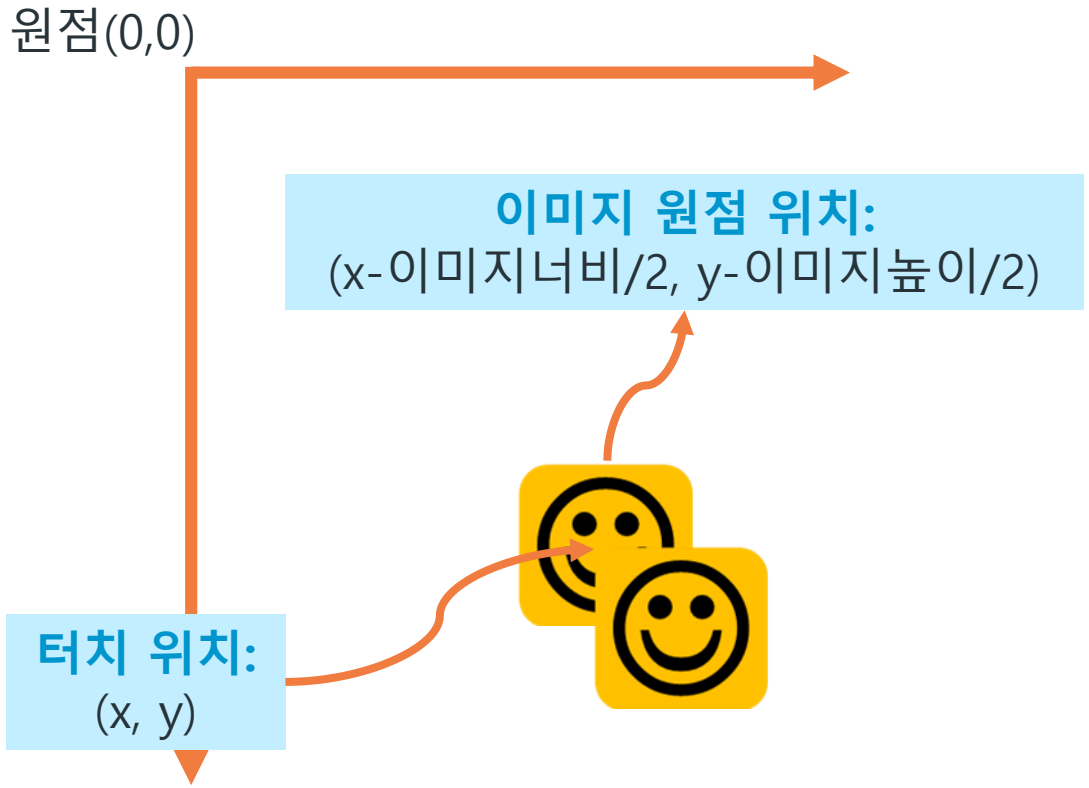
AVD

스마트폰 또는 AVD를 선택하고 'OK' 버튼을 클릭

터치한 위치에 아이콘 이미지 위치시키기

- 이미지의 원점과 터치 위치 사이의 오차처리

```
이미지 x = x-이미지 너비/2;
이미지 y = y-이미지 폭/2;
```



activity_main.xml × | dimens.xml × | MainActivity.java ×

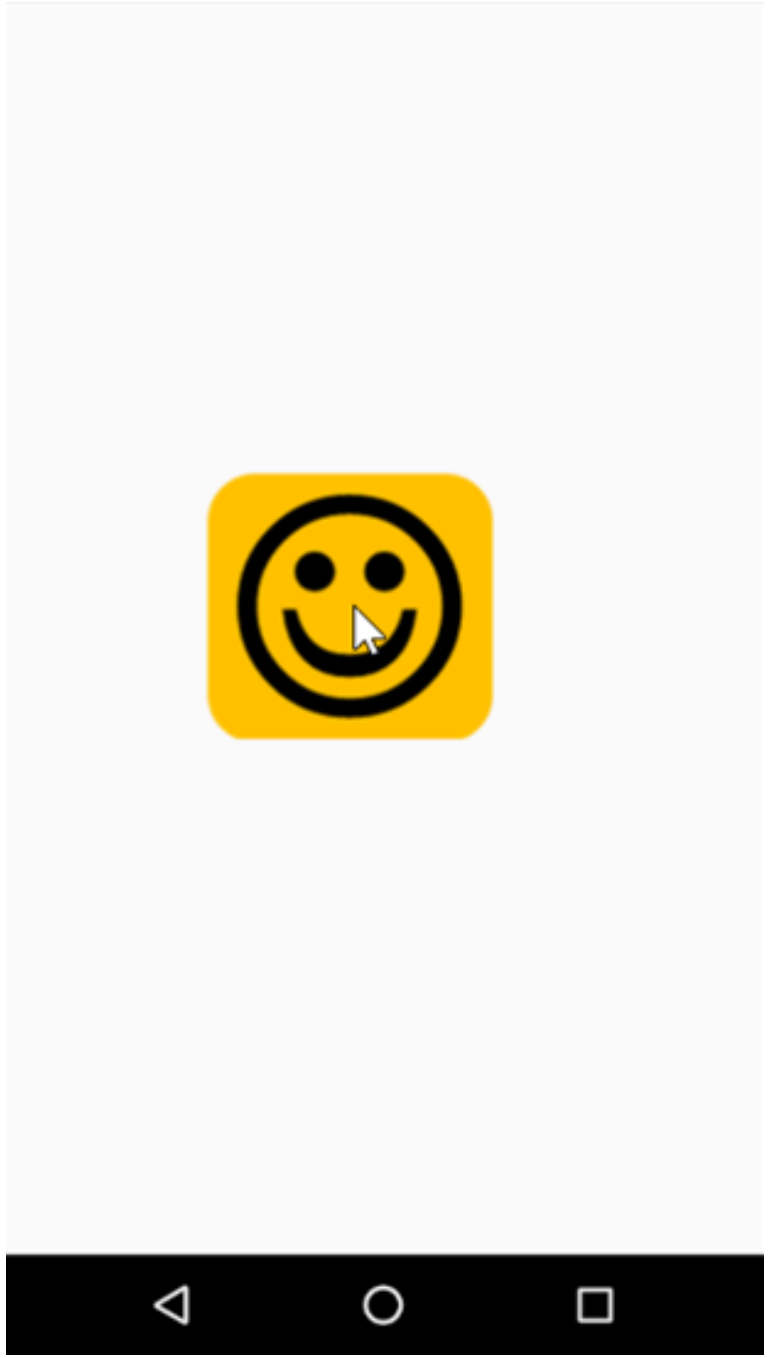
MainActivity | onCreate()

```java
package com.example.kyungtae.touch;

import android.animation.ObjectAnimator;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.WindowManager;
import android.widget.ImageView;

import static android.R.attr.x;

public class MainActivity extends FragmentActivity {

    ImageView ivSmile;
    float previousX = 0;
    float previousY = 0;

    float ivWidth = 0;
    float ivHeight = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // 화면을 Full Screen 모드로 만듦
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);
        // 레이아웃을 액티비티에 출력
        setContentView(R.layout.activity_main);
        // ID가 'smile'인 이미지를 인식
        ivSmile = (ImageView) findViewById(R.id.smile);
        // 아이콘 이미지의 가로와 세로 크기 얻기
        ivSmile.measure(ivSmile.getMeasuredWidth(), ivSmile.getMeasuredHeight());
        ivWidth = ivSmile.getMeasuredWidth();
        ivHeight = ivSmile.getMeasuredHeight();
    }
```

```java

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        switch (event.getAction()){ // 터치 유형에 따른 실행
            case MotionEvent.ACTION_DOWN:   //  누르기 시작하는 상태일 때
                break;

            case MotionEvent.ACTION_MOVE: // 화면을 누른 상태에서 움직일 때, 드래그
                // 터치 지점의 위치
                int touch_x = (int) event.getX();
                int touch_y = (int) event.getY();

                // 아이콘 이미지를 터치 지점으로 이동하는 애니메이션 실행
                ObjectAnimator smileX = ObjectAnimator.ofFloat(ivSmile, "translationX", previousX, touch_x - ivWidth / 2);
                smileX.start();
                ObjectAnimator smileY = ObjectAnimator.ofFloat(ivSmile, "translationY", previousY, touch_y - ivHeight / 2);
                smileY.start();
                // 현재 위치를 이전 위치로 설정
                previousX = touch_x - ivWidth / 2;
                previousY = touch_y - ivHeight / 2;
                break;

            case MotionEvent.ACTION_UP: // 터치 후 손을 떼는 상태일 때
                break;
        }

        return false;
    }
}

```

- 실행 결과

Outputs



Question & Answer