

REPORT



학과		컴퓨터공학과
학년		4 학년
학번		2017076063, 2017076064, 20176065 2017037061, 2016038027
이름		박원진, 김여름, 차서은, 박재성, 왕소림
과목명		오픈소스전문프로젝트
담당교수		박 수창
제출일자		2020-03-25



충북대학교

목차

1.조별로 스마트폰에 내재된 센서
& 새로운 서비스에 대한 고찰

2. Hello Android 코드와 주석, 상세 과정

3. Slide 68의 '직접 풀어보기'
(전화걸기, 갤러리 열기 수행 시 추가 점수)

01. 조별로 스마트폰에 내재된 센서들에 대해 조사하고 이들의 복합적 정보를 기반으로 제공할 수 있는 새로운 서비스에 대해 고찰 보고서를 작성하시오. (박 원진, 왕 소림)

01-1 스마트폰에 내재된 센서

1. 가속 센서

가속 센서는 스마트폰의 움직임을 감지하고 보통 자이로 센서와 함께 사용합니다. X, Y, Z로 좌표를 만들고 이 좌표의 움직이는 속도를 측정할 때 사용하는 모션 센서 중 하나입니다. 가속 센서는 주로 움직이는 물체 또는 스마트폰의 속도를 측정합니다.

2. 자이로 센서 (Gyro Sensor)

자이로 센서는 가속 센서와 함께 모션 센서와 함께 모션 센서의 대표적인 센서입니다. X, Y, Z 좌표에서 움직이는 방향을 측정할 때 사용합니다. 자이로 센서를 이용하면 스마트폰을 바닥에 놓고 있다 집어 올렸을 때 특정한 앱을 실행하거나 기능을 동작시킬 수 있습니다.

3. 근접 센서

근접 센서는 어떤 물체가 센서에 근접했는지 알 수 있게 해줍니다. 보통 스마트폰의 앞면에 있으며 통화용 스피커 옆에 있는 것이 일반적입니다. 이 센서가 있어야 통화 중일 때 스마트폰 화면이 자동으로 꺼지는 기능을 구현할 수 있습니다.

4. RGB 센서

RGB 센서는 주변 빛의 색 농도를 검출하는 기능을 합니다. RGB 센서가 있는 스마트폰은 주변 빛 농도에 따라 디스플레이 색을 보정할 수 있는데요.

예를 들어 노란색 등이 있는 곳에서 스마트폰을 사용하면 이에 대응해서 노란색을 낮추고 파란색과 녹색을 더 밝게 해서 자연스러운 화면 색을 구현하게 됩니다.

5. 밝기 센서(Light Sensor)

조도 센서라고도 하며 주변 빛의 밝기를 감지합니다. 이 센서는 디스플레이의 밝기를 자동으로 조절할 때 사용하며, 보통 근접 센서 옆에 있습니다.

주변의 빛이 밝으면 스마트폰 화면을 더 밝게 해주고, 어두운 곳에서는 화면을 어둡게 해서 눈의 피로를 줄이고 가독성을 높여주는 중요한 역할을 합니다.

6. 홀 센서(Hall Sensor)

자기장의 세기를 감지할 때 사용하는 센서입니다. 스마트폰에서는 홀 센서를 이용해 플립 커버의 닫힘 유무를 확인할 때 사용합니다.

7. 모션 센서

물체의 움직임을 인식하는 센서입니다. 모션 인식 센서는 보통 하나의 센서가 아닌 여러 가지의 센서가 복합된 것을 말하며, 지자기 센서, 가속 센서, 기압계 등 움직임이나 위치를 측정할 때 사용합니다.

모션 센서로는 많은 일들을 할 수 있는데요, 스마트폰이 꺼져 있을 때 자동으로 켜진다가나, 반대로 화면을 보고 있을 때는 화면을 끄지 않고 계속 보여주는 기능 등이 여기에 속합니다. 또한, 음악 재생 중 스마트폰을 흔들면 다음 곡으로 이동하는 것도 모션 센서가 필요합니다.

8. 온도/습도 센서

온도/습도 센서는 단말기 주변의 온도와 습도를 측정하여 보여줍니다. 현재는 온도계와 습도계로만 활용되고 있지만, 나중에는 이를 활용한 다양한 기능이 추가될 것으로 보입니다. 대중적으로 사용하고 있는 센서는 아닙니다.

9. 기압계(barometer)

기압계는 말 그대로 공기의 압력을 감지할 때 사용합니다. 일반적인 환경에서는 사용하지 않고 헬스 기능에서 주로 사용하는데요. 고도를 측정할 수 있기 때문에 경사나 내리막길도 알 수 있어 더욱 정확한 운동량 체크를 할 수 있습니다.

10. 지자기 센서

지자기 센서는 지구의 자기장을 탐지해 방위를 알 수 있는 센서입니다. 이 센서가 있으면 나침반 앱을 이용해 방위를 정확하게 측정할 수 있습니다.

11. 심장 박동 센서 (Heart beat Sensor)

심장 박동을 측정하기 위해 사용하는 센서입니다. 갤럭시S5에 처음으로 사용되었고 이후 갤럭시 기어2 시리즈와 기어 핏에도 탑재되었습니다. 주로 운동과 피트니스용으로 사용됩니다.

12. 지문 인식 센서

지문 인식 센서는 말 그대로 사람의 고유한 지문 패턴을 읽을 수 있는 센서입니다. 스마트폰 보안을 위해 사용되죠. 아이폰5s에 처음으로 사용되었고 이후 갤럭시S5와 갤럭시S5 광대역 LTE-A 모델에 탑재됐습니다.

13. 얼굴 인식 센서

얼굴 인식 센서는 말 그대로 사람의 고유한 이목구비의 패턴을 파악하여 읽을 수 있는 센서입니다. 스마트폰 보안을 위해 많이 사용되는 센서 중 하나입니다.

01-1 스마트폰에서 가능한 새로운 서비스

첨단 기술이 점점 발달하면서 스마트폰에 대한 활용도가 점점 높아지고 있는데요. 예를 들면 음성 인식 센서를 사용하는 아이폰의 시리를 통해 운전을 하면서도 휴대폰을 만지지 않더라도 문자를 보낸다거나 메일을 보낼 수도 있고 전화를 할 수도 있습니다.

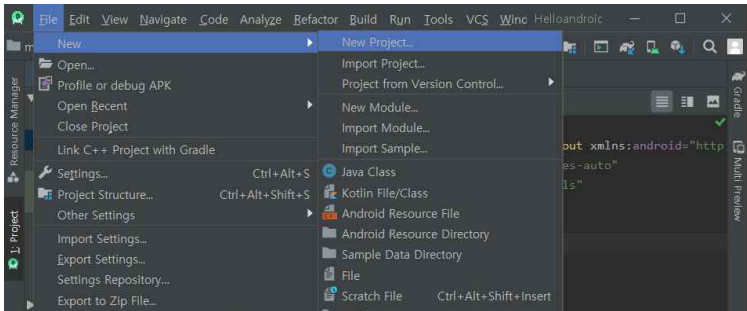
스마트폰은 우리 생활에서 빼놓을 수 없는 중요한 it 기기가 되어 버렸는데요. 위에서 언급하였던 일상생활을 편리하게 해주는 기능도 많지만 최근에는 스마트폰을 사용해 재미있는 기능도 많이 찾아볼 수 있습니다. 사진을 못 찍는 사람들을 위해서 위치를 잡아주는 카메라 어플, 토익 및 자격증 문제를 재미있게 푸는 어플 등을 예로 들 수 있습니다.

이런 복합적인 기능 및 센서를 포함하여 미래에는 집에서 간편하게 운동할 수 있는 어플이 나올 거 같습니다. 카메라를 통해 모션을 감지하여 자세를 교정해주고, 보다 재미있는 방식으로 집에서 운동할 수 있는 어플이 나올 거 같습니다.

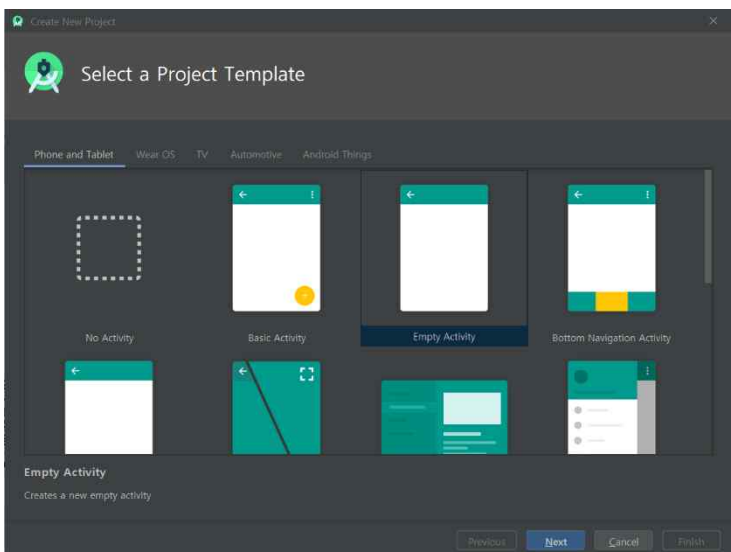
02. 조별로 첨부된 주교제 내용을 따라 Hello Android를 작성하고 코드와 주석, 상세 과정들을 캡처하고 단계별 설명하시오. 섹션 별 또는 설명 별 보고서 상 담당 조원 이름을 함께 적으시오. (차 서은)

[Hello Android 프로그램]

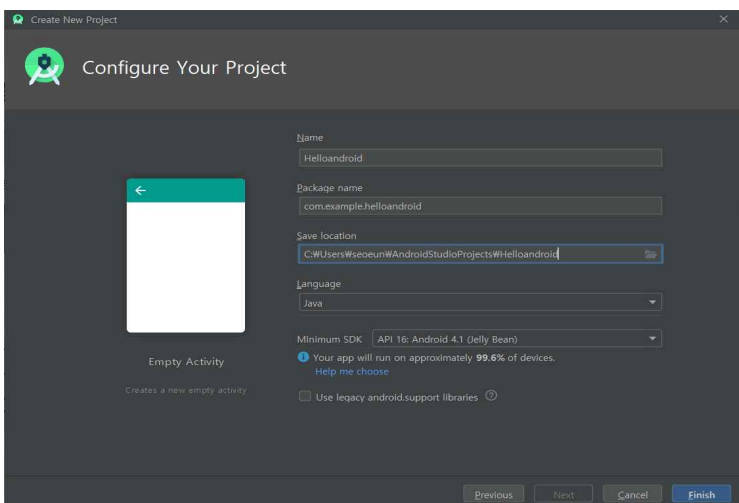
1. 새로운 프로젝트 생성하기 (File -> New -> New Project)



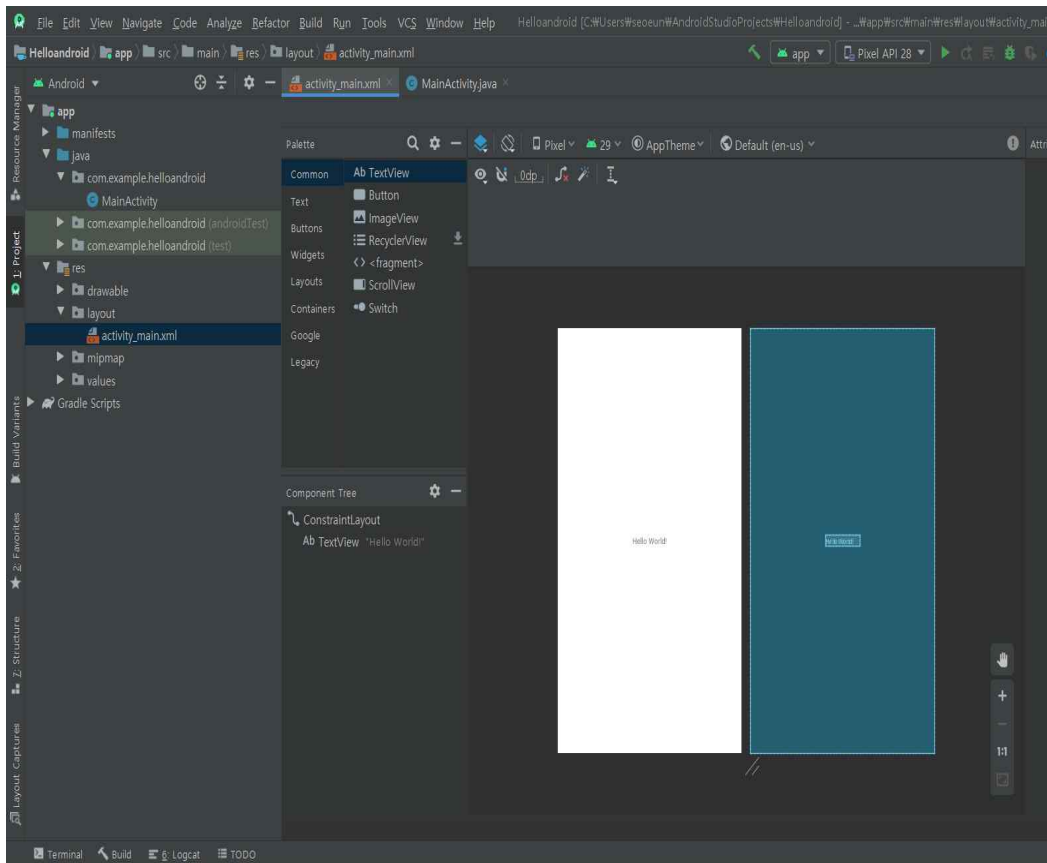
2. 이 화면이 뜨면 “Empty Activity”클릭하기



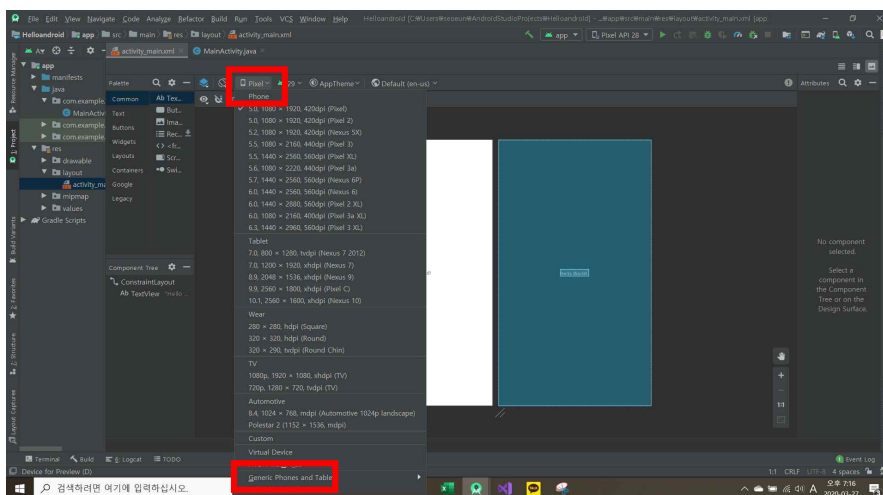
3. finish 클릭하기



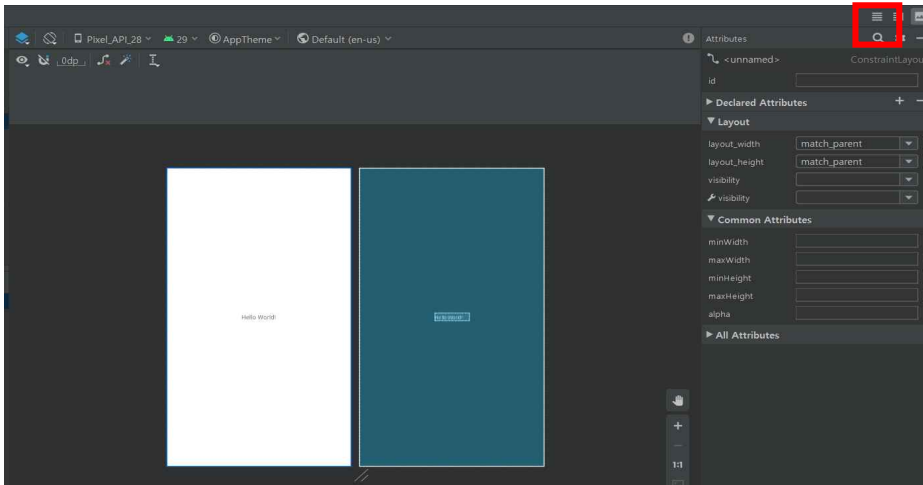
4. res -> layout -> activity_main.xml 더블클릭



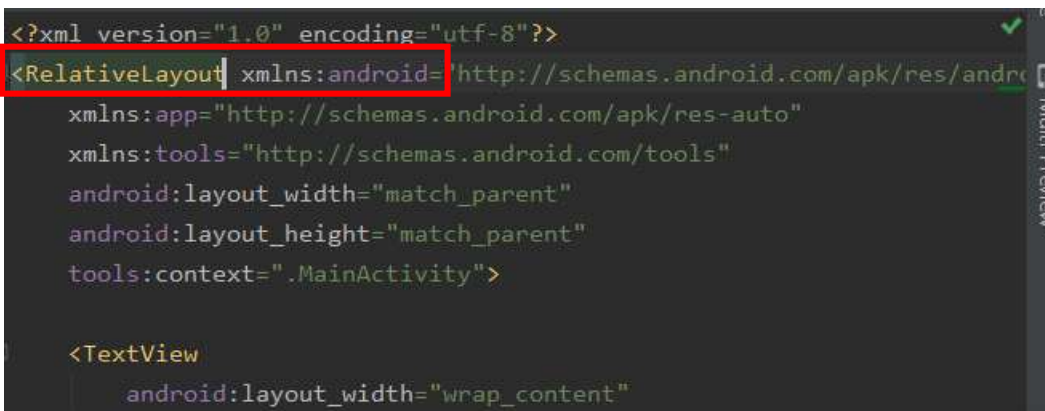
5. pixel 표시부분 클릭 후 AVD: PixelAPI_28 클릭하기



6. 오른쪽상단의 표시한 네모 클릭



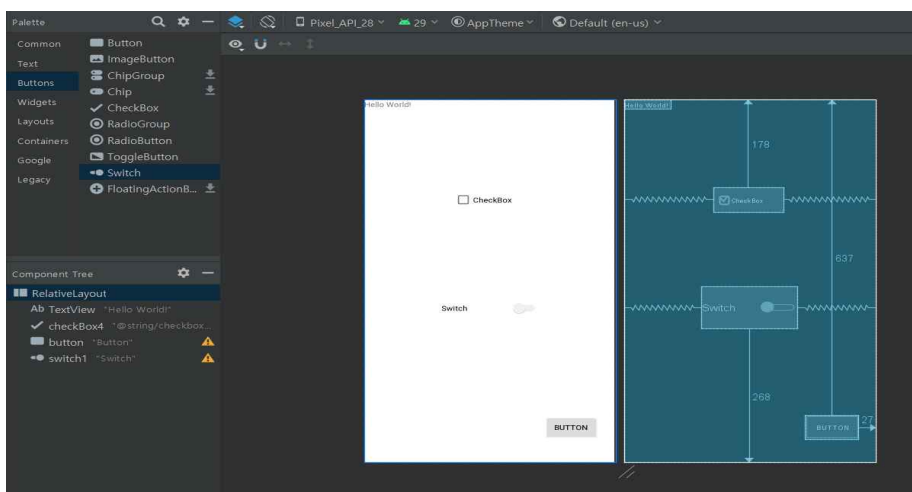
7. Android.constraintlayout.widget.ConstraintLayout을 -> RelativeLayout으로 변경



코드 설명(“RelativeLayout”):

RelativeLayout은 부모(Parent) View 또는 자식(Child) View의 상대적 위치 관계를 정의하여 UI를 배치하는 Layout 이다. 주의할 점은 A를 B에 배치하고 싶다면, B를 먼저 기술한 뒤, A를 기술해야 한다. 위의 코드에서 “android:layout_width=”와 “android:layout_height=”는 뷰와 레이아웃에서 공통으로 쓰이는 속성이다. 이것의 의미는 뷰의 폭과 높이를 설정해주는 것이다. “”안에 있는 “match_parent”의 의미는 부모 컨테이너의 남아있는 모든 여유 공간을 채운다는 의미이다. (추가적으로 : 뷰 안에 들어있는 내용물의 크기에 맞게 뷰의 크기가 결정하고, : 부모 컨테이너의 남아있는 모든 여유 공간을 채운다.)

8. Buttons -> CheckBox에서 원하는 button 넣고, 위치 조정 및 크기 조절하기



9. 다시 코드 화면으로 들어가서 아래 표시 부분 삭제하기

```
android:layout_marginLeft="147dp"
android:layout_marginTop="178dp"
android:text="@string/checkbox1" />

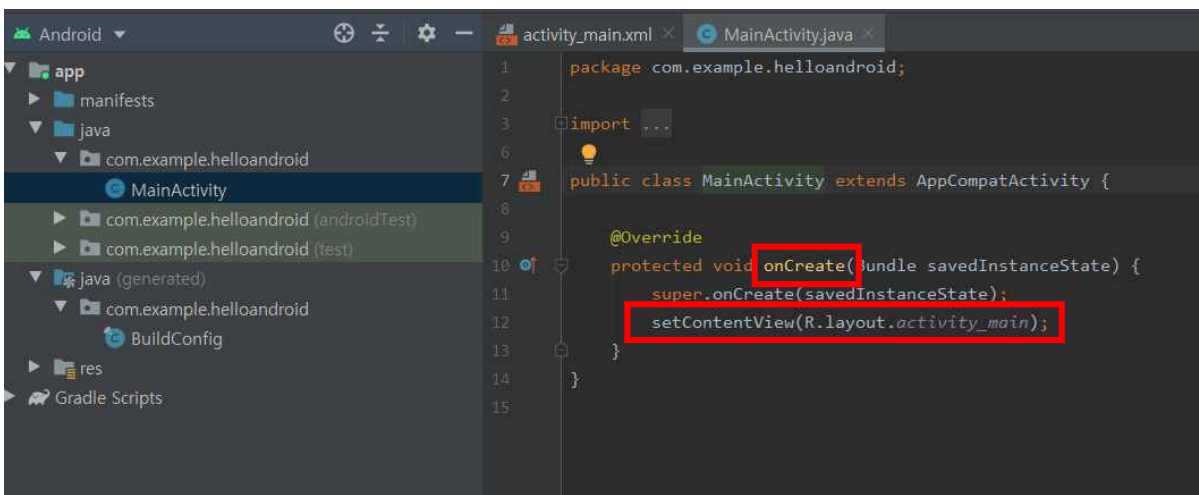
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginTop="637dp"
    android:layout_marginEnd="27dp"
    android:layout_marginRight="27dp"
    android:text="Button" />

<Switch
    android:id="@+id/switch1"
    android:layout_width="157dp"
```

코드 해설(<button>)

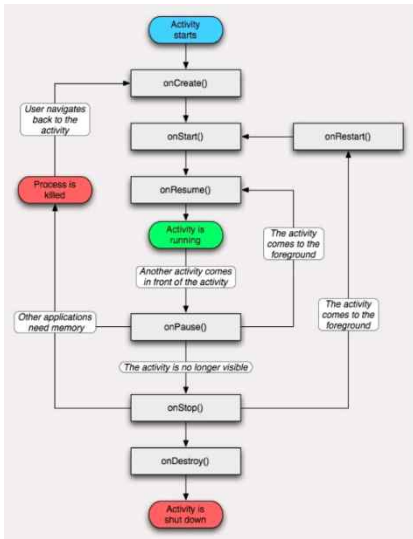
Button은 사용자가 화면을 터치했을 때 발생하는 클릭 이벤트를 처리하는 기능을 가진, 텍스트 또는 아이콘으로 구성된 View 위젯이다. TextView와 마찬가지로 안드로이드 UI를 구성할 때 가장 많이 사용되는 위젯 중 하나이다. 그리고 Button에서 발생할 수 있는 클릭 이벤트는 부모 클래스인 View 클래스에 이미 정의되어 있다. 또한 클릭 이벤트를 위한 리스너도 Java 인터페이스의 형태로 정의되어 있으며, 이벤트 발생 시 호출될 함수의 형식(이름, 파라미터, 리턴 값)도 인터페이스 내에 정의되어 있다. 더 나아가 어떤 리스너 객체가 Button의 클릭 이벤트를 처리할 것인지 지정하는 함수까지 만들어져 있다. 아래 코드는 layout인데 width와 height는 너비와 높이를 의미하고, "android:layout_alignParentTop" 그리고 End, Right는 의미는 뷰를 부모의 상단 가장자리 또는 끝에 또는 오른쪽에 붙일 것인지를 여부를 판단하고, margin의 코드에서는 뷰와 부모간의 간격을 지정하는 코드이다. 근처에 형제뷰가 있으면 형제 뷰와의 간격도 마찬가지로 떨어진다.

10. [java] - [com.cookandroid.helloandroid] - [MainActivity] 선택하기 후 체크 부분 코드 확인하기



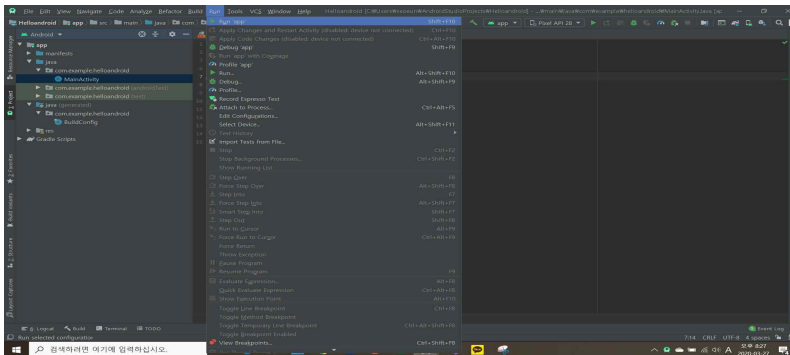
```
1 package com.example.helloandroid;
2
3 import ...
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13     }
14 }
15
```


이코드에서의onCreate()는activity의 생명 주기이다.

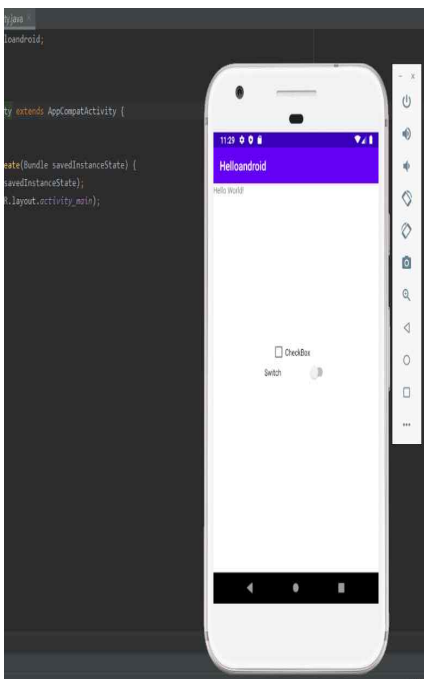


왼쪽 그림을 보면 쉽게 이해할 수 있다. 액티비티가 처음 생성될 때 호출되는 함수이다. 처음 프로젝트를 생성하면 MainActivity에서 자동으로 생성해줄 만큼 액티비티 생명주기에 있어서 중요한 함수이다. 해당 함수에서는 액티비티를 실행하기 위하여 정의된 사용자 UI를 올리기 위해 레이아웃 리소스 파일을 읽어와 객체를 생성하고 각종 초기화 작업을 진행하게 된다. 해당 함수 수 호출 후에는 항상 onStart() 함수가 같이 호출된다. 그리고 setContentView()의 의미는 activity사용자의 눈에 보이는 application이다. 따라서, 화면을 설정해줘야 하는데 이 method가 화면을 설정해 주는 것이다. 그리고 setContentView의 파라미터인 R.layout.main은 android project를 보면 src라는 폴더도 있지만, res라는 폴더도 있다. 이 안에 layout이라는 폴더 안에, 기본적으로 main.xml이란 파일이 있다. 이 파일이 빌드 되면 자동으로 'R'이란 클래스가 생성되고, 그 안에 layout이라는 폴더 때문에, layout이 생성 되고 그 안의 main이라는 내용이 값으로 설정 되어 들어 가는 것이다.

11. 실행 하기([Run 'app'] 클릭하기)

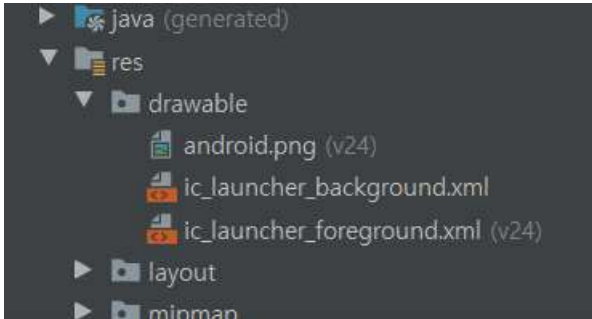


12. AVD가 부팅 후 실행 결과 화면이 나온다.

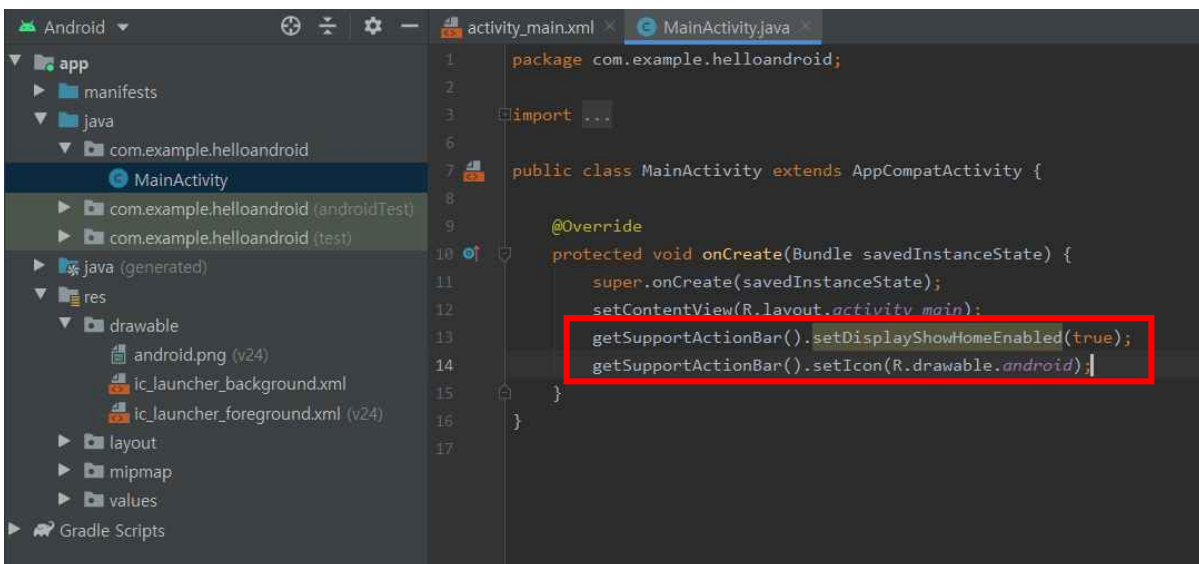


13. AVD 아이콘 출력을 위해

1) 96*96크기의 .png파일을 저장해둔다. 그리고 저장된 파일을 [res]->[drawable]에 드래그하여 파일을 넣는다.



2) Project tree에서 [app]->[java]->[com.example.파일명]으로 들어가 아래의 소스를 추가한다



위의 getSupportActionBar()의 함수는 액티비티의 객체에 대한 참조를 획득하고 나면, 클래스에서 제공하는 함수를 통해 앱바(App Bar)의 표시항목을 설정하거나 앱바(App Bar)가 제공하는 여러 기능을 사용할 수 있다.

3) 선택사항!

Project tree에서 [app]->[manifests]->[AndroidManifest.xml]을 더블클릭하여 들어가

테마 색을 아래와 같이 수정해준다.(11번 줄)

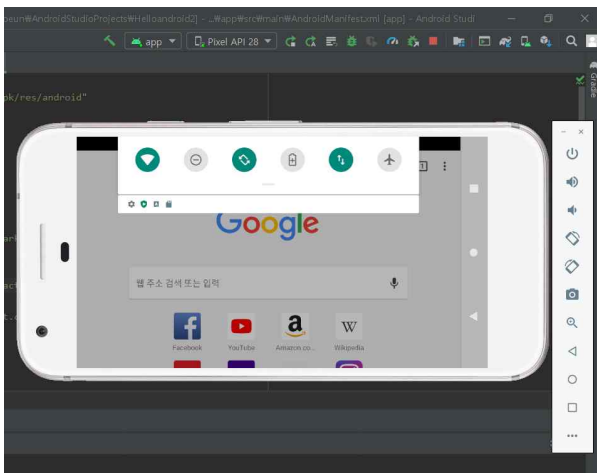
```
android:supportsRtl="true"
android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
<activity android:name=".MainActivity">
```

4) 결과



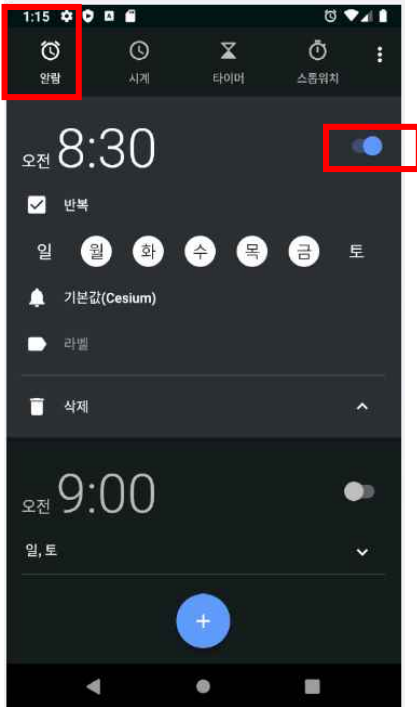
〈AVD 명칭과 사용법〉

1. 화면 전환(자동 회전을 설정하고, [Ctrl]+[←] or [Ctrl]+[→])



2. 알람 추가

- 처음 화면에서 위쪽으로 스와이프하여 [시계] 클릭한다.
- [알람]을 클릭 후 스위치를 클릭해주면 알람이 설정이 된다.



3. 뒤로가기 ◀ 버튼을 누른 후 [카메라]를 클릭한다.



1)셔터를 누르면 아래와 같이

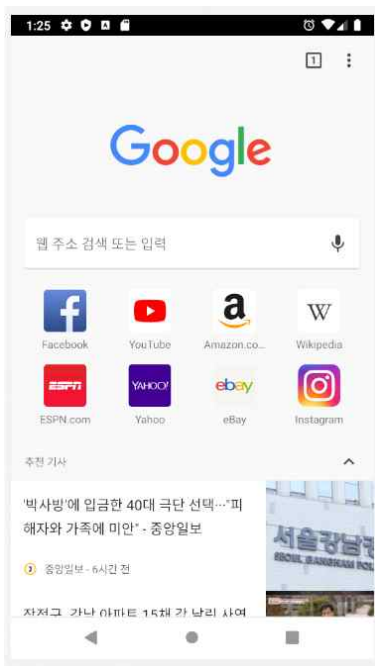
->

2)◀ 클릭 후, [포토]를 클릭
하면 오른쪽과 같은 화면을
볼 수 있다.



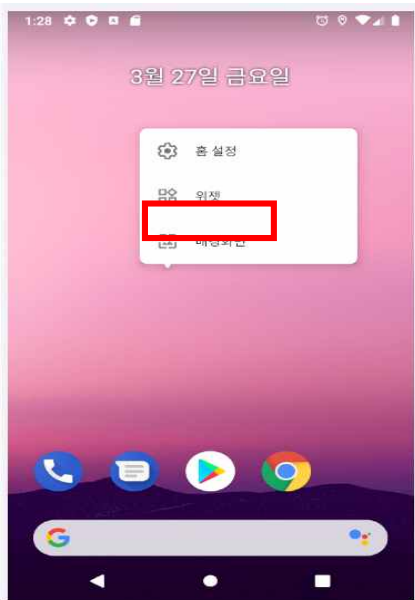
촬영이 된다.

4. 웹브라우저 사용



5. 배경화면 변경

1) 바탕화면을 몇 초 동안 누르면 아래와 같은 화면이 뜬다. 이후 바탕화면 클릭



2) 원하는 배경 클릭.



3) 배경화면 설정 클릭

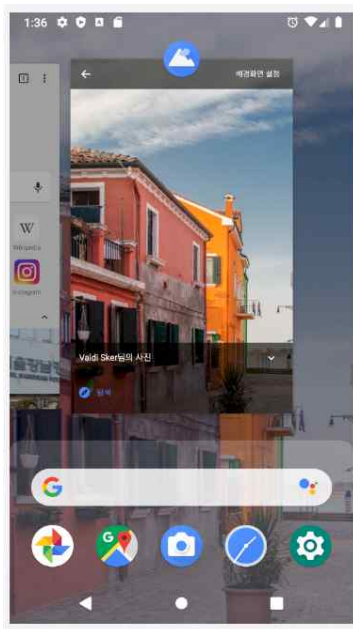


4) 배경화면 변경 완료



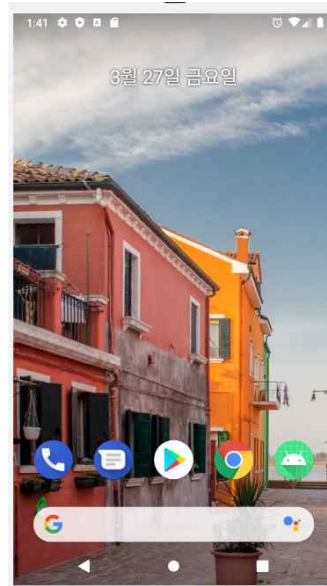
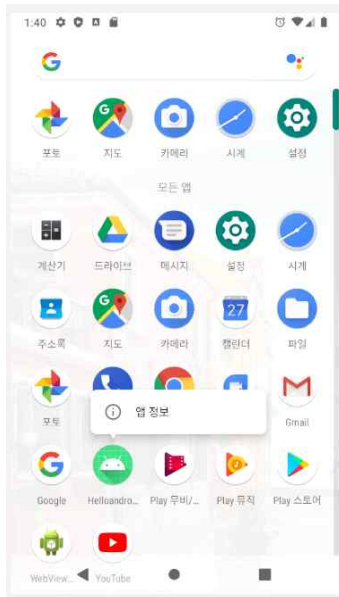
6. 오버뷰

- 아래 화면에서 마우스를 위로 스와이프하면 하나의 실행 중인 앱을 종료할 수 있다.

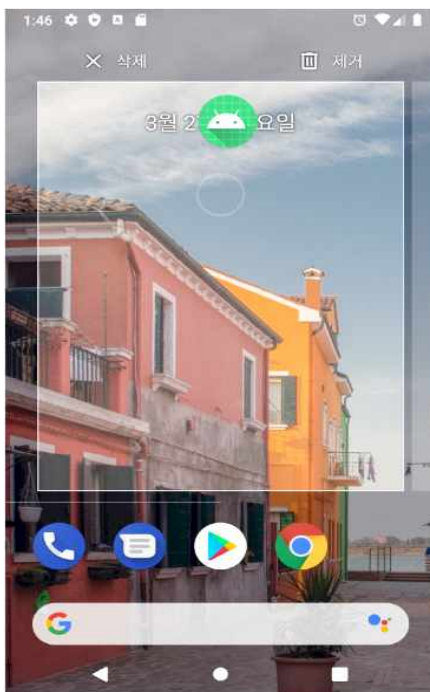


7. 자주 사용하는 애플리케이션 복사

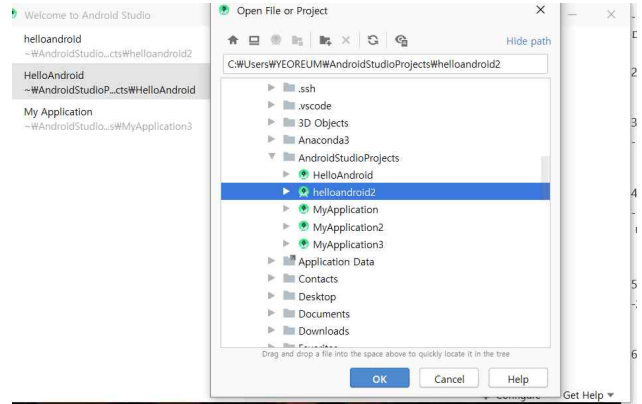
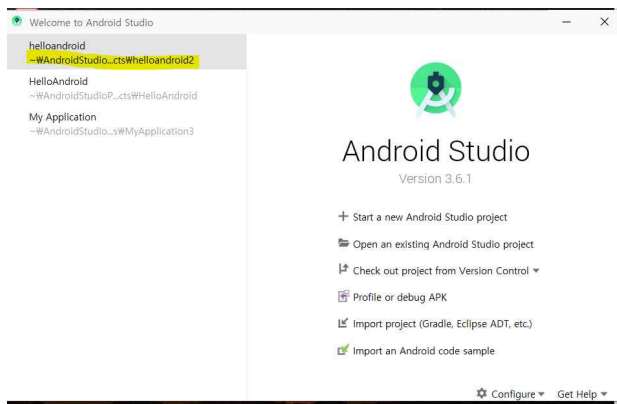
1) 메뉴에서 HelloAndroid 를 몇 초간 클릭하면 2) 계속 클릭되어 있는 상태에서 원하는 아래와 같이 [앱 정보]가 나온다. 위치에 옮긴다.



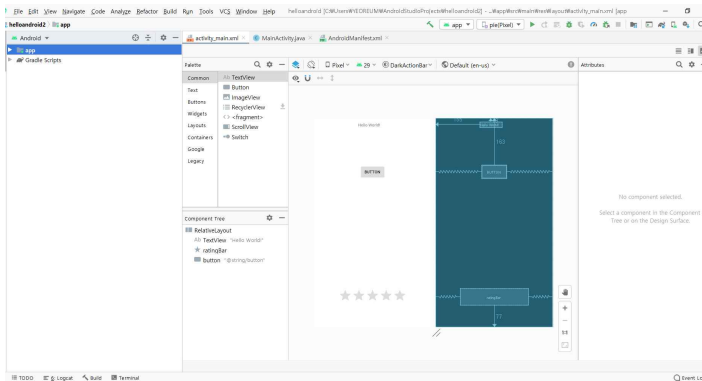
3) 아이콘을 배경화면에서 삭제하기 위해서 [앱 정보가 나오면] 아예 제거하려면 [제거], 화면에서 삭제하려면 [삭제]로 옮긴다.



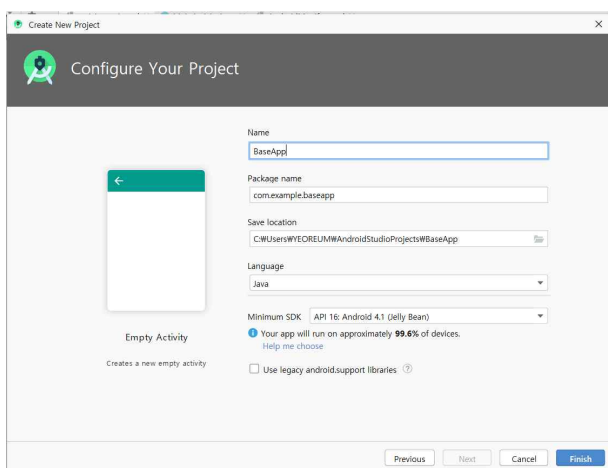
02.2 안드로이드 애플리케이션 작성 [버튼 만들기] (김 여름)



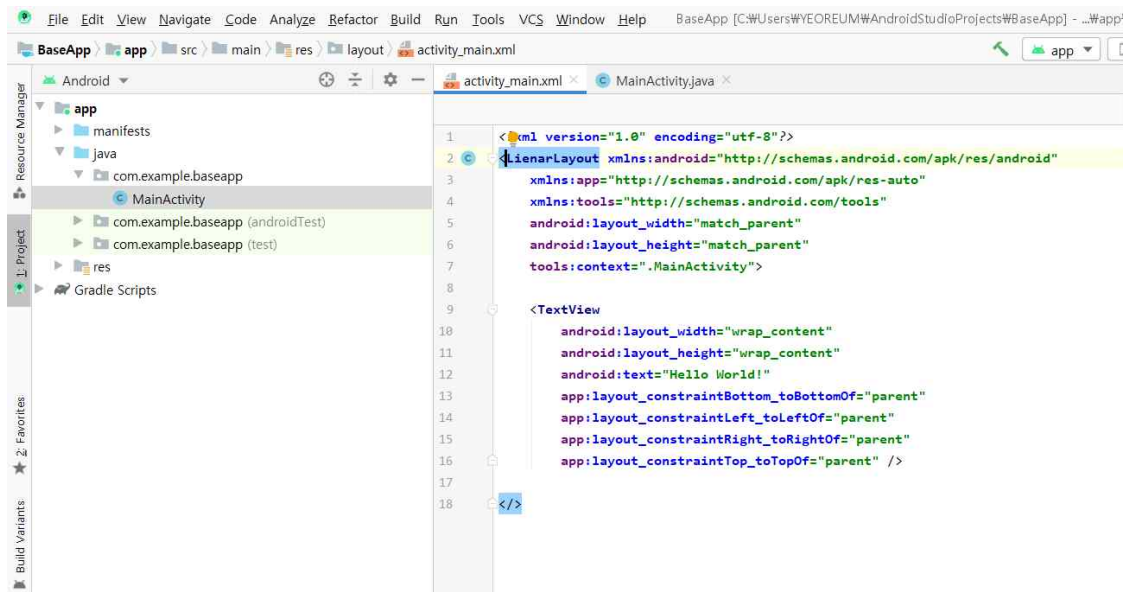
- [file] - [close project]를 통해 프로젝트 닫은 후, 프로젝트 이름을 클릭하면 다시 프로젝트가 열리며, 폴더 검색을 통해 프로젝트를 열 수도 있음.



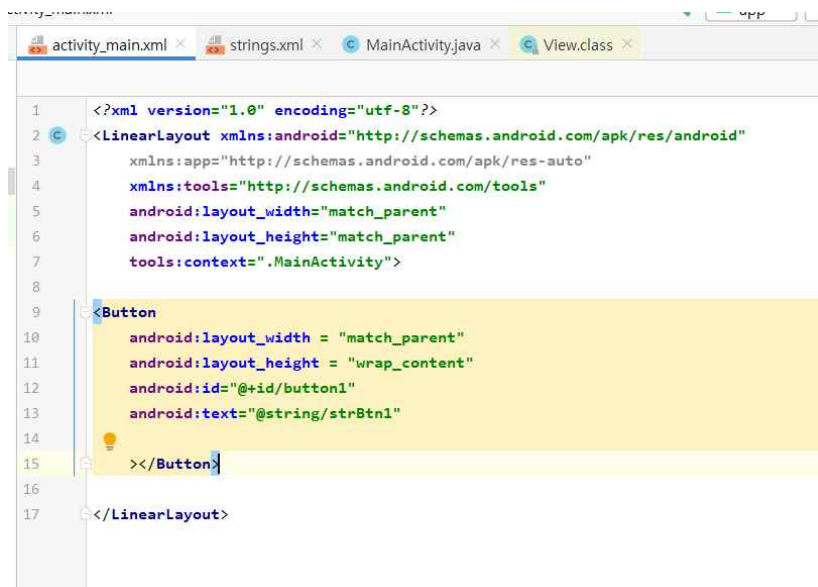
- 이클립스용 프로젝트 가져오기
import project 선택 후, 프로젝트 폴더를 선택하여
이클립스용 프로젝트로 변환한다.



- BaseApp 이라는 새로운 프로젝트 생성



- activity_main.xml에서 LinearLayout으로 코드 변경
- 리니어 레이아웃 : 가장 많이 사용하는 레이아웃으로 박스 모델을 사용한다.
- 한쪽 방향으로 차례로 뷰를 추가하여 화면을 구성하는 방법이다.
- 사용이 간편하고 표시 형태가 직관적이라는 장점을 가졌기 때문이기도 하지만, 무엇보다 사각형 박스 형태의 디스플레이 화면에 UI 요소들을 일렬로 배치할 수 있어서 안정감 있는 화면 구성이 가능하다는 이유로 가장 많이 사용한다.



- 버튼을 넣음

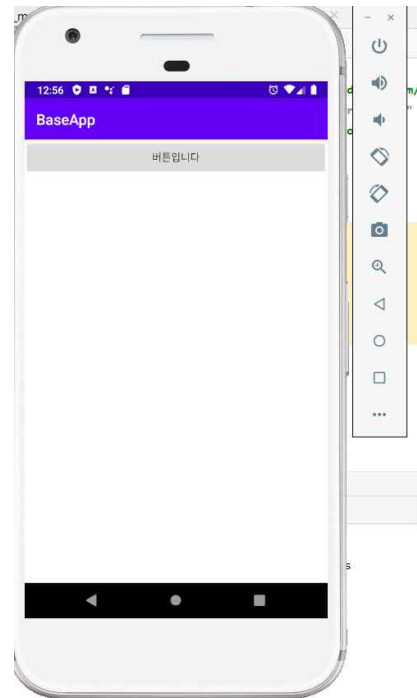
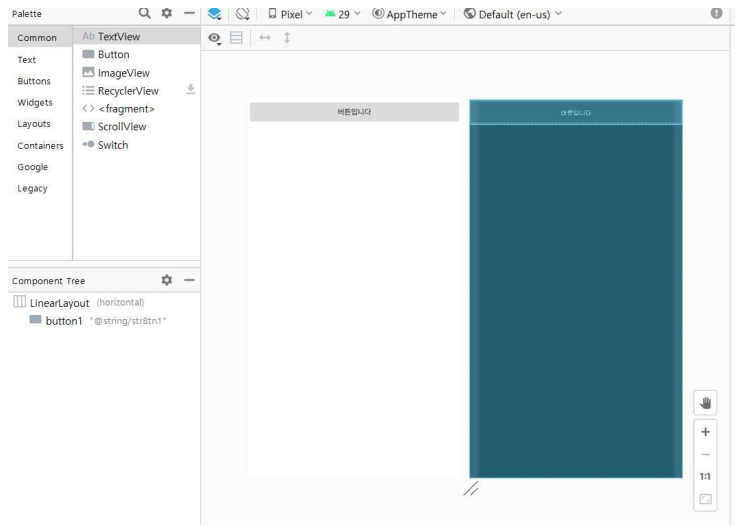
wrap_content : 포함된 내용을 보여줄 수 있는 크기로 설정

match_parent : 부모 뷰의 내부 여백을 제외한 크기로 설정

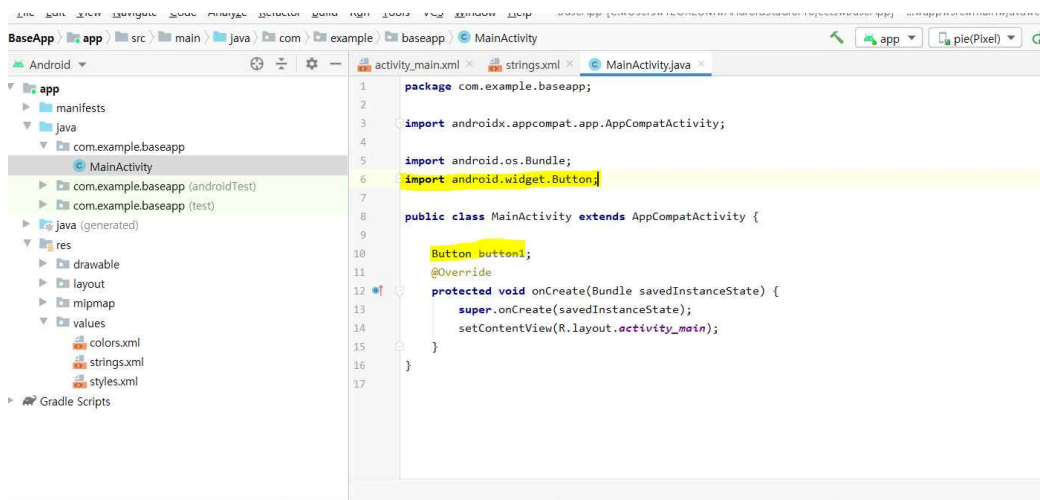
즉 android.layout_width 는 너비를 화면 전체에 차게 설정한 것.

wrap_content는 layout안의 뷰들을 나타낼 수 있을 정도의 크기만 자동으로 설정되는 것.

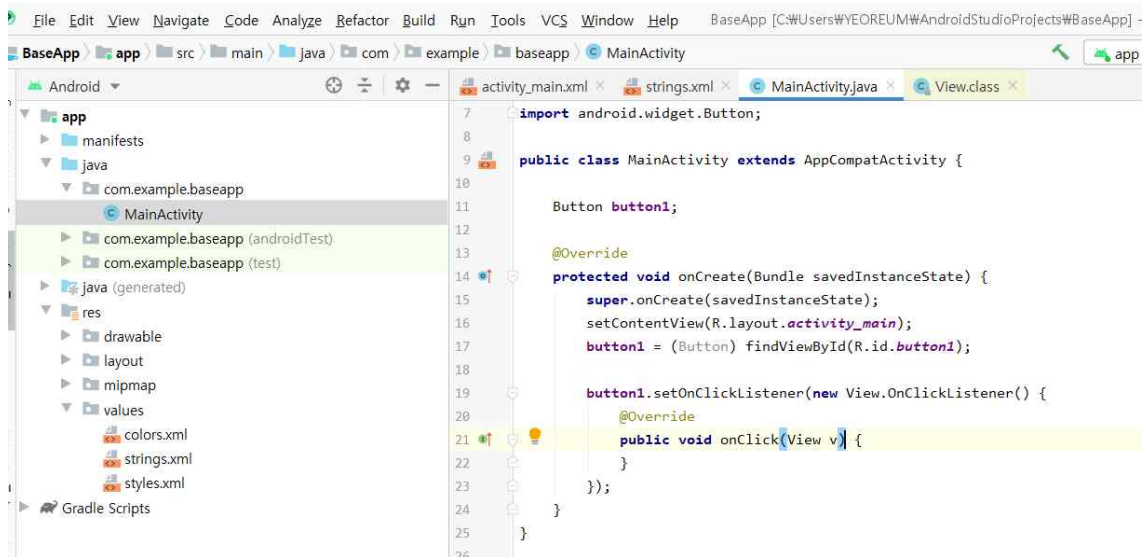
따라서 실행화면은 다음과 같다.



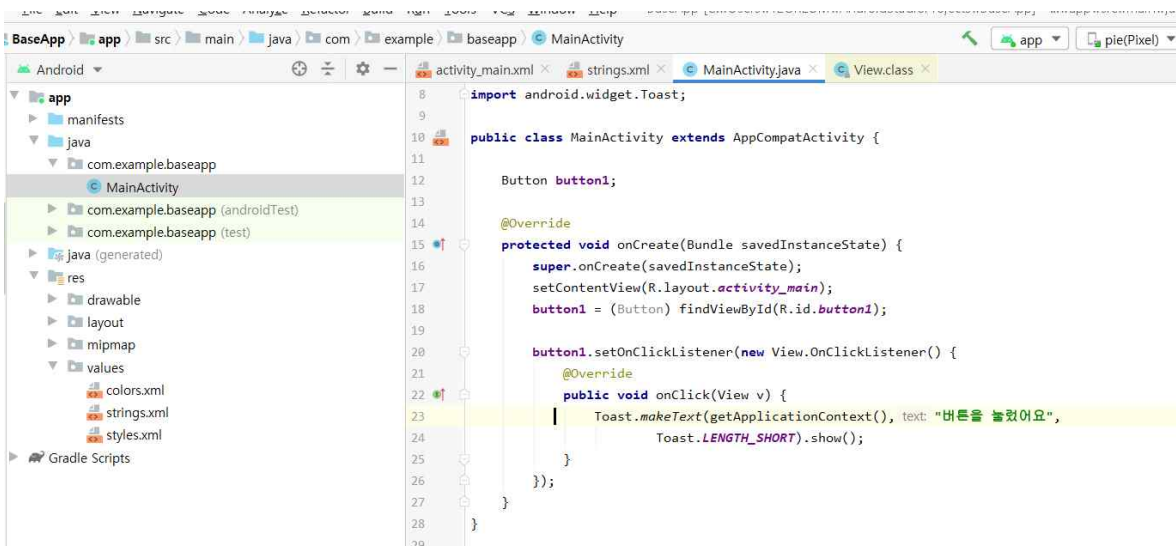
- activity_main.xml에서 봤을 때



- button과 관련된 클래스를 import문에 추가함.



- findViewById(R.id.button1) : R은 Resource의 준말, android의 resources를 담당하는 클래스이다. 각 화면의 view디자인을 담당하는 xml 파일 중 화면 담당을 보면 버튼, 텍스트 뷰, 이미지 등에 개별적으로 id가 부여되어있으므로 R.id.아이디를 불러와 처리하기 위해 findViewById란 메소드를 사용하며, 이 메소드를 사용하여 객체를 인스턴스화 시킴



클릭했을 때 화면에 보여질 것을 나타냄

- toast.makeText(getApplicationContext()).“출력할 문자열”,“Toast.LENGTH_SHORT).show()
- LENGTH_SHORT : 짧은 시간동안 보기 또는 문자 알림을 표시함
- android에서의 context?

앱에 대한 전반적인 정보를 가진 인터페이스이며 안드로이드 시스템으로부터 제공되는 추상클래스로 Activity를 띄우거나 Broadcast하거나와 같이 앱의 자원과 클래스에 접근하도록 한다.

- getApplicationContext() : return the context of th single, global Appliocation object of the curren process.

- `getApplicationContext()`는 앱에 대한 모든 것을 가진 context를 반환, 하나의 Activity내에서가 아닌, 앱의 생명주기가 context로 묶여있는 경우, `getApplicationContext()`로 해야함.



- “버튼입니다”를 클릭했을 때, 짧은 시간동안 아래 안내 문구가 뜬.

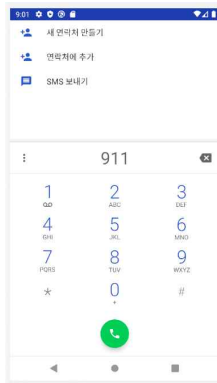
03. Slide 68의 '직접 풀어보기 2-3'에서 홈페이지 열기까지를 수행하고 1과 같은 결과 보고서를 작성하시오. (전화걸기, 갤러리 열기 수행 시 추가 점수) (박 재성)



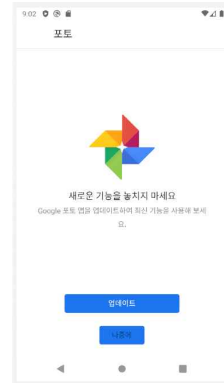
초기 화면



네이트 홈페이지 열기



911 응급전화 걸기



갤러리 열기

* 작성 과정

MainActivity.xml에서

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"><!--View 위젯들을 세로 방향으로 나열-->
```

레이아웃을 LinearLayout으로 설정하고

android:orientation 속성을 vertical로 설정한다.

```
<!--네이트 홈페이지 열기 버튼-->
<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/strBtn1" />
<!--911 응급전화 걸기 버튼-->
<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/strBtn2" />
<!--갤러리 열기 버튼-->
<Button
    android:id="@+id/button3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/strBtn3" />
<!--끝내기 버튼-->
<Button
    android:id="@+id/button4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/strBtn4" />
```

버튼 4개를 만든다.

android:layout_width="match_parent"

// 부모(레이아웃)에 좌우길이를 맞춤

android:layout_height="wrap_content"

// 콘텐츠(글씨) 크기에 상하길이를 맞춤

strings.xml에서

```
<resources>
    <string name="app_name">FourButton</string>
    <string name="strBtn1">네이트 홈페이지 열기</string>
    <string name="strBtn2">911 응급전화 걸기</string>
    <string name="strBtn3">갤러리 열기</string>
    <string name="strBtn4">끝내기</string>
</resources>
```

버튼 텍스트를 설정한다.

MainActivity.java에서

버튼 4개를 선언한 후 findViewById 함수를 사용해서 버튼을 지정해 준 후


```

public class MainActivity extends AppCompatActivity {
    Button button1; Button button2; Button button3; Button button4;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setIcon(R.drawable.icon_android); //앱 아이콘 설정
        button1 = (Button) findViewById(R.id.button1); //네이트 홈페이지 열기 버튼
        button2 = (Button) findViewById(R.id.button2); //911 응급전화 걸기 버튼
        button3 = (Button) findViewById(R.id.button3); //갤러리 열기 버튼
        button4 = (Button) findViewById(R.id.button4); //끝내기 버튼
        button1.setBackgroundColor(Color.GRAY); //네이트 홈페이지 열기 버튼 회색
        button2.setBackgroundColor(Color.GREEN); //911 응급전화 걸기 버튼 초록색
        button3.setBackgroundColor(Color.RED); //갤러리 열기 버튼 빨간색
        button4.setBackgroundColor(Color.YELLOW); //끝내기 버튼 노란색
    }
}

```

SetBackgroundColor 메소드를 사용해서 버튼의 색을 지정해준다.

```

button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { // button1 클릭 시 네이트 홈페이지를 연다
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://m.nate.com"));
        startActivity(mIntent);
    }
});
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { // button2 클릭 시 전화 창을 띄우고 911을 입력한다
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("tel:/911"));
        startActivity(mIntent);
    }
});
button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { // button3 클릭 시 갤러리를 연다
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("content://media/internal/images/media"));
        startActivity(mIntent);
    }
});
button4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { // button4 클릭 시 앱을 종료한다
        finish();
    }
});
}

```

onClick 매소드를 만들어서 버튼 클릭 시 어떤 동작을 할 것인지 선언한다.

한 줄의 코드로 홈페이지 열기, 911에 전화걸기, 갤러리 열기 등의 동작들이 가능한데, 안드로이드에서는 인텐트(Intent)를 이용해 애플리케이션 구성 요소 간에 데이터를 전달하거나 실행하려는 기능이 무엇인지 시스템에 알려 줄 수 있다.

git hub link

2017076063(박원진) : https://github.com/wonjinparkz/open_source_project

2017076064(김여름) : https://github.com/kyeoreum/opensource_project

2017076065(차서은) : https://github.com/helloseoeun/opensource_project

2017037061(왕소림)

2016038027(박재성) : https://github.com/qkrwotjd1731/open_source_project