

오픈소스 전문 프로젝트 보고서

-Four Button-

<8조>

담당 교수님 : 박수창 교수님

담당 조교님 : 김소연 조교님

제 출 일 : 2020년 3월 28일

학 과 : 컴퓨터 공학과

조 원 : 2018037021 강주형

2016038003 이찬진

2016038019 정주형

2016038022 최준호

2016038035 채범근

목 차

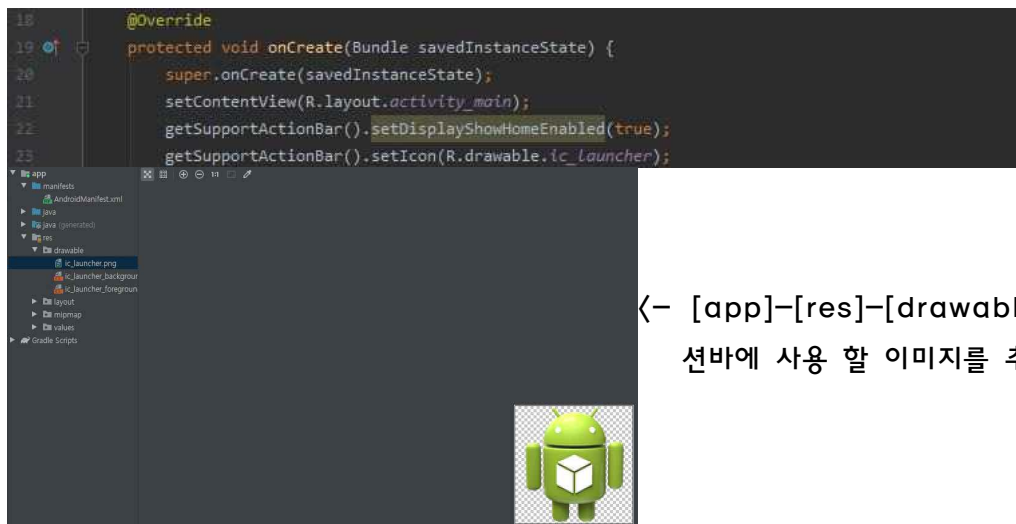
I. 액션바 생성, 홈페이지 버튼	3
1. 액션바 생성	3
2. 홈페이지 버튼	4
II. 추가 버튼	5
1. 응급전화 걸기 버튼	5
2. 갤러리 열기 버튼	6
3. 끝내기 버튼	7
4. 전체 코드	8
III. 실행	9
IV. XML 부분 문법	11
V. JAVA 문법	13

버튼을 생성하기 위해 사용된 메서드나 문법들은 주석이 지저분하고 길어질 것 같아 목
차 4,5 에 따로 설명한 부분 양해 부탁드립니다.

I. 액션바 생성, 홈페이지 버튼

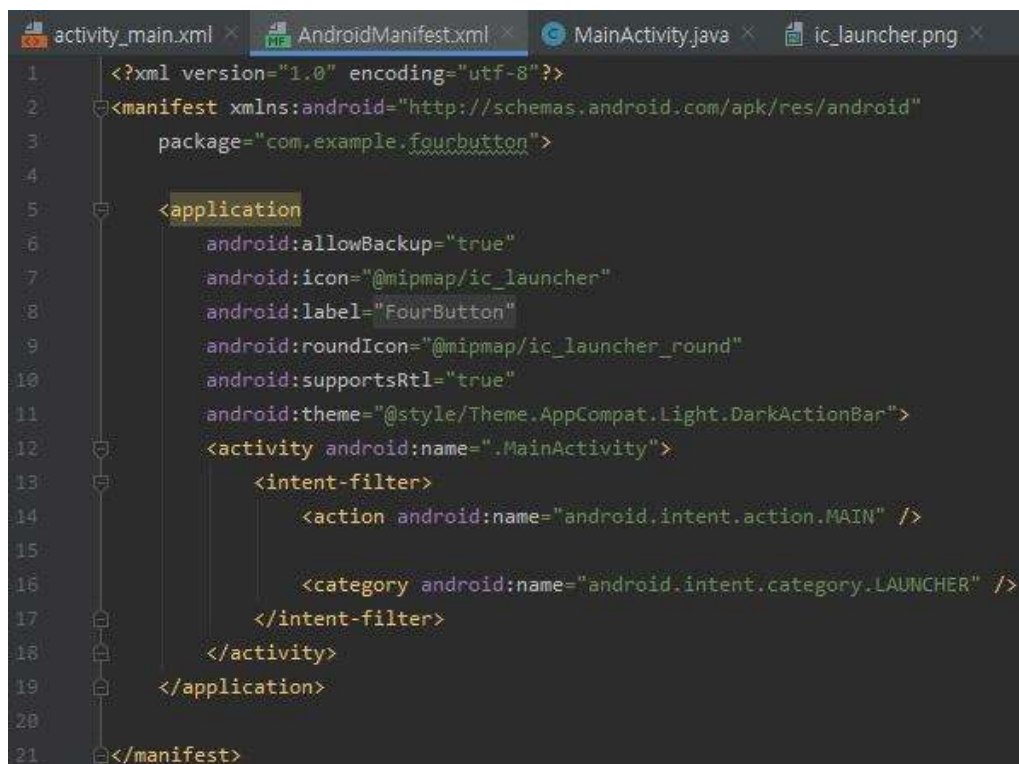
액션바 생성과 홈페이지 이동버튼은 채범근 조원이 담당하였다.

1. 액션바 생성



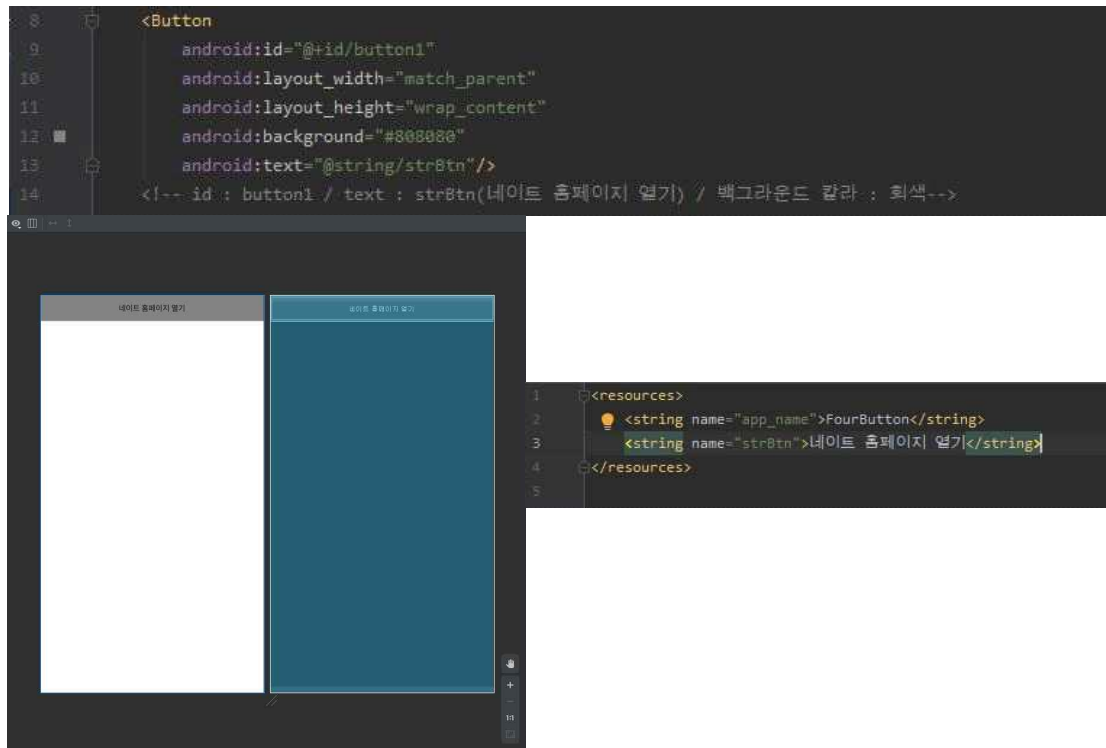
← [app]-[res]-[drawable]에 액션바에 사용 할 이미지를 추가한다.

1) Main에 `getSupportActionBar().setDisplayHomeAsUpEnabled(true);` 와 `getSupportActionBar().setIcon(R.drawable.ic_launcher);` 를 추가한다.

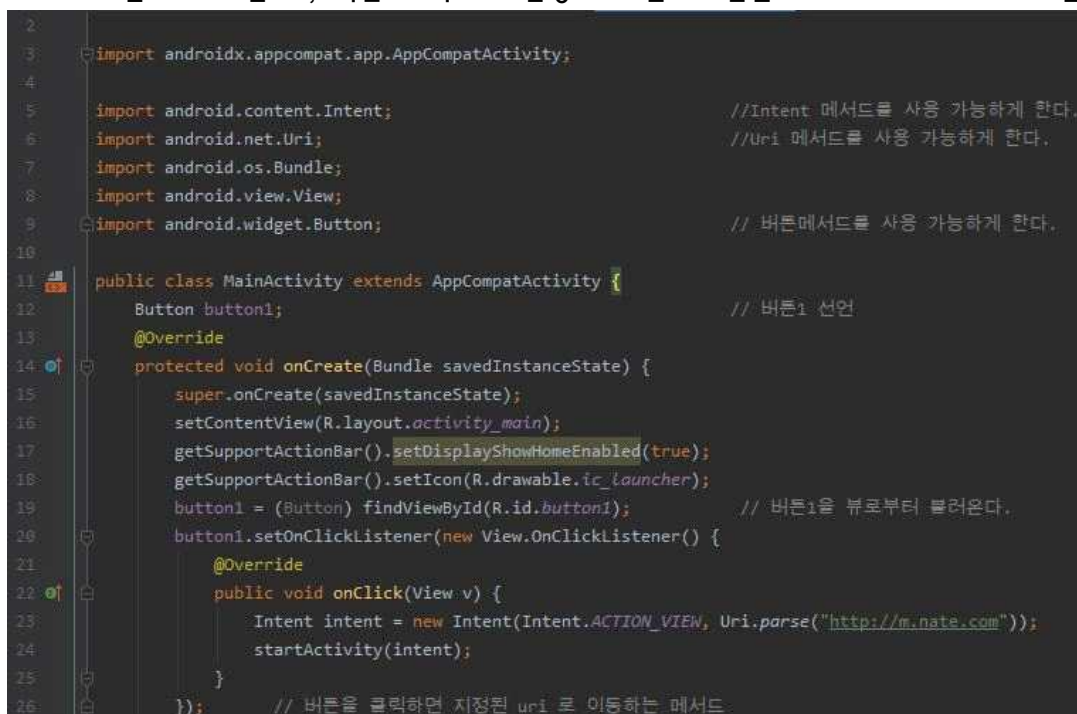


2) [app]-[manifests]-AndroidManifest.xml에서 11행 큰 따옴표 안의 내용을 “@style/Theme.AppCompat.Light.DarkActionBar” 로 고친다.

2. 홈페이지 버튼



1) 이름을 “button1”로 하는 버튼을 하나 추가하고 텍스트는 strBtn에 저장되어있는 “네이트 홈페이지 열기”, 색은 회색으로 설정하며 맨 위 칸을 가로로 차지하게 위치한다.

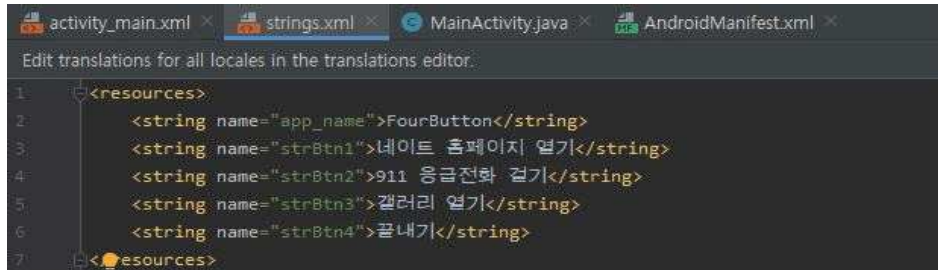


2) MainActivity.java에 클래스import, 변수 button1를 추가하고 뷰로부터 가져온다.
그리고 버튼을 클릭하면 지정된 uri로 이동하는 메서드를 추가한다.

II. 추가 버튼

추가 버튼 생성은 최준호 조원이 담당하였다.

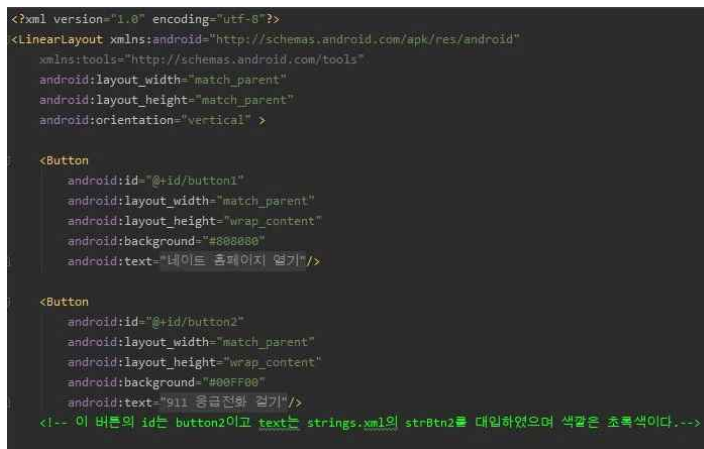
먼저 strings.xml에 변수들을 생성하자.



```
1 <resources>
2     <string name="app_name">FourButton</string>
3     <string name="strBtn1">네이트 홈페이지 열기</string>
4     <string name="strBtn2">911 응급전화 걸기</string>
5     <string name="strBtn3">갤러리 열기</string>
6     <string name="strBtn4">끝내기</string>
7 </resources>
```

버튼을 만들기 전에, 버튼에 대입 될 문자들을 [res]-[values]-strings.xml에 정의함.

1. 응급전화 걸기 버튼

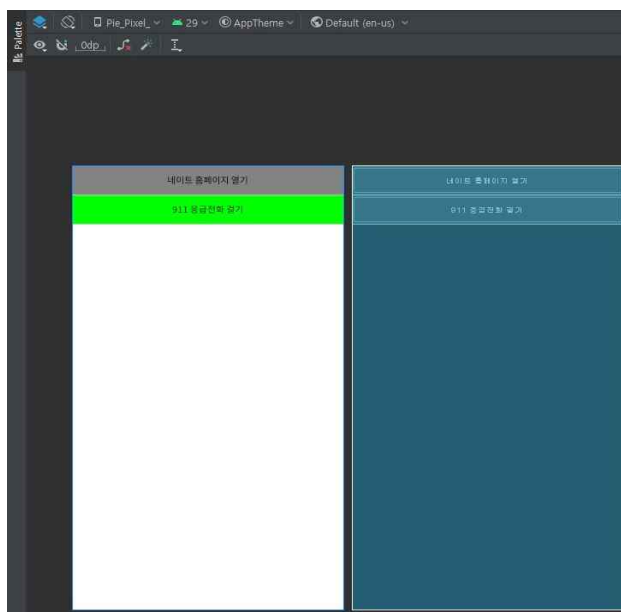


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#808080"
        android:text="네이트 홈페이지 열기"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#90EE90"
        android:text="911 응급전화 걸기"/>

<!-- 이 버튼의 id는 button2이고 text는 strings.xml의 strBtn2를 대입하였으며 색깔은 초록색이다.-->
```



- 1) id를 "button2"로 하는 버튼을 하나 추가하고 텍스트는 strBtn2에 저장되어있는 "911 응급전화 걸기", 색은 초록색으로 설정하며 button1의 아래에 위치한다.

```

1 package com.example.fourbutton;
2
3 import ...
4
5
6
7
8
9
10
11 public class MainActivity extends AppCompatActivity {
12     Button button1;
13     Button button2;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
19         getSupportActionBar().setIcon(R.drawable.ic_launcher);
20         button1 = (Button) findViewById(R.id.button1);
21         button2 = (Button) findViewById(R.id.button2); // 버튼2를 가져온다
22         button1.setOnClickListener((v) -> {
23             Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://m.nate.com"));
24             startActivity(intent);
25         });
26         button2.setOnClickListener(new View.OnClickListener() {
27             @Override
28             public void onClick(View v) {
29                 Intent dial = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:911"));
30                 startActivity(dial);
31             }
32         }); // 버튼을 클릭했을때, 지정한 번호로 다이얼을 열어 입력을 시켜주는 함수이다.
33     }
34 }
35
36
37
38

```

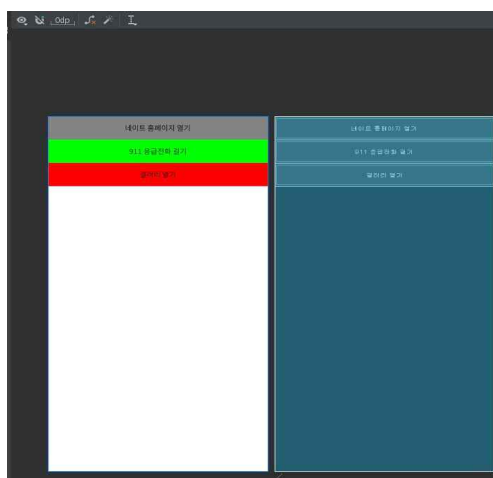
- 2) MainActivity.java에 변수 button2를 추가하고 xml로부터 가져온다.
그리고 버튼을 클릭하면 다이얼을 열어 911을 걸어주는 메서드를 추가한다.
이제 911 응급전화를 걸어주는 버튼이 완성되었다.

2. 갤러리 열기 버튼

```

48 <Button
49     android:id="@+id/button3"
50     android:layout_width="match_parent"
51     android:layout_height="wrap_content"
52     android:layout_below="@id/button2"
53     android:background="#FF0000"
54     android:text="갤러리 열기"
55 />
56
57 <!-- 이 버튼의 id는 button3이고 text는 strings.xml의 strBtn3을 대입하였으며 색깔은 빨간색이다.-->

```



- 1) id를 “button3”로 하는 버튼을 하나 추가하고 텍스트는 strBtn3에 저장되어있는 “갤러리 열기”, 빨간색으로 설정하며 button2의 아래에 위치한다.

```

4
5 import android.content.Intent;
6 import android.net.Uri; // Uri 클래스를 사용할 수 있다.
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10
11 public class MainActivity extends AppCompatActivity {
12     Button button1;
13     Button button2;
14     Button button3;
15     private int REQ_PICK_CODE = 100; // 상수 REQ_PICK_CODE 를 선언한다.
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
21         getSupportActionBar().setIcon(R.drawable.ic_launcher);
22         button1 = (Button) findViewById(R.id.button1);
23         button2 = (Button) findViewById(R.id.button2); // 버튼2를 가져온다.
24         button3 = (Button) findViewById(R.id.button3); // 버튼3를 가져온다.
25         button1.setOnClickListener((v) -> {
26             Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://m.nate.com"));
27             startActivity(intent);
28         });
29         button2.setOnClickListener((v) -> {
30             Intent dial = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:911"));
31             startActivity(dial);
32         }); // 버튼을 클릭했을때, 지정한 번호로 다이얼을 열어 입력을 시켜주는 메서드이다.
33         button3.setOnClickListener(new View.OnClickListener() {
34             @Override
35             public void onClick(View v) {
36                 Intent pickerIntent = new Intent(Intent.ACTION_PICK);
37                 pickerIntent.setType(android.provider.MediaStore.Images.Media.CONTENT_TYPE);
38                 pickerIntent.setData(android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
39                 startActivityForResult(pickerIntent, REQ_PICK_CODE);
40             }
41         }); // 버튼을 클릭했을때, 갤러리를 열어주는 메서드이다.
42     }
43 }

```

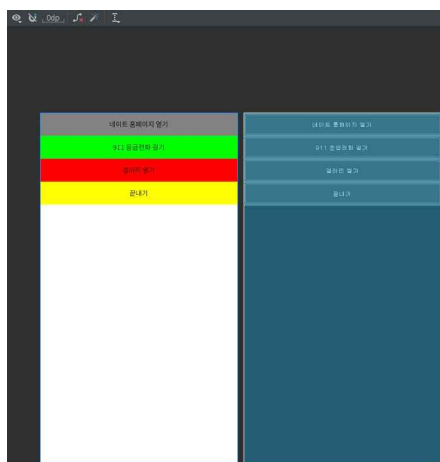
- 2) MainActivity.java에 변수 button3와 상수 REQ_PICK_CODE를 추가하고 xml로부터 가져온다. 그리고 버튼을 클릭하면 갤러리를 여는 메서드를 추가한다.
이제 갤러리를 여는 버튼이 완성되었다.

3. 끝내기 버튼

```

40 <Button
41     android:id="@+id/button4"
42     android:layout_width="match_parent"
43     android:layout_height="wrap_content"
44     android:layout_below="@id/button3"
45     android:background="#FFFF00"
46     android:text="@string/strBtn4"
47     />
48 <!-- 이 버튼의 id는 button4이고 text는 strings.xml의 strBtn4를 대입하였으며 색깔은 노란색이다.-->
49 </RelativeLayout>

```



1) id를 “button4”로 하는 버튼을 하나 추가하고 텍스트는 strBtn4에 저장되어있는 “끝내기”, 색은 노란색으로 설정하며 button3의 아래에 위치한다.

```
51 button4.setOnClickListener(new View.OnClickListener() {
52     @Override
53     public void onClick(View v) {
54         android.os.Process.killProcess(android.os.Process.myPid());
55     }
56 }); // 버튼을 클릭했을때, 실행 중인 프로그램을 종료시키는 메서드이다.
```

2) MainActivity.java에 변수 button4를 추가하고 xml로부터 가져온다. 그리고 버튼을 클릭하면 프로그램을 종료시키는 메서드를 추가한다.

이제 끝내는 버튼이 완성되었다.

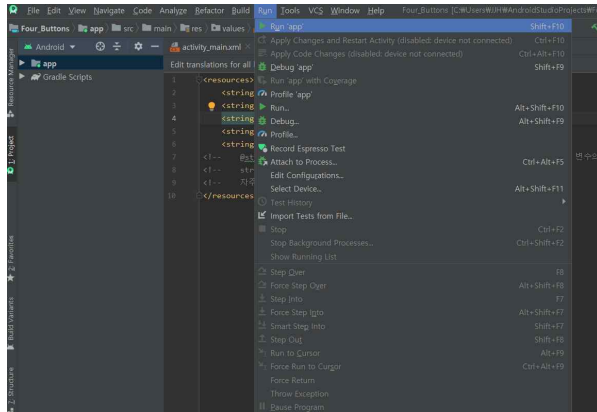
4. 전체 코드

```
activity_main.xml x strings.xml x MainActivity.java x
1 package com.example.fourbutton;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.net.Uri;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10
11 public class MainActivity extends AppCompatActivity {
12     Button button1;
13     Button button2;
14     Button button3;
15     Button button4;
16     private int REQ_PICK_CODE = 100;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
23         getSupportActionBar().setIcon(R.drawable.ic_launcher);
24         button1 = (Button) findViewById(R.id.button1);
25         button2 = (Button) findViewById(R.id.button2);
26         button3 = (Button) findViewById(R.id.button3);
27         button4 = (Button) findViewById(R.id.button4);
28         button1.setOnClickListener(new View.OnClickListener() {
29             @Override
30             public void onClick(View v) {
31                 Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://m.nate.com"));
32                 startActivity(intent);
33             }
34         });
35         button2.setOnClickListener(new View.OnClickListener() {
36             @Override
37             public void onClick(View v) {
38                 Intent dial = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:911"));
39                 startActivity(dial);
40             }
41         });
42         button3.setOnClickListener(new View.OnClickListener() {
43             @Override
44             public void onClick(View v) {
45                 Intent pickerIntent = new Intent(Intent.ACTION_PICK);
46                 pickerIntent.setType(android.provider.MediaStore.Images.Media.CONTENT_TYPE);
47                 pickerIntent.setData(android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
48                 startActivityForResult(pickerIntent, REQ_PICK_CODE);
49             }
50         });
51         button4.setOnClickListener(new View.OnClickListener() {
52             @Override
53             public void onClick(View v) {
54                 android.os.Process.killProcess(android.os.Process.myPid());
55             }
56         });
57     }
58 }
59
```

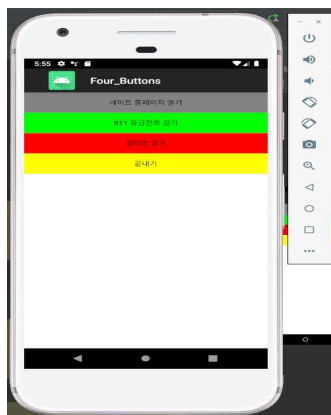

III. 실행

실행 부분은 강주형 조원이 담당하였다.

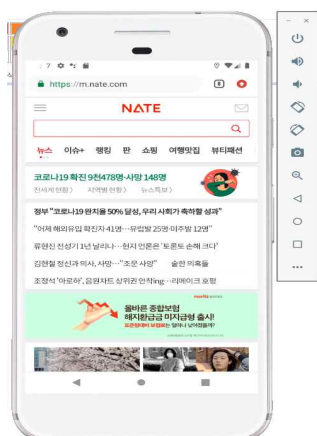
1.아래와 같이 [Run]-[Run 'app']을 눌러 실행시킨다.



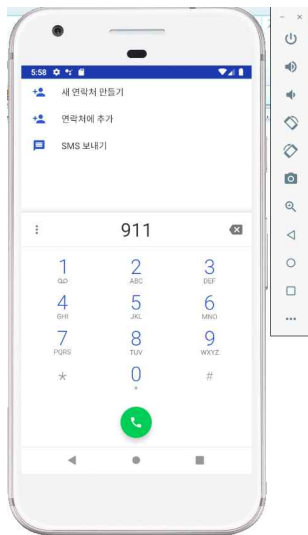
2.그러면 다음과 같이 AVD에 Four_Buttons이 실행됨을 확인할 수 있다.



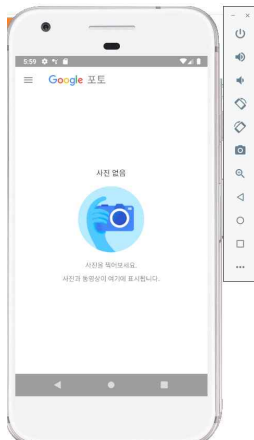
3.각 버튼을 클릭하면 아래 사진과 같이 결과가 출력된다.



〈네이트 홈페이지 열기〉



〈911 응급전화 걸기〉



〈갤러리 열기〉



〈끝내기〉

IV. XML 부분 문법

XML파일에서 사용되는 문법의 조사는 정주형 팀장이 담당하였다.

1. activity_main.xml에서 화면 디자인을 LinearLayout으로 설정하였다.Linear은 '(여러 단계의)선'이란 뜻으로, 위젯을 하나씩 나열하게 해준다.

orientation은 '방향'이라는 뜻을 가져,배치되는 위젯을 어느 방향으로 할지 정할 수 있다.

vertical은 '수직적인'이란 뜻이므로 위젯을 수직(기본적으로 아래방향)으로 놓게 된다.

horizontal은 '수평선의'란 뜻이므로 위젯을 수평(기본적으로 오른쪽방향)으로 놓게 된다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical" >
```

2. 각 버튼 내에 id를 지정했는데, 이는 버튼에 이름을 주었다고 볼 수 있다.

3. 버튼마다 색을 다르게 지정해야 한다. 위젯의 색상은 android:background로 설정할 수 있고,컴퓨터는 green,red같은 추상적인 표현을 모르기 때문에 색상코드를 직접 입력해야 한다.

4.버튼의 layout(배치) width(폭),height(높이)가 각각 match_parent, wrap_content이다. 폭과 높이에 대한 설명은 다음과 같다.

1)match_parent : 부모의 크기에 맞추다.

버튼은 프로그램에서 파생됐으므로 프로그램이 부모라 할 수 있다.부모의 크기라 함은,화면의 크기에 맞춘다고 볼 수 있다.따라서 폭을 화면에 맞춘다는 뜻이다.

2)wrap_contetn : 기존 사이즈를 사용하다

버튼의 기존 사이즈를 사용한다는 뜻이다. 따라서 기존의 버튼 높이를 사용한다는 뜻이다. 직접 조절하지 않으면 default값을 사용하고 원한다면 사이즈를 수직로 바꿀 수 있다.

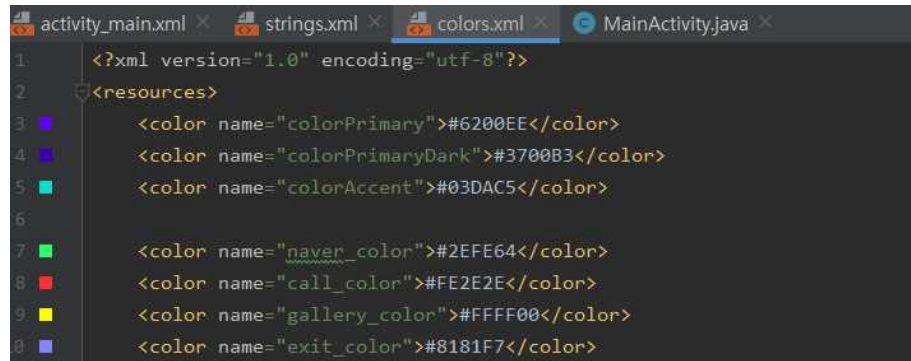
```
<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#508080"
    android:text="네이트 홈페이지 열기"/>

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00FF00"
    android:text="911 응급전화 걸기"/>

<!-- 이 버튼의 id는 button2이고 text는 strings.xml의 strBtn2를 대입하였으며 색깔은 초록색이다.-->
```

(2,3번 항목의 예시)

※색상을 자주 사용하면 우리가 보기 힘든 색상코드를 쓰면 보기 어려울 수 있다.
이때 res파일의 values파일 내 colors.xml에 이러한 색상코드를 보기 쉽게 정의할 수 있다.또한, strings.xml에 자주 쓸 문자도 새로 정의할 수 있다.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="colorPrimary">#6200EE</color>
4     <color name="colorPrimaryDark">#3700B3</color>
5     <color name="colorAccent">#03DAC5</color>
6
7     <color name="naver_color">#2EFE64</color>
8     <color name="call_color">#FE2E2E</color>
9     <color name="gallery_color">#FFFF00</color>
10    <color name="exit_color">#8181F7</color>
```

(색상코드를 다른 이름으로 정의한 예시)

V. JAVA 문법

MainActivity.java파일에서 사용되는 문법의 조사는 이찬진 조원이 담당하였다.

1. 버튼 1에 사용된 코드입니다.

```
button1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent myIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://m.nate.com"));  
        startActivity(myIntent);  
    }  
});
```

위에 코드에서 알아야 하는 것은 Intent와 Uri, startActivity 정도가 있습니다. Intent는 앱 컴포넌트가 무엇을 할 것인지를 담는 메시지 객체입니다. 메시지를 사용하는 목적은 다른 액티비티, 서비스, 브로드캐스트 리시버, 콘텐츠 프로바이더 등을 실행하는 것입니다. Intent는 그 사이에 데이터를 주고 받기 위한 용도로 사용됩니다. 여기서는 화면 전환을 위해 사용했다고 보면 됩니다.

Uri는 콘텐츠 프로바이더의 주소를 나타낸다. 콘텐츠 프로바이더는 실제의 주소를 대신해 가상의 고유주소를 가짐으로써 외부의 접근을 대신해서 맡아주는 것을 말한다. 따라서 Uri를 사용하면 홈페이지 주소 뿐만 아니라 파일 경로 데이터베이스로의 접근 등을 할 수 있다.

startActivity가 사용되는 이유는 화면은 하나의 액티비티이기 때문에 화면 간의 이동을 하는 것은 하나의 액티비티를 띄우는 과정이다. 따라서 startActivity()가 사용된다.

2. 버튼 2에 사용된 코드입니다.

```
button2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent myIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("tel:911"));  
        startActivity(myIntent);  
    }  
});
```

버튼 1에 사용된 것과 똑같기에 여기서는 tel:만 알면 되는 데 tel:은 전화걸기 링크를 말합니다.

3. 버튼 3에 사용된 코드입니다.

```
button3.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent myIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("content://media/internal/images/media"));  
        startActivity(myIntent);  
    }  
});
```

여기서는 content://media/internal/images/media를 알 필요가 있습니다. content:는 웹페이지 http://와 비슷하다고 볼 수 있습니다. 버튼 1에서 Uri를 설명할 때 Uri는 콘텐츠 프로바이더의 주소를 통해 접근한다고 하였는데 content://가 콘텐츠 프로바이더의 시작 주소이다. 그래서 위에 코드는 콘텐츠 프로바이더를 이용해 스토리지에 미디어폴더로 접근하여 갤러리를 보여준다.

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    final int REQ_PICK_CODE=100;
```

```
button3.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent intent = new Intent(Intent.ACTION_PICK);  
        intent.setType(android.provider.MediaStore.Images.Media.CONTENT_TYPE);  
        intent.setData(android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);  
        startActivityForResult(intent, REQ_PICK_CODE);  
    }  
});
```

위와 같은 방법으로도 갤러리를 불러올 수 있는데 위 방법은 갤러리 파일 선택 액티비티를 띄우는 방법이다.

4. 버튼 4에 사용된 코드입니다.

```
button4.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        finish();  
    }  
});
```

여기서는 현재 실행중인 activity를 종료하는 finish()가 사용되었습니다.

하지만 finish()말고도

android.os.Process.killProcess(android.os.Process.myPid())라는 명령어를 사용할 수도 있다.