

1번

(1). 안드로이드 창 activity_main.xml

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The code editor displays the XML layout file 'activity_main.xml' for the 'MainActivity' class. The layout consists of a horizontal LinearLayout containing an ImageView and two TextViews. The first TextView has Korean text and bold styling. The second TextView has Korean text and larger margins. A performance improvement notification is visible at the bottom right.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:orientation="horizontal">

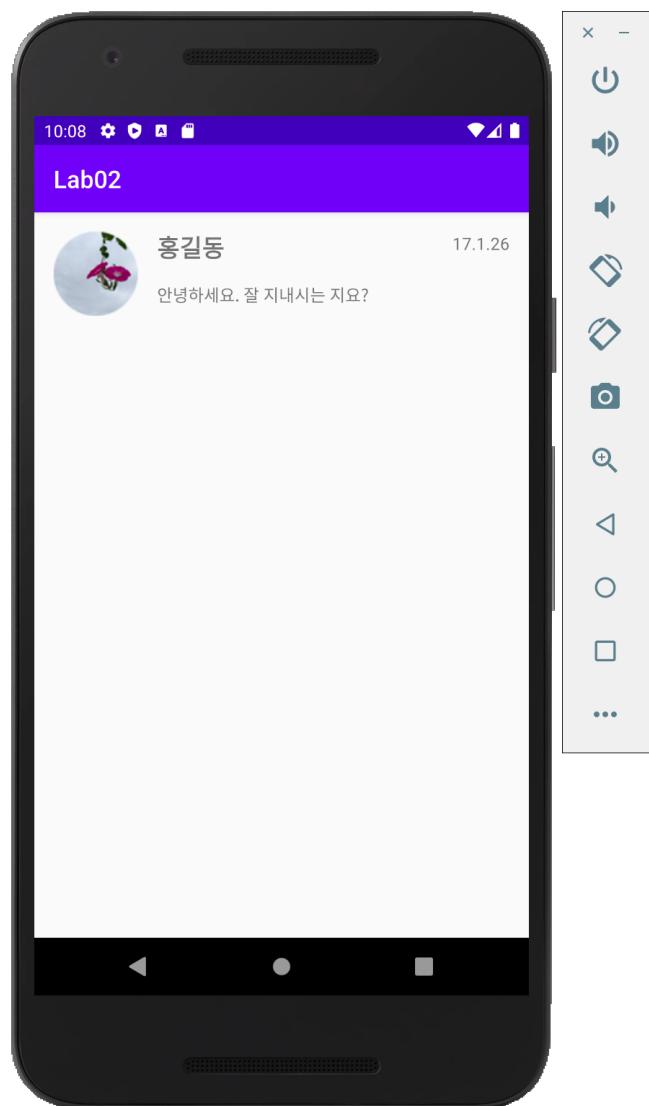
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/icon" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_weight="1" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="홍길동"
            android:textSize="20dp"
            android:textStyle="bold"
            android:layout_marginLeft="16dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="안녕하세요. 잘 지내시는지요?"
            android:layout_marginTop="16dp"
            android:layout_marginLeft="16dp" />
    </LinearLayout>
</LinearLayout>
```

Studio performance could be improved
Increasing the maximum heap size from...
Actions ▾ Don't show again

(2). AVD 실행 화면



(3). 학습한 내용

android:orientation="horizontal">

LinearLayout 의 horizontal 옵션은 뷰들을 수평으로 배치시킨다.

android:padding="16dp"

padding 은 안쪽으로의 여백을 의미한다.

android:layout_weight="1" >

두번째 LinearLayout에서 layout_weight를 1로 주었고, 나머지 ImageView와 마지막 TextView는 width를 wrap_content 주었기 때문에 ImageView와 세번째 TextView가 차지하고 남은 여백은 모두 LinearLayout 것이다.

android:orientation="vertical"

vertical로 옵션은 뷰들을 수직으로 배치한다.

android:layout_marginLeft="16dp" />

layout_marginLeft는 개체의 왼쪽 마진을 의미한다. margin은 바깥쪽으로의 여백을 의미한다.

android:layout_marginTop="16dp"

마찬가지로 layout_marginTop은 개체의 위쪽 마진을 의미한다.

android:layout_height="wrap_content"

마지막 TextView의 height도 wrap_content 이기 때문에 '홍길동' text를 가진 TextView와 같은 선상에 위치한다.

2부

(1). 안드로이드 창 activity_lab02_2.xml

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The 'activity_lab02_2.xml' file is selected in the 'layout' directory of the 'lab02' module. The code editor displays the following XML structure:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <ImageView
        android:id="@+id/icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/icon" />

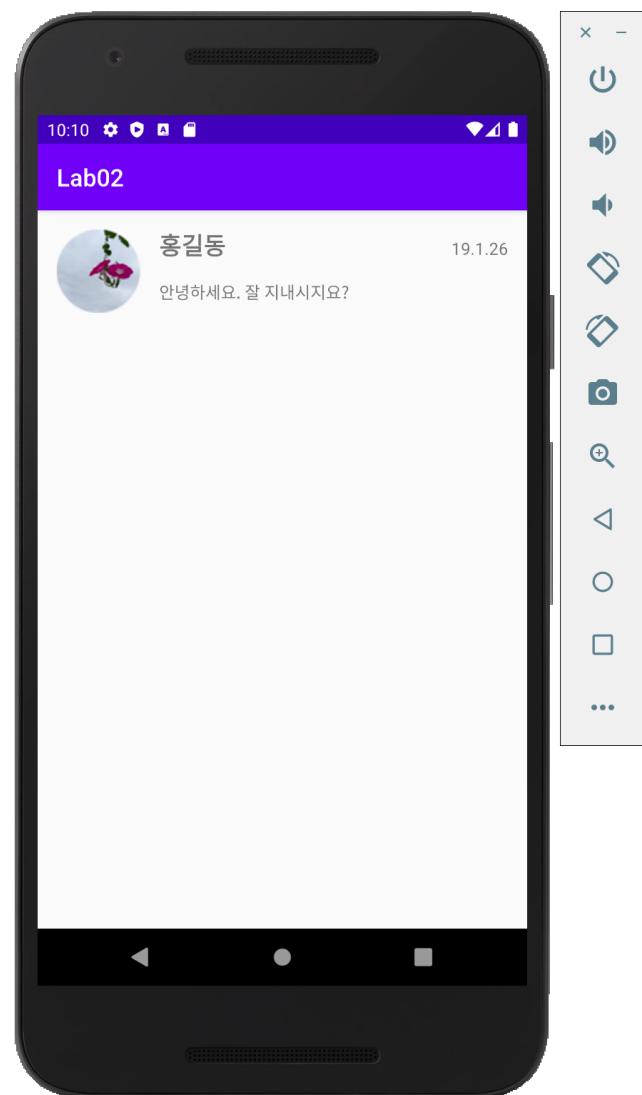
    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:text="홍길동"
        android:textSize="20dp"
        android:textStyle="bold"
        android:layout_toRightOf="@+id/icon" />

    <TextView
        android:id="@+id/content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="안녕하세요. 잘 지내시지요?"
        android:layout_below="@+id/name"
        android:layout_alignLeft="@+id/name" />

    <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

The code editor has syntax highlighting and code completion. A tooltip at the bottom right says: 'Studio performance could be improved Increasing the maximum heap size from... Actions ▾ Don't show again'. The bottom status bar shows: 'Install successfully finished in 228 ms.: App restart successful without requiring a re-install. (a minute ago)'. The bottom right corner shows the time as 26:45 and other system information.

(2). AVD 실행 화면



(3). 학습한 내용

```
android:layout_width="match_parent"  
android:layout_height="match_parent"
```

RelativeLayout은 width와 height가 match_parent이기 때문에 전체 화면을 차지한다.

```
android:padding="16dp">
```

padding이 16dp이기 때문에 전체 화면보다 가로 세로가 16x2 dp 씩 작은 화면이다.

```
android:layout_toRightOf="@id/icon" />
```

layout_toRightOf는 기준 뷰의 오른쪽에 해당 뷰를 위치시킨다.

```
android:layout_marginTop="16dp"
```

두번째 TextView는 우선, layout_marginTop으로 위쪽 뷰로부터의 마진, 간격이 16dp 떨어져 있고,

```
android:layout_below="@id/name" />
```

첫번째 TextView의 아래쪽에 위치한다.

```
android:layout_alignLeft="@id/name" />
```

또한 layout_alignLeft를 통해 기준이 되는 뷰 좌측 오른쪽에 배치된다.

```
android:layout_alignBaseline="@id/name" />
```

마지막 TextView는 기준 뷰와 텍스트와 기준선을 맞추고,

```
android:layout_alignParentRight="true" />
```

해당 뷰를 상위 뷰의 오른쪽에 위치시킨다.

1번

1. 안드로이드 창 Lab02_3Activity.java

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The code editor displays the Java file `Lab02_3Activity.java`. The code implements a `TabHost` to manage three tabs. The `activity_lab02_3.xml` layout file is also visible in the layout folder. A success message at the bottom indicates the app was installed successfully.

```
1 package com.example.lab02;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.core.content.res.ResourcesCompat;
5
6 import android.os.Bundle;
7 import android.widget.TabHost;
8
9 public class Lab02_3Activity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_lab02_3);
15
16         TabHost tabHost = findViewById(R.id.host);
17         tabHost.setup();
18
19         TabHost.TabSpec spec = tabHost.newTabSpec("tab1");
20         spec.setIndicator(null, ResourcesCompat.getDrawable(getResources(), R.drawable.tab_icon1, null));
21         spec.setContent(R.id.tab_content1);
22         tabHost.addTab(spec);
23
24         spec = tabHost.newTabSpec("tab2");
25         spec.setIndicator(null, ResourcesCompat.getDrawable(getResources(), R.drawable.tab_icon2, null));
26         spec.setContent(R.id.tab_content2);
27         tabHost.addTab(spec);
28
29         spec = tabHost.newTabSpec("tab3");
30         spec.setIndicator(null, ResourcesCompat.getDrawable(getResources(), R.drawable.tab_icon3, null));
31         spec.setContent(R.id.tab_content3);
32         tabHost.addTab(spec);
33
34     }
35 }
```

Install successfully finished in 1 s 395 ms.

Emulator: emulator: INFO: QtLogger.cpp:68: Critical: Uncaught TypeError: Cannot read property 'update' of undefined (qrc:/html/js/location-mock-web-channel.js:130, (null)) (a minute ago) 5:1 LF UTF-8 4 spaces 😊 😐

1. 안드로이드 창 activity_lab02_3.xml

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The left sidebar displays the project structure, including modules 'lab01' and 'lab02', and Java files like 'Lab02_2Activity', 'Lab02_3Activity', and 'MainActivity'. The main editor window shows the XML code for 'activity_lab02_3.xml'. The code defines a TabHost with two tabs, each containing a FrameLayout with a TextView. A tooltip at the bottom right suggests increasing the heap size.

```
<?xml version="1.0" encoding="utf-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/host"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TabWidget
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimary" />

        <FrameLayout
            android:id="@+id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

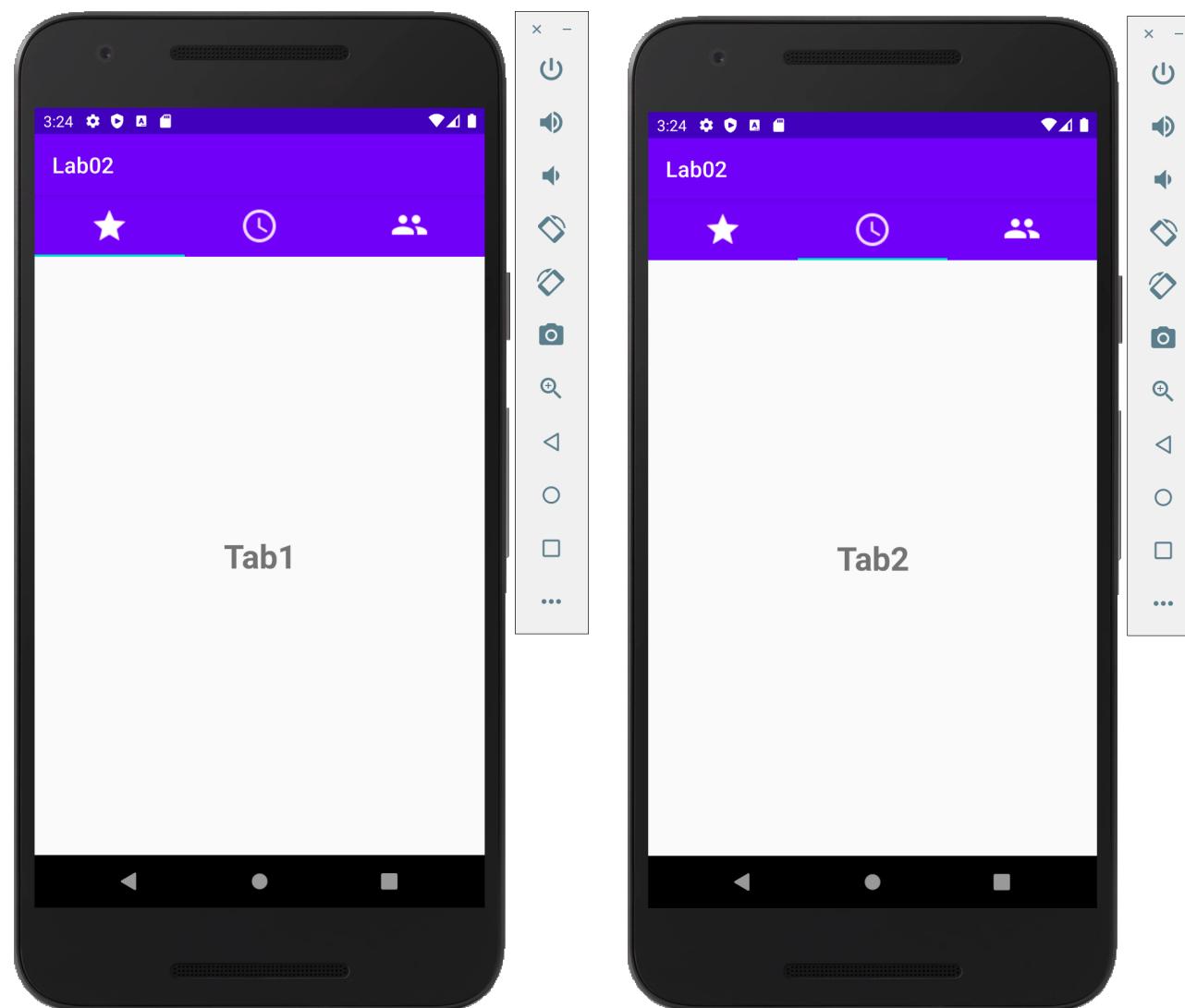
            <TextView
                android:id="@+id/tab_content1"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="Tab1"
                android:textSize="30dp"
                android:textStyle="bold"
                android:gravity="center" />

            <TextView
                android:id="@+id/tab_content2"
                android:layout_width="match_parent"
                android:layout_height="match_parent" />
        
    

```

Studio performance could be improved
Increasing the maximum heap size from...
Actions ▾ Don't show again

2. AVD 실행 화면. (초기화면 - 두번째 버튼을 눌렸을 때)



3. 학습한 내용

(1). activity_lab02_3.xml

<TabHost

탭 화면은 FrameLayout 을 이용할 수도 있고, TabHost 를 이용할 수도 있다.

TabHost 를 이용하면 조금 더 쉽게 기능들을 작성할 수 있다.

`android:orientation="vertical">`

LinearLayout 에는 TabWidget 과 FrameLayout 을 갖고 있다. 또한 이 뷰들은 수직으로 배치된다.

`<TabWidget android:id="@+id/tabs" | <FrameLayout android:id="@+id/tabcontent"`

TabHost 는 반드시 TabWidget 과 FrameLayout 으로 구성해야 한다. 간단히 설명하면, TabHost 는 탭 화면 전체 영역을 뜻하고, TabWidget 은 탭 버튼이 들어갈 영역, FrameLayout 은 탭 버튼 클릭 시 보일 화면 영역이다. 또한 TabWidget 과 FrameLayout 의 id 값은 이미 약속되어 있는데, TabWidget 은 "@+id/tabs", FrameLayout 은 "@+id/tabcontent"이다.

`android:gravity="center" />`

이 실습 앱에서는 3 개의 탭을 사용하므로 3 개의 탭에 대한 3 개의 뷰가 필요하다. FrameLayout 은 3 개의 TextView 를 갖고 있다. 3 개의 TextView 들은 gravity 가 center 로 지정해있기 때문에 지정한 text 가 TextView 의 정 가운데에 나타난다.

(2). Lab02_3Activity.java

```
tabHost.setup();
```

TabHost 를 사용할 때 먼저 setup 을 해주어야 한다.

```
TabHost.TabSpec spec = tabHost.newTabSpec("tab1");
```

TabSpec 클래스는 탭 버튼과 본문 화면의 짹을 맺어주는 역할을 수행한다. TabSpec 객체를 생성할 때 매개변수의 문자열은 각 TabSpec 의 식별자 문자열이다.

```
spec.setIndicator(null, ResourcesCompat.getDrawable(getResources(), R.drawable.tab_icon1, null));
```

TabSpec에서 탭 버튼은 indicator이다. setIndicator를 통해 drawable 파일의 tab_icon1을 버튼의 이미지로 설정한다. 이때 매개변수가 drawable 객체이므로 getDrawable을 사용한다.

```
spec.setContent(R.id.tab_content1);
```

TabSpec에서 본문은 content이다. setContent를 통해 activity_lab02_3.xml에서 생성한 TextView들을 본문으로 지정해준다.

```
tabHost.addTab(spec);
```

이렇게 각각의 탭 버튼과 본문을 짹 지은 spec을 tabHost에 추가해주면 하나의 탭에 대한 하나의 본문 연결이 완성된다. 이 과정을 3번 반복하면 3개의 버튼에 대한 3개의 본문이 연결되는 것을 확인할 수 있다.

2부

(1). 안드로이드 창 activity_lab02_4.xml

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The left sidebar displays the project structure, including modules 'lab01' and 'lab02', and various Java and XML files. The main editor window shows the XML code for 'activity_lab02_4.xml'. The code defines a vertical LinearLayout containing a TextView and a horizontal LinearLayout. This horizontal layout contains a TableLayout with two rows. Each row contains a Button. The code uses attributes like 'match_parent', 'wrap_content', and 'weight' to define the layout parameters. A floating notification at the bottom right suggests improving studio performance by increasing heap size. The bottom status bar indicates a successful app restart.

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TableLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="3">

            <TableRow>

                <Button
                    android:layout_width="0dp"
                    android:layout_weight="1"
                    android:text="7"
                    android:paddingTop="32dp"
                    android:paddingBottom="32dp" />

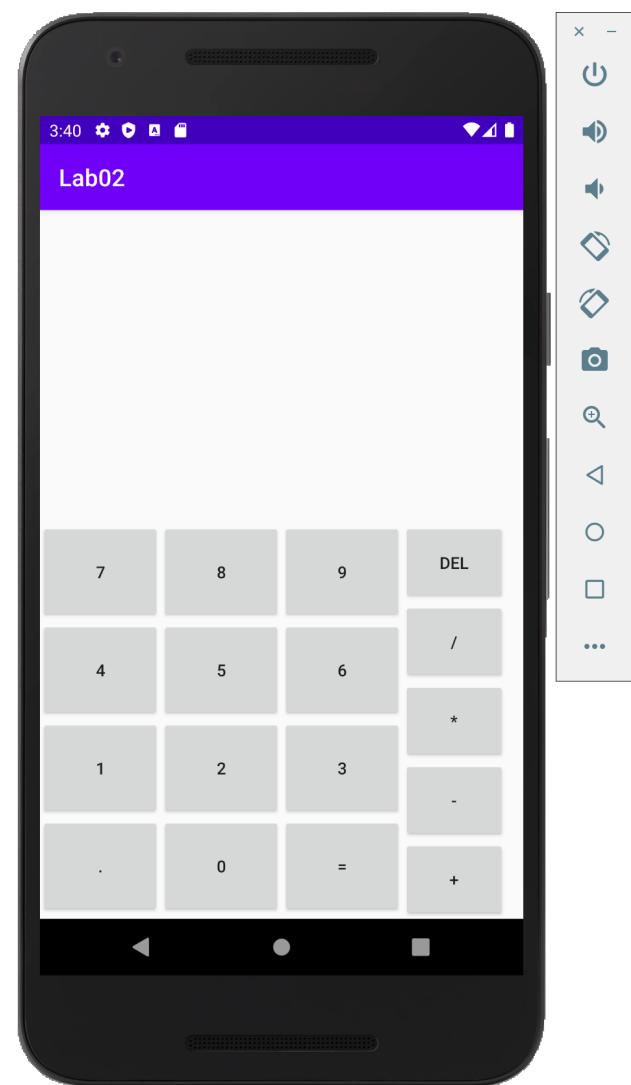
                <Button
                    android:layout_width="0dp"
                    android:layout_weight="1" />
            
    

```

Studio performance could be improved
Increasing the maximum heap size from...
Actions ▾ Don't show again

Install successfully finished in 157 ms.: App restart successful without requiring a re-install. (a minute ago)

(2). AVD 실행 화면



(3). 학습한 내용

activity_lab02_4.xml의 구조

먼저 LinearLayout에 TextView와 LinearLayout을 수직으로 배치시킨다. 수직으로 배치된 두번째 LinearLayout에는 TableLayout과 GridLayout을 갖고 있다. TableLayout은 4개의 행이 있고, 각각의 행은 3개의 버튼을 갖고 있다. GridLayout은 5개의 버튼을 갖고 있다.

<TextView android:layout_height="0dp" android:layout_weight="1" /> | <LinearLayout android:layout_height="wrap_content"
LinearLayout의 높이를 wrap_content로 지정했기 때문에 LinearLayout이 보여지는 최소한의 높이를 사용한다. TextView에서는 높이가 0dp이고,
weight를 1로 지정했기 때문에 LinearLayout이 보여지고 남은 화면에는 모두 TextView가 보이게 된다.

<TableLayout android:layout_width="0dp" android:layout_weight="3">
<GridLayout android:layout_width="0dp" android:layout_weight="1"

TableLayout과 GridLayout은 모두 너비가 0dp로 지정하고 layout_weight를 줌으로써 비율을 통해 너비가 결정된다. TableLayout이 3,
GridLayout이 1 만큼의 3:1 비율로 너비가 지정된다.

<TableRow> </TableRow>

이 태그를 사용해 TableLayout 행의 뷰들을 지정해준다. 이 앱에서는 TableRow에 3개의 버튼을 할당한다.

<Button android:layout_weight="1"

3개의 버튼들은 너비가 0dp이고 wieght를 1로 지정해줬기 때문에 1:1:1 비율로 너비가 지정된다.