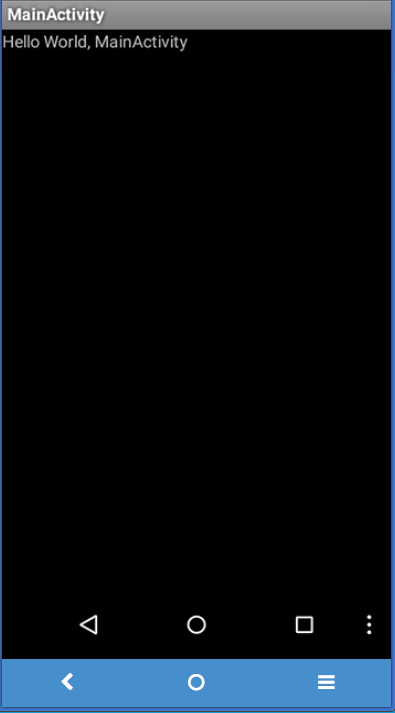
**LAB 2: Simple UI**

1. **Create simple app and basic command builder**

//%appdata% to find the hidden AppData

1. Goto path C:\Users\ekarat\AppData\Local\Android\sdk\tools

2. Generate project

android create project --target 1 --path d:\lab2 --activity MainActivity --package com.tni.mobile.ekarat

3. Set JAVA\_HOME

C:\>setx JAVA\_HOME " "c:\Program Files (x86)\Java\jdk1.8.0\_91""

4. Use ant to compile inside d:\lab2

D:\apache-ant-1.9.7\bin\ant dubug

//If not works, delete /.android/debug.keystore

5. Show the available android version

android list target

6. Call sdk manager

android sdk

7. Create new EMU

android create avd -n myEMU -t 1

8. Call myEMU

emulator @myEMU

//9. Install apk on emulator

adb -s emulator-5554 install [AppName]-debug.apk

D:\lab2\bin>c:\users\ekarat\AppData\Local\Android\Sdk\platform-tools\adb -s emulator-5554 install MainActivity-debug.apk

//Option, uninstall

adb uninstall com.tni.mobile.ekarat

1. **Building Simple UI (Layout)**
   1. Add EditText
      * Replace TextView with EditText node

<EditText android:id="@+id/edit\_message"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:hint="@string/edit\_message" />

* + - Compile & Run, Success or fail? Why?
  1. Add Button

<Button  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="@string/button\_send" />

* + - Compile & Run, Success or fail? Why?

Exercise

2.1 Change the layout’s orientation to “horizontal”.

2.2 Change the button XML to

<Button

android:layout\_width="188dp"

android:layout\_height="wrap\_content"

android:layout\_x="126px"

android:layout\_y="361px"

android:text="@string/button\_send" />

2.3 Change the linear layout to absolute layout, and adjust button size and position arbitarily.

2.4 Modify the XML to relative layout as shown in page 13 of lecture 2.

1. **Building simple UI (view)**
   1. Icon button
      * Create a folder “drawable” inside the folder res
      * Put green.png into /res/drawable

<ImageButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/green" />

* 1. Event
     + Add line below
     + android:onClick="sendMessage" />
     + In MainActivity.java add

public void sendMessage(View view)

{

Context context = getApplicationContext();

CharSequence text = "Hello toast!";

int duration = Toast.LENGTH\_SHORT;

Toast toast = Toast.makeText(context, text, duration);

toast.show();

}

//import android.view.View;

//import android.content.Context;

* 1. Custom background button
     + Create three png files (green.png, red.png, white.png) and save them alls inside res/drawable
     + Create a new XML (button\_custom.xml) and put it inside res/drawable/

<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:drawable="@drawable/green"  
          android:state\_pressed="true" />  
    <item android:drawable="@drawable/red"  
          android:state\_focused="true" />  
    <item android:drawable="@drawable/white" />  
</selector>

* + - Add Button xml

<Button  
    android:id="@+id/button\_send"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button\_send"  
    android:onClick="sendMessage"  
    android:background="@drawable/button\_custom"  />

Exercise

3.1 Implement page 24 of lecture 2

1. **Experiment by yourself & HW2 (Follows the lecture 2)**
   1. Create EditText
      * Change input type to “text”, “textEmailAddress”, “textUri”, “number”, “phone”
   2. Checkboxes
      * Create three checkbox (design whatever you want) with three title “rock”, “pop”, “dance”
      * Show Toast with the checked, e.g., if “rock” and “pop” are checked, then show “rock, pop”.
   3. Radio Buttons
      * Do as same as checkboxs
   4. Toggle Buttons
      * Toast shows “On” or “Off”
   5. Spinners
      * On page 34
   6. Pickers
      * On page 36

Lab 2

2.1. Use Toast to show an input message of a plain Text after clicking a button.

2.2. Use AbsoluteLayout instead of LinearLayout.

2.3. Implement RelativeLayout on page 12. (Only design)