JEAS
Journal of Economics and Administrative Sciences
E-ISSN: 2148-1792 P- ISSN: 1302-2024
Volume 3, Supplement Issue 3
www.jeasweb.org

Designing an android-powered button to perform something

Marjan Faizabadi
Bachelor of Computer Engineering(Software), Khajeh Nasir eDin Tusi University of Technology(KNTU), Iran.

ABSTRACT

Android became widely accepted in a very short time as it was free. Credible companies making mobile phones, tablets and smart devices, i.e., gadgets have customized the latest Android version as provided by Google, and offered it along with their products. The Android operating system is architecturally a Software Stack, meaning it is a set of small interconnected applications that all run as a single operating system. To program via Android, one should have the knowledge of some concepts, be familiar with such concepts as object-oriented programming, and have experiences of working with Java, and of some capabilities of Android. One should also be familiar with coding skills to run Java codes on an Android application. In this research, an Android-assisted button is designed and its ability to do something is explored.

Keywords: Android, programming, Java, designing a button

Introduction

Android is the Google's operating system for mobile phones, tablets, and smart TVs, whose first version was launched in 2008. Google's goal was to produce an all-purpose operating system to be widely used by users. In 2005, by purchasing the small company, Android, Google began to produce smart operating systems.

After joining Google, Andy Rubin, the CEO of the said company, was appointed the head of the Google's Android project. Android is based on the Linux operating system and is open source and free.

The Android architecture allows it to be a set of small interconnected applications that all run as a single operating system. In the bottom layer of this system there is the central core, which is also the primary layer. The other segment of Android includes libraries, such that they tell the system how to treat the data. Android also includes Java libraries, which are critical for Android applications to run.

Part of Android is comprised of a virtual machine on which different operating systems are installed, preparing the operating system environment and its resources for use by other operating system applications.

The upper layer of the Android stack is the Application Framework. This layer of the operating system is the same layer that is openly provided to the programmers, allowing them to access mobile resources and the operating system.

The other layer of the Android Library stack is the operating system. This layer includes various instructions that direct the device how to deal with different data. For example, the Media Framework library contains information on the execution of various file, video and music file formats.

In fact, Android is said to includes a set of c/c ++ classes used by the Android system's components.

At the same level of the Android Stack is Android Runtime, which includes java libraries used to create Android applications and are completely critical to running them.

Android Runtime consists of 2 sections. The lower section of which is a virtual machine called Dalvik, which is tasked with running Java applications for Android. This virtual machine (Dalvik Executable) runs with a .dex extension, and the other section is the Core Library. Low-level library is for working with virtual machines. This library provides a Java core. If its structure is compared with Microsoft .net, Dalvik will be the same CLR and Core, the same BCL.

The highest layer, i.e., Applications built using the upper-level application (Application Framework), are installed and run on this operating system.

The Application layer is the highest level in the architecture of the Android operating system. In general, Android users are connected with this level and are mainly attracted to this level. Using these programs, users do their desired work; for example, making a call or browsing some web pages using these programs.

Developers and programmers produce the applications needed by in this section, and therefore developers are mostly concerned with the lower layer, i.e., Application Framework.

Each application consists of several parts:

1-Components (Activities, Services, Content Providers, Contact Receivers)

2. Manifest file

3. Resources

Android features

1- It supports all communication technologies (GSM, Bluetooth, WIFI, etc.)

2- It supports various media formats: Gif, Png, Jpeg, Mp3, Mpeg4, etc.

3- It supports SMS and MMS to send messages.

4- It supports SqlLite style databases.

5- It supports Multi Touch capability.

6- It supports Multi Tasking.

7- It supports various hardware (Camera, Sensors, GPS, etc.)

8. It supports Adobe Flash and Adobe AIR

Required Android programming software

For programming in Android, install several programs on the system, as the way these programs are installed and arranged is important, and should be as follows:

Written programs for Android are created with the .apk extension and can be installed and run on Android with the appropriate version.

JDK (Java Developer Kit)

Android SDK

IntellijIDEA

To install the Android SDK, first install the JDK software. It is quite easy to install this software. What needs to be done is to select the desired location and then to click Next and Finish in the end.

The next step is to install the Android SDK. To install this file, download and install the version related to the operating system (if you download the zip file, it suffices to unzip it at the desired location).

It is recommended that all the options remain in the default state, and finally click Finish with the "Start SDK Manager" option marked.

When SKD Manager is running, SDK settings and updates can be performed through the SDK Manager.

At last, install Intellij IDEA, as this software also keeps the default options and the Lunch option is activated for the program to immediately run.

Developing a Virtual Device or Emulator for Android

Sometimes, it is necessary to test the program while writing it; to do this, a virtual machine is used. To develop this machine, first run the SDK management program and select the virtual device from the left column.

Select the New option from the right column and name the virtual device in the window opening.

In the Target section, select the API version already downloaded and enter the volume of the memory card (SD card).

Enable the Snapshot option. This is very important, as it allows for a taking a snapshot of the last situation when the emulator stops.

To examine if the virtual machine works properly or not, click on the virtual machine created and select the Start option from the right column, then select the intended size for the emulator and click on Launch.

An emulator is seen in Figure 1.

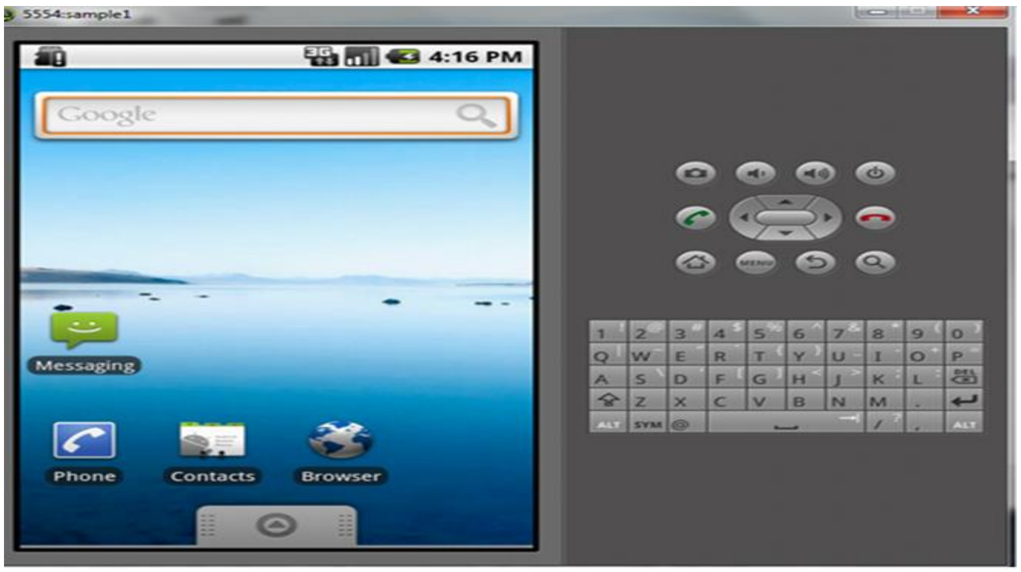


Figure 1: Emulator

Android project structure

To better acquaint with the way Android is structured, it is necessary to understand the following in order to have an overview of a project's general environment before developing it.

• Storage of IDE related files for running a project: IDEA
• Storage of additional files to be used in the project (e.g., text, xml, db, sound, etc,): Assets
• Compiled and run project files used in AVD or device: Bin-Out
• R.java class and classes built by the compiler. They must not be edited: Gen
• Other libraries used in the program: libs
• Internal and main project sources (layout, value, etc.) with several sub-folders: Res
Java Source Code (Programming): Src
• Project runtime settings: AndroidManifest.xml
• Android library in External Libraries.

How to create a new project

When installing and running the intellij idea software, a page opens. Click on the Create New Project option. Select the module from the left called Android from the Application Module.

Then specify the name of the project and in the Project Location section, specify the point where the projects are created, and in the Project SDK section, specify the AndroidSdk address, and in the Build Target section, specify the desired API Level, and then click Next.

In this section, specify the name of the application, the name of the package and that of Activity, and in the Target Device section, also use the emulator to test the program.

Then, upon pressing the Run button in the IDE, the desired program is installed on the system and can be used.

Before doing this project, some concepts need to be clearly defined:

Activity: It is actually the page seen in the Android program. Each program can be associated with one or more Activity, and each Activity has a life cycle. Each Activity contains one or more views, which are actually the UI elements programmers work with. For example Button, TextBox, TextView, ImageView, etc.

At first, when Activity is loaded, the sentence Hello World, that can be changed in the Text section. As in Figure 2, keyboard design can be placed in the Activity. Later, we discuss the way Views are placed.

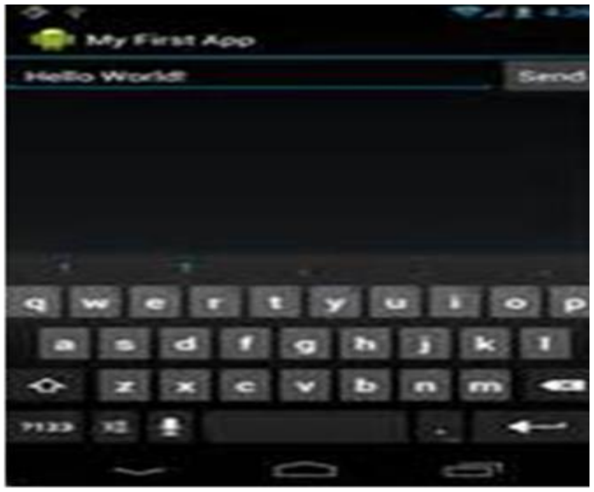
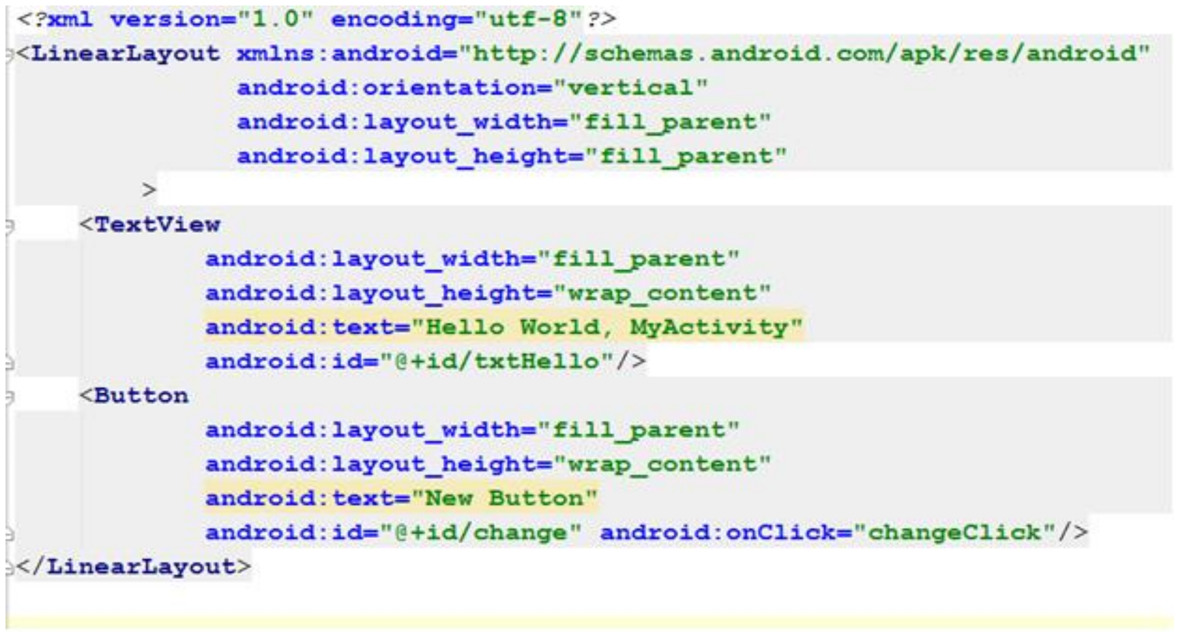


Figure 2: Activity

Xml structure

This is the same structure stated in Main. In the txt section, changes can be made to the Activity writing. As well, in class R an id can be assigned to new moving elements in this section. Other elements obtain their id by default when defined in this section.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:orientation="vertical"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
        >
    <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello World, MyActivity"
            android:id="@+id/txtHello"/>
    <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="New Button"
            android:id="@+id/change" android:onClick="changeClick"/>
</LinearLayout>
```

Figure 3: Xml structure

Java language

To do Android projects, one needs to learn the Java language and be acquainted with such concepts as object-orientation and if, else, switch.

To understand object-orientation, consider a computer which is considered an object. Now this computer is different from a personal computer (PC).

A personal computer is an object that can involve several other objects in itself, i.e, RAM, hard drive, etc. A personal computer has a set of features or attributes that distinguish it from other computers, e.g., it has a dual-core CPU. At the same time, the personal computer has a set of behaviors that are specific to that object. For example, it can play music, while a refrigerator cannot.

Now these objects we create must take root from a point where the concept of class is introduced. Classes are used to make objects. In fact, we define in class what features and behavior our object has. In fact, an object is defined as an instance of a class. To develop a class, we as programmers must consider four things:

Abstraction, Polymorphism, Inheritance, Encapsulation

Abstraction: Suppose you ask your friend to give you a glass of water. However, you do not mean to have a glass of water with a width of 10 mm and a length of 15 cm and the one which is brown on the table. This is also the case with programming. You need to have a general concept.

Encapsulation: A capsule holds the drug inside. In object-orientation, it also means a little more. This feature allows the program to display the features we intend. In reality, we want to minimize the dependencies among the various parts such that a small change leads to changes in other parts of the program.

Inheritance: Inheritance allows us to define a class that inherits some of its features from another class, instead of defining a new one. The class from which the features are inherited is called the Super Class and the inherited class, the Sub Class.

Polymorphysm: It denotes several forms. Consider the + sign. In this expression,9-2+7, with the sign + adding the two numbers together, but if we add the two words of a string kind together, the two are placed next to each other: Hello + world = helloworld

Variables:

All kinds of variables are available in Java. Each variable in Java is an object other than 3 categories: numbers, characters, boolean; characters are called premetives, implying that they are very fast, while the main variables are applied for faster programs and thus occupy less memory.

Defining a variable in the program

1- Data type: int

2- Variable name: my Integer

3- Initial value (optional)

Note: Each variable has a scope: inside the function

String myString = new string ();

And outside the function, which is considered inside the class, such as Field, which are variables that are defined outside the function.

Types of numerical variables

Premetives

Byte: 127, -128:

Byte myByte=127;

Short: 32000,-32000

Short myShort=32000;

Int: 2^32

IntmyInt=2000000000;

Long:2^64

Long myLong=123L;

Float:2^32
Float myFloat=123f;
Double:2^64
Double myDouble=132d;
Note: If an initial value is not given to our numbers, their value will be 0:
Byte myByte:

Converting numeric variables to each other:

Double A;
Float B;
Long C;
Int D;
Short E;
Byte F;
F=100;
A=F;

Converting a variable in a bottom-up form is correct, but vice versa is problematic.
A=100.od;
F=A; → F=(byte) A
Writing and displaying STRING:
One of the most basic display solutions in Java is to apply the following command:
System.out.println ();
Instead of parentheses, one can write any sentence we desire.

Java operators

Now we look at the addition and subtraction as well as multiplication and division operators in Java.
Int studentNumber = 121;
Here we have a variable with a value of 121. We want to add number 10 to the variable:
Int studentNumber = 121;
studentNumber = studentNumber + 10;
If we want to subtract 10 from this value:
Int studentNumber = 121;
studentNumber = studentNumber - 10;
In this example, we want to multiply the value of student number by a number:
Int studentNumber = 121;
studentNumber = studentNumber * 10;
Finally, to divide by 10, we do the following:
Int studentNumber = 121;
studentNumber = studentNumber / 10;
Adding 1 unit to the current value:
Int studentNumber = 121;
studentNumber++;
Subtracting 1 unit from the current value:
Int studentNumber = 121;
studentNumber--;

In this program, you see, before the variable is displayed, 1 unit is added to it, which is called Pre-incrementing:
System.out.println (++ studentNumber);

Conditional commands

Suppose we want to examine conditions in a program.

In fact, by using such commands, we want the program to assess if a condition is correct or not, and depending on whether our condition is correct or not, does the condition executes the command we want?

Here's how to write an if statement in Java:

```
public class If Class { }
public static void main (String[] args) { }
```

Switch command

The if and else commands are used when we have only two probabilities, but if we want to increase the probabilities, we use the switch, which is more appropriate.

```
public class MySwitchClass { }
public static void main (String[] args) { }
```

In fact, the switch structure is the same as if. The input parameter is written in parentheses and the conditional commands must be in {..}

Loop

A loop is a command that makes something constant to be defineed or repeated infinitely many times.

```
public class ForLoop {public static void main (String [] args) {for (intnumber = 1; number <= 10;
number ++) {System.out.println (number);
```

While loop

These types of loops are structurally different from "for" loops. In While, we need to first define our variable, then in While (), we consider our own condition which is the same as the Loop End Point.

We the define the value for increasing the variable inside {} placed against While ().

```
public class WhileLoop {}
public static void main (String [] args) {}
intnumber = 1; while (number <= 10) {}
System.out.println ("My Number Is:" + number);
```

Layout: Layout is one of the UI elements. Layout is like a container in which different views such as Button, TextView, EditText, etc. are included. Layout is a ViewGroup in which other Views are included. Also, the way elements are arranged in the Activity is specified by Layout.

In Android, it is possible to create Views by XML file and via Java code. All layouts are included in the/Res/Layout folder.

There are different Layouts:

1. Linear Layout
2. Relative Layout
3. Table Layout
4. Grid View
5. Tab Layout
6. List View

In the Layout, all the elements are included linearly. The elements can be included in Vertical and Horizontal forms. This features can be changed in the android: orientation section.

For example: <LinearLayoutandroid:orientation="horizontal"> .... </LinearLayout>

In a LinearLayout, as the name suggests, the elements on the screen are placed one line after another. This is while, these elements are placed next to each other vertically or horizontally, depending on the settings we do inside this layout.

As seen in the above code, the Value orientation method of this Layout is equivalent to a Vertical. In other words, if another TextView is added to this layout, for example, this new TextView will be placed under the previous TextView and both will be placed vertically next to each other.

In Vertical LinearLayout only one element can be placed in each row and in Horizontal LinearLayout there is a row of elements inside the page.

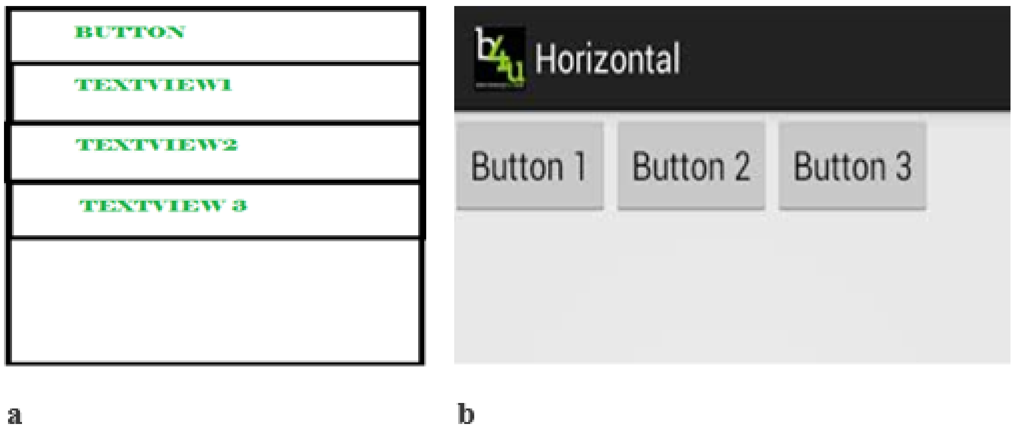


Figure 4: a) linear layout vertically, b) linear layout horizontally

Manifest file

Each application must have a file titled AndroidManifest.xml in its main directory, which is the same as Root Directory.

This file is responsible for giving general information about the application designed for the Android system. In other words, this file introduces the components we have used in our application to the Android operating system. On the other hand, this file It also involves the permissions the user must issue to install the program. Android is highly secure. One of the reasons why Android is secure and that the virus for the Android operating system is not genuine is the Manifest.xml file.

Some points to be followed in the manifest file are:

1- Defining any permissions the user needs to work with the application, such as access to the Internet or asking for access to user contact information.

2- Defining the minimum API access level of the program, depending on what kind of API the program applies.

3- Defining the hardware and software features used by the program, such as, camera, Bluetooth and multi-touch screen

4- API libraries used in the program, such as using the Google Maps library and other items.

5- The most important responsibility of the manifest is to inform the Android system of the program components.

Component capability

The capabilities of the intended component are included in the <Intent-Filter>. When a component is introduced in the Manifest program, one can also optionally introduce the Intent-filter that displays the program's capabilities. Thus, by doing so, we enable the program to run the program and to use the program result in case another program needs to use the program's capabilities. To introduce Intent-filter to a component, the <intent-filter> element must be stated as a subgroup in the component introduction section:

For example, an e-mail application with an Activity to send an e-mail may have a filter Internet defined in the Manifest that responds to Intent send (to send e-mail). Now an Activity in the program can create an Intent and request the Send (ACTION_SEND); when the start Activity () command is issued, and when it checks the Internet with the available filters, it reaches the email program and runs it.

Requirements used in applications

Various mobile phones have been developed powered by Android, but not all of them have the same features or capabilities. In order to prevent the installation of applications on phones lacking the minimum requirements of the application, it is very important to clearly mention application requirements so that only mobile phones that have the minimum requirements for the application can install the application. This is done as the required hardware and of course the software in the Manifest file are introduced. Many of the definitions are just for information and are not read by the system, but they are read in external services such as Android Market so that applications are filtered when users search the market.

The following are very important features to consider when designing and programming:

Screen size and density

To classify mobile phones based on the type of the screen, Android defines two features for each mobile phones: size of the screen (physical dimensions of the screen) and density of the screen (physical density of pixels on the screen marked by dpi). To simplify different types of pages introduced, the Android system has divided the pages into different groups to be selected more easily.

Classification based om screen size: small, normal, large, and extra large

Classification based on page density: low density, medium density, high density and extra high density

On a default mode, the program is compatible with all types of pages and various densities, with the reason being that the Android system performs the appropriate settings for the user interface (UI layout) and image resources of the program. However, the program should be designed and selected based on a specific page size with images being based on page density. In the Manifest file, the supported screen size should be mentioned using the <supports-screens> element.

Input configurations: Different tools have introduced different inputs to receive information from the user. Some examples of different inputs are hardware keyboard, trackball, and five-way navigation pad. If the program supports a specific type of input, it is introduced in the Manifest file using the <uses-configuration> element. However, the program rarely needs a specific input.

Device features: There are various hardware and software features on different Android devices, such as camera, light sensor, Bluetooth, a specific version of OpenGL or touch screen. One should never assume that there is only one specific feature on all different tools, so specific features used in a program should be introduced in the Manifest file using the <uses-feature> element.

Platform version: Most of the time, different Android tools make use of different versions, such as Android 1.6 and Android 2.3. Newer versions use APIs that not existing in older versions. In order to introduce the accessible APIs, each version of the platform has a specific API level. One should specify the minimum API level using the <uses-sdk> element.

Design process

First a new project was created using the method explained. Then the desired emulator was constructed. This emulator runs automatically when the run option is selected. One should note that the codes should be written between the Set contentview and the last two brackets. When a project is written and opened, the graphic mode of the project is also visible.

Different xml parts

The Main.xml section is comprised of two parts, design and text. Using design, one can develop various elements such as Textview without the need to write code. In other words, using a pallete positioned in this part, different types of Views are provided. Select a button, textview, ratingbar and use drag & drop to place it in the page that looks like a small emulator with hello world written on it. The text section is mostly used for id.

This code was written to design a simple button in the Java section of the project and under Setcontentview:

```
Button btn1=(Button) findviewbyId(R.id.button);
```

Define a variable of Button kind that has an id button as it exists automatically and state it to display text. This is done by the Settext function. It should be mentioned that it is not necessary to write the name of the function completely and by default:

```
btn1.Settext=("salam")
```

The code you see below is the complete code written as a project with its description in the end. We start writing the code from the Setcontentview section:

```
Setcontentview(R.layout.main);
ll=(linearlayout)findviewById(R.id.linear);
btn2=new Button (this);
btn2.setText("hi");
Button btn1=(Button)findviewById(R.Id.button);
Btn1.setText("salam");
Btn1.setonClicklistener(new view.onClicklistener(){
@override
Public viodonclick(View V){
Textview.txt=(Textview)findviewById(R.id.textview);
Txt.settext("11111111111");
Txt.setBackgroundColor(Color.RED);
Txt.setTextColor(Color.YELLOW);
ll.addview(btn2);
```

Press the run button to compile and run the program. The emulator made is Started in this section and displays the result.

Explanation

First, define a moving button, and an id is defined for it in Text of the main.xml section (android.id=@+id/linear); then call the relevant layout and define a new button and ask it to display the text intended. Then we call the button we defined above in this section and give it the relevant command. Here we can design the first button in such a way that this text is displayed by clicking on it and for this purpose the onclicklistener function is used. Then we create a variable named Textview and give it a value and then call it in the Onclick function for the text to be printed inside txt. Using setbackgroundcolor, one can color the background of the text and the text color with settextcolor. Finally, one can add a new button to the layout using addview.

Figure 5 illustrates the program running:

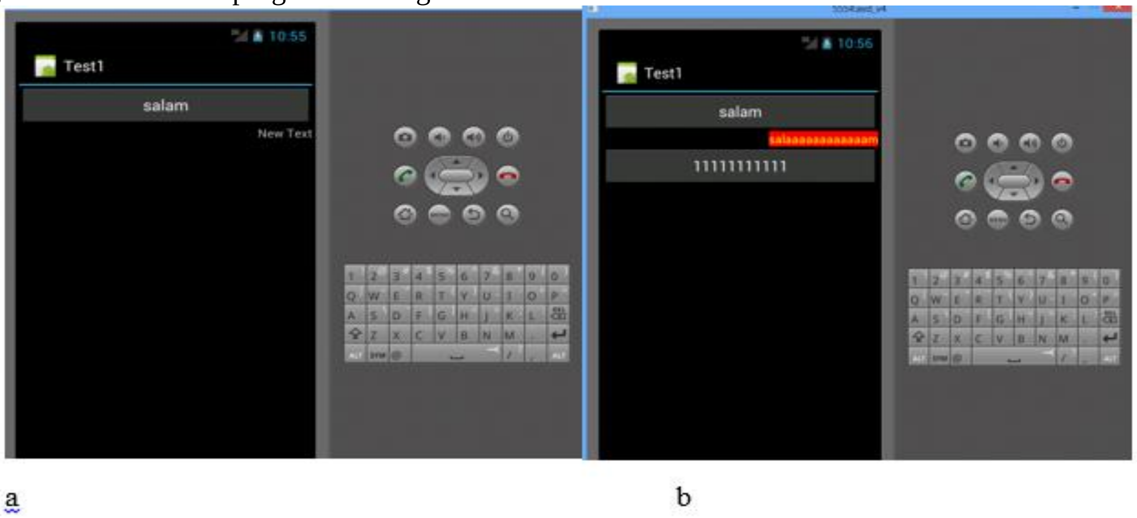


Figure 5: a) Running the designed program, b) Colored display of the designed button

Conclusion

Android is a language whose prerequisite is Java. One needs to learn the concept of object-oriented before understanding it. The word Android means humanoid robot. The fact that Android was free helped it be largely applied and spread in a very short time. Reputable companies that produce mobile phones, tablets and gadgets have customized the Android operating system according to the latest version provided by Google, providing it with their products. Android phones provide such features which are easy to use and economic both in terms of prices for those who have a limited budget, and in terms of specifications for those who are interested in technology. Most importantly, if the way the operating system on the phone is not attractive, it can be customized in a way you like. Its hardware features depend on the device you have bought; however, the Android operating system provides the necessary support for cameras, GPS location, Bluetooth, accelerometer, compass, gyroscope, etc.

The Android operating system is characterized by its software components such as animated wallpapers and widgets, notification bars for important apps, typing by voice, and application integration that allows you to easily access information through networks. Social, or other communication methods such as Bluetooth are also some of its features.

Compared to the mobile devices on which an operating system is installed, Android has the biggest number of users. This provided an exceptional opportunity for programmers to write programs to be used by hundreds of millions of Android users. This system may have a complicated function for those who are not familiar with Android programming, but the project teaches the user step by step how to start working in this field.

This language, which is one of the advanced disadvantages, provides us with many possibilities we saw in this project such that by making a button, it can be expanded in different ways, or that one can add a fixed or moving button to it. The project I presented yields such results as the ability to control the computer using Android, which is included in programming sockets. Using the interface between the computer and the Android phone and the button design, various tasks can be done, such as turning off a computer and opening the files; this is while, in the future, improvements can be achieved by expanding such systems, all of which are possible by Android.

References

[1] SDK software
[2] http://www.oracle.com/technetwork/java/javase/downloads/index.html
[3] Android SDK software
[4] http://developer.android.com/sdk/index.html
[5] Jet Brains IntelliJ IDEA software
[6] http://iranisoft.mihanblog.com/post/8872