ANDRIOUS (HTTPS://ANDRIOUS.COM/)

Think, Imagine and create yourself

JAVA (Https://Andrious.Com/Category/Java)

How to Create Simple Calculator in Android Studio Java

Israfil Mahmud Raju (Https://Andrious.Com/Author/Admin) / June 22, 2020

In this Tutorial we will learn how to create a simple calculator in android studio.Lets start

How to Create Simple Calculator in Android Studio Java

Download Now! (https://andrious.com/?smd_process_download=1&download_id=48) 24 Downloads

Step 1: **Building the user interface**

First we design our user interface using TextView, Button and ImageButton. TextView are used to display user input and result, Button and ImageButton are Used to get user Input.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="15dp"
    android:weightSum="3"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:orientation="vertical"
        android:layout_weight="1">
        <TextView
            android:id="@+id/display"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:padding="5dp"
```

Step 2: Setup Resource file

1. Right click on **drawable -> new -> Image Asset** Choose Icon Type **Action Bar and Tab Icons** then click Click Art Image and select Which Icon Name is **close** then press ok and then name it **back** and change color and use this color code **D81B60** finaly click **next->Finish**.

2. Right click on **drawable -> new -> Drawable Resource File** then name it **ripple_bg** and click Ok. You can see **ripple_bg.xml** file under your **drawable** folder. Copy the bellow code and paste it into your **ripple_bg.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<ripple xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:color="@color/colorPrimaryDark"
    tools:targetApi="lollipop">
    <item android:id="@android:id/mask">
        <shape android:shape="rectangle">
            <solid android:color="@color/colorPrimaryDark" />
            <corners android:radius="0dp" />
        </shape>
    </item>
    <item android:id="@android:id/background">
        <shape android:shape="rectangle">
            <gradient
                android:angle="90"
                android:endColor="@android:color/transparent"
                android:startColor="@android:color/transparent"
                android:type="linear" />
            <corners android:radius="0dp" />
        </shape>
    </item>
```

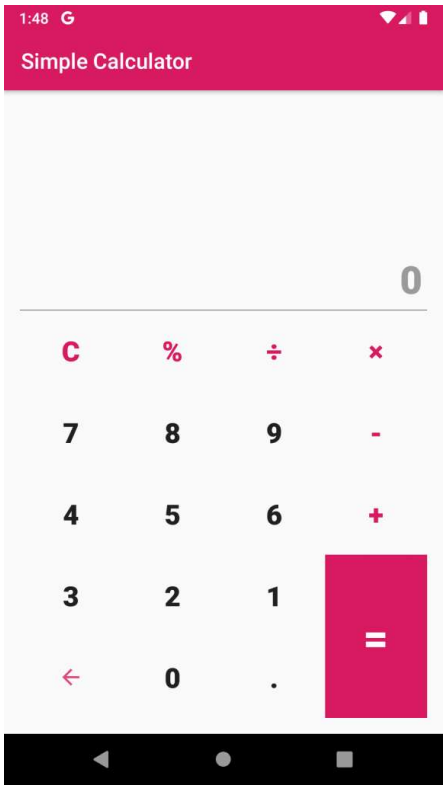
3. Right click on **drawable -> new -> Drawable Resource File** then name it *ripple_bg* and click Ok. You can see **ripple_btn.xml** file under your **drawable** folder. Copy the bellow code and paste it into your **ripple_btn.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<ripple xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:color="#88000000"
    tools:targetApi="lollipop">
    <item android:id="@android:id/mask">
        <shape android:shape="rectangle">
            <solid android:color="#88000000" />
            <corners android:radius="0dp" />
        </shape>
    </item>
    <item android:id="@android:id/background">
        <shape android:shape="rectangle">
            <gradient
                android:angle="90"
                android:endColor="@android:color/transparent"
                android:startColor="@android:color/transparent"
                android:type="linear" />
            <corners android:radius="0dp" />
        </shape>
    </item>
```

4. Right click on **drawable -> new -> Drawable Resource File** then name it **ripple_bg** and click Ok. You can see **ripple_equal_btn.xml** file under your **drawable** folder. Copy the bellow code and paste it into your **ripple_equal_btn.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<ripple xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:color="#000000"
    tools:targetApi="lollipop">
    <item android:id="@android:id/mask">
        <shape android:shape="rectangle">
            <solid android:color="#000000" />
            <corners android:radius="0dp" />
        </shape>
    </item>
    <item android:id="@android:id/background">
        <shape android:shape="rectangle">
            <gradient
                android:angle="90"
                android:endColor="@color/colorPrimaryDark"
                android:startColor="@color/colorPrimaryDark"
                android:type="linear" />
            <corners android:radius="0dp" />
        </shape>
    </item>
```

Now your design look like this



Step 3: Coding the Functionality

In your Java file, which is almost always **MainActivity.java** by default, head on over to the **MainActivity extends AppCompatActivity** class. Copy bellow code and paste into your **MainActivity.java** file. But without changing your project package name **[package com.andrious.simplecalculator;]**

```
import android.content.res.Configuration;
import android.graphics.Color;
import android.os.Bundle;
import android.util.TypedValue;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    Double result_value = 0.0;
    Boolean checker = true;
    int orientation = 0;

    TextView result, display;
    Button button_0,button_1,button_2,button_3,button_4,button_5,button_6,button_7,butto
n_8,button_9;
    Button button_dot, button_clear, button_modulas, button_add, button_sub, button_mult
i, button_div, button_equal;
```

Now every thing is done. Run and enjoy the calculator app.

Step 4: Function description

Fetching values from the elements into the working of our app

First declare necessary variable inside our class

```
Double result_value = 0.0;
Boolean checker = true;
int orientation = 0;
TextView result, display;
Button button_0,button_1,button_2,button_3,button_4,button_5,button_6,button_7,button_8,
button_9;
Button button_dot, button_clear, button_modulas, button_add, button_sub, button_multi, b
utton_div, button_equal;
ImageButton button_back;
```

Inside the **onCreate** function, we will fetch all values and assign them to our objects. We do this by the following code


```
button_0  = findViewById(R.id.button_0);
```

Here, we are fetching the value of 'button_0', which is the ID of our button zero, and storing it in the object 'button_0' that we created earlier. Though they are named the same, Android Studio can identify the correct one. But if it gets confusing for you, feel free to give either one of them any other names.

We do the same with our TextView.

```
result  = findViewById(R.id.result);
```

Changing the TextView value by pressing the buttons.

```
button_0.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v){
                display.setText(display.getText().toString() + "0");
                text_size_color_controller();
                checker = false;
                entry_controller(checker);
            }
        });
```

We use OnClickListener for this part of the program. This is what happens when you push zero. The TextView, first already fetches the value that it is already displaying, null in this case, and then adds 0 to it.

Text Font size control when change the phone orientation.

In onCreate method we initialize it and get the device current orientation.

```
orientation = getResources().getConfiguration().orientation;
```

Finally use conditional statement in button_equal onClick function based on device orientation.

```
if (orientation == Configuration.ORIENTATION_LANDSCAPE) {
            // In landscape
            display.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30F);
            result.setTextSize(TypedValue.COMPLEX_UNIT_DIP,20F);
        } else {
            // In portrait
            display.setTextSize(TypedValue.COMPLEX_UNIT_DIP,36F);
            result.setTextSize(TypedValue.COMPLEX_UNIT_DIP,26F);
        }
```

Text size and color control function

```
public Void text_size_color_controller() {
    display.setTextColor(Color.parseColor("#000000"));
    result.setTextColor(Color.parseColor("#bdbdbd"));
    if (orientation == Configuration.ORIENTATION_LANDSCAPE) {
        // In landscape
        display.setTextSize(TypedValue.COMPLEX_UNIT_DIP,30F);
        result.setTextSize(TypedValue.COMPLEX_UNIT_DIP,20F);
    } else {
        // In portrait
        display.setTextSize(TypedValue.COMPLEX_UNIT_DIP,36F);
        result.setTextSize(TypedValue.COMPLEX_UNIT_DIP,26F);
    }
    button_modulas.setEnabled(false);
    return null;
}
```

This function control text size and color when equal button click. This function also check the device orientation for controlling the text size.

User Entry Control Function

```
public Void entry_controller(Boolean b) {
    if(b){
        button_add.setEnabled(false);
        button_sub.setEnabled(false);
        button_multi.setEnabled(false);
        button_div.setEnabled(false);
        button_equal.setEnabled(false);
    }else{
        button_add.setEnabled(true);
        button_sub.setEnabled(true);
        button_multi.setEnabled(true);
        button_div.setEnabled(true);
        button_equal.setEnabled(true);
    }
    return null;
}
```

This function work like this if user don't write any number but click addition or subtraction button, this function handle this type of error. If user don't type any number then all the operation button are disabled. Other exception is if user write like this **2+2+** and press enter then app will be crashed. This function also handle this. If user input last value is any operator then equal button automatically disabled. It also handle if user click any operator button then all the operator buttons are automatically disabled. If user need to change the operator click the back button then automatically enable all the operator buttons.

Calculation Function

```
public static double calculation(final String str) {
    return new Object() {
        int pos = -1, ch;
        void nextChar() {
            ch = (++pos < str.length()) ? str.charAt(pos) : -1;
        }
        boolean eat(int charToEat) {
            while (ch == ' ') nextChar();
            if (ch == charToEat) {
                nextChar();
                return true;
            }
            return false;
        }
        double parse() {
            nextChar();
            double x = parseExpression();
            if (pos < str.length()) throw new RuntimeException("Unexpected: " + (char)c
h);
            return x;
        }
```

When user click equal button then equal button onClickListener function get the all user input as string then pass it int this calculation function. This function phrase the string and calculate the value and return result as decimal data type.

Related Posts

(https://andrious.com/java/implementing-navigation-drawer-in-android-app-using-java.html)

Implementing Navigation Drawer in Android App using Java (https://andrious.com/java/implementing-navigation-drawer-in-android-app-using-java.html)

July 7, 2020

(https://andrious.com/java/create-a-note-book-app-using-java-in-android-studio.html)

Create a Note Book App using Java in Android Studio (https://andrious.com/java/create-a-note-book-app-using-java-in-android-studio.html)

July 2, 2020

(https://andrious.com/java/list-view-in-android-studio-java.html)

ListView in Android Studio Java (https://andrious.com/java/list-view-in-android-studio-java.html)

June 24, 2020

About Israfil Mahmud Raju

I am Israfil Mahmud Raju. My Passion is Android App Developing. This site is a way for me to showcase all of my works, projects in one location.

View all posts by Israfil Mahmud Raju → (https://andrious.com/author/admin)

How to Create Simple Calculator in Android Studio Kotlin (https://andrious.com/kotlin/how-to-create-simple-calculator-in-android-studio-kotlin.html)

ListView in Android Studio Java (https://andrious.com/java/list-view-in-android-studio-java.html)

Search … Search

ISRAFIL MAHMUD RAJU

RECENT POSTS

Implementing Navigation Drawer in Android App using Java (https://andrious.com/java/implementing-navigation-drawer-in-android-app-using-java.html)

Implementing Navigation Drawer in Android App using Kotlin (https://andrious.com/kotlin/implementing-navigation-drawer-in-android-app-using-kotlin.html)

Create a Note Book App using kotlin in Android Studio (https://andrious.com/kotlin/create-a-note-book-app-using-kotlin-in-android-studio.html)

Create a Note Book App using Java in Android Studio (https://andrious.com/java/create-a-note-book-app-using-java-in-android-studio.html)

ListView in Android Studio Kotlin (https://andrious.com/kotlin/listview-in-android-studio-kotlin.html)

CATEGORIES

JAVA (https://andrious.com/category/java)

KOTLIN (https://andrious.com/category/kotlin)