SIMPLE CALCULATOR

Mini Project

By

D MADHUSHALNE

On the Month of November, Year 2023

Description of the project

Creating a Simple Calculator App with XML code, and Additional Features

Introduction: In this extended project, we will design and build a simple calculator app using XML for the user interface (UI) and Kotlin for the logic. In addition to the basic calculator functionalities, we will incorporate a menu bar, a search box text view, and a button. The goal is to create a user-friendly calculator that can perform arithmetic operations while allowing users to search for results and perform other actions. This project will help you understand how to use XML and Kotlin to create a versatile app with multiple features.

Page 1: XML Design for the Calculator

Layout Structure: We will begin by designing the layout for our calculator using XML. The calculator UI will consist of a display area at the top, a menu bar, a search box text view, a button, and a grid of buttons for the numeric digits and operators. We will use a Relative Layout as the root layout to position elements.

Display Area: The display area will be a Text View that will show the input expression and the result of the calculation. We will define its attributes such as text size, alignment, and padding to make it visually appealing.

Menu Bar: We will incorporate a menu bar at the top, allowing users to access various functions and settings, such as changing the calculator mode, theme, or accessing help and about information.

Search Box Text View: A search box text view will be placed below the display area, enabling users to enter a query or search for specific calculations, equations, or results.

Button: A button with a label like 'Search' or 'Go' will be placed next to the search box text view to initiate the search when clicked.

Numeric Buttons: We will create buttons for digits 0-9 and arrange them in a grid using a Grid Layout. Each button will have a unique ID and display the respective digit.

Operator Buttons: For the arithmetic operations (addition, subtraction, multiplication, and division), we will create buttons with appropriate symbols (+, -, *, /). These buttons will also have unique IDs.

Other Buttons: We will include additional buttons for functions like clearing the input and calculating the result. These buttons will be labelled with 'C' for clearing and '=' for calculating.

Button Layout: We will use a combination of Linear Layouts and Grid Layouts to arrange the numeric and operator buttons in an organized and user-friendly manner.

Page 2: Kotlin Logic for the Calculator

Button Click Listeners: To make the calculator functional, we need to add click listeners to the buttons. In Kotlin, we will define these listeners for each button.

Input Handling: We will create a string variable to store the input expression and initialize it as an empty string. When the numeric or operator buttons are clicked, we will append the corresponding value to this string.

Calculation Logic: For the '=' button, we will implement the calculation logic. We will use Kotlin's built-in libraries to parse and evaluate the expression and display the result in the Text View.

Clear Function: The 'C' button will clear the input by setting the input string to an empty string.

Search Function: When the 'Search' or 'Go' button is clicked, we will implement a search algorithm to find and display results related to the query entered in the search box text view.

UI Updates: After each button click or search, we will update the display area to show the current input expression, the result, or the search query results.

Error Handling: We will add error handling to catch any exceptions that may occur during the calculation, such as division by zero or invalid expressions, and display appropriate error messages. Additionally, we will handle errors in the search functionality.

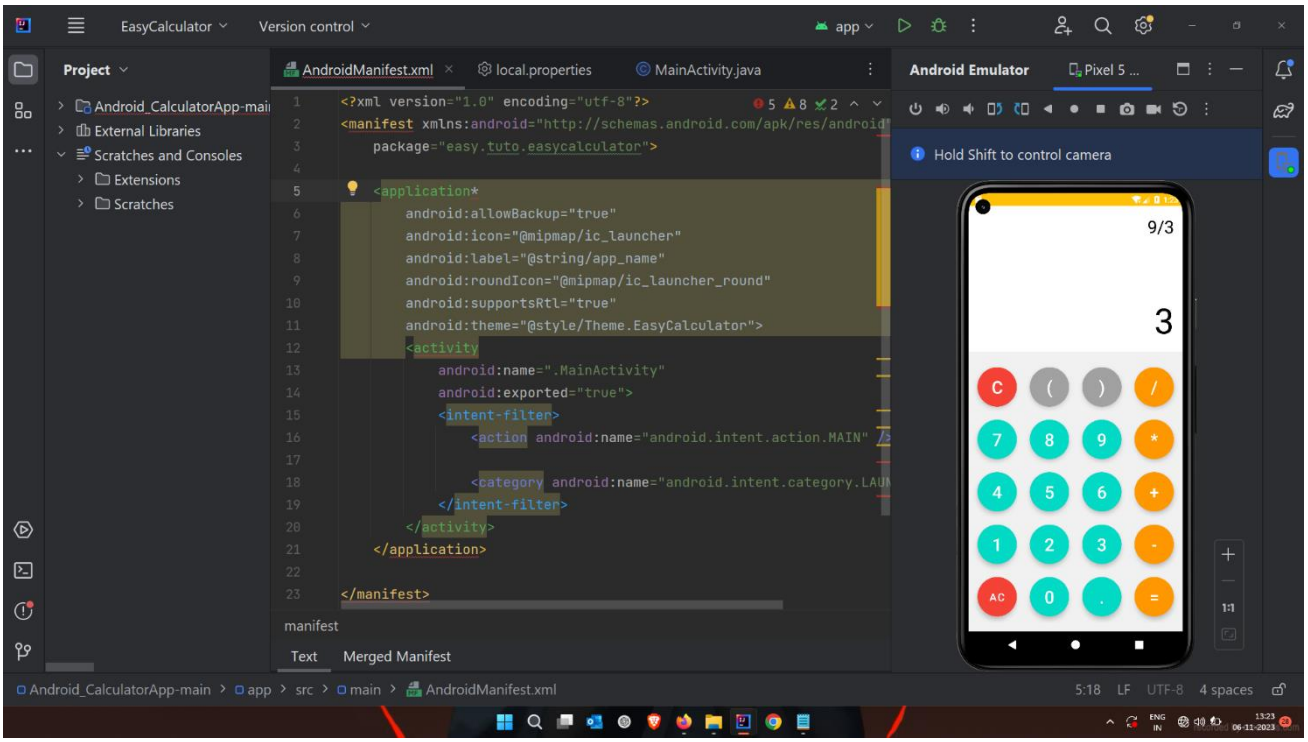
Menu Functionality: The menu bar will provide access to various features, allowing users to change the calculator mode, theme, and access additional information. We will implement the necessary functionality for these menu options.

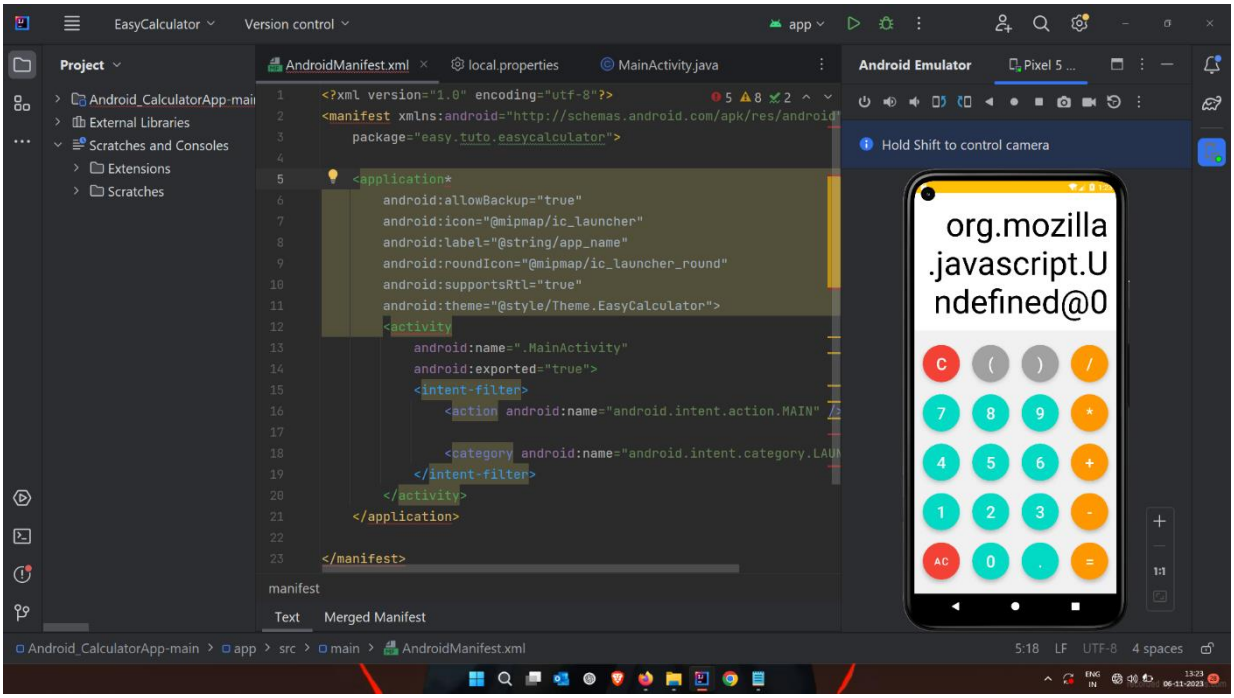
Testing: We will thoroughly test the calculator and search functionality to ensure that they perform correctly and handle various user interactions and edge cases gracefully.

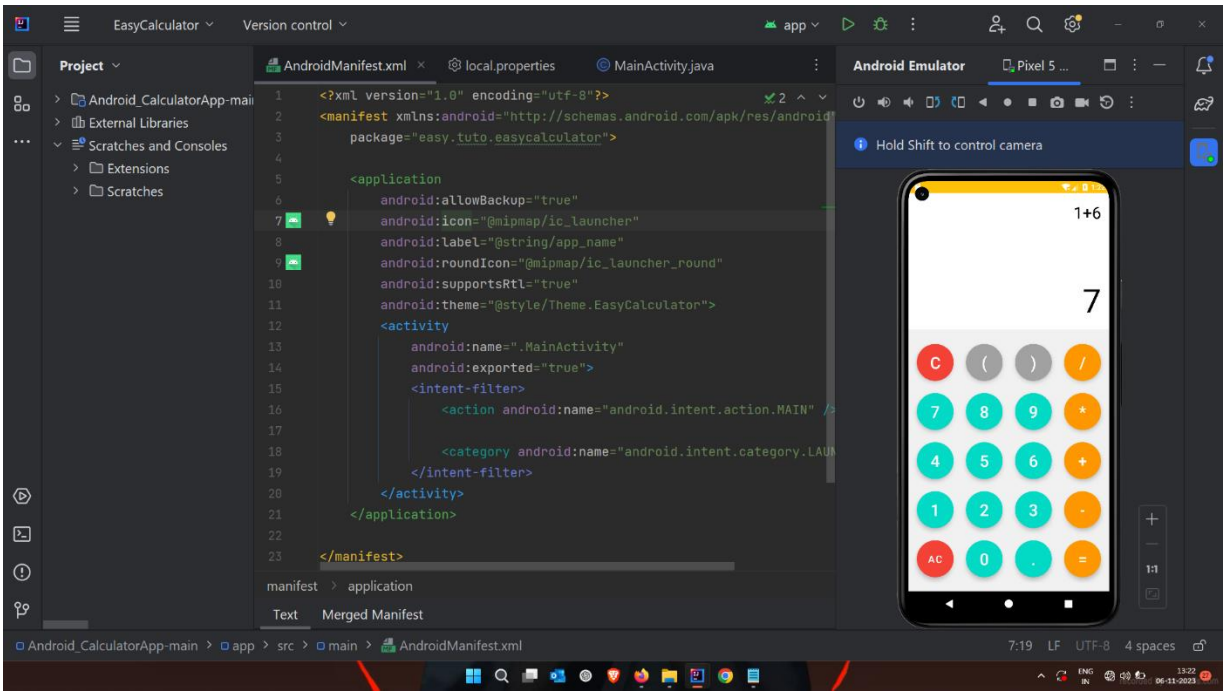
User-Friendly Features: To enhance the user experience, we can implement additional features like decimal point input, backspace functionality, and keyboard support for the search box text view. We will also add a feature to switch between different calculator modes, such as a scientific mode, and change the calculator's theme.

Conclusion: In this extended project, we have designed a versatile calculator app using XML for the user interface and Kotlin for the logic. The XML layout defines the structure of the calculator, including the display area, menu bar, search box text view, button, numeric buttons, operator buttons, and other controls. The Kotlin logic handles button clicks, input handling, calculation, clearing, error handling, and the search functionality. This project serves as an excellent example of building an interactive calculator app with multiple features, catering to user needs beyond basic calculations.

Screen shot of project




EasyCalculator
Version control
app
Project
Android_CalculatorApp-mai
External Libraries
Scratches and Consoles
Extensions
Scratches
AndroidManifest.xml
local.properties
MainActivity.java
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="easy.tuto.easycalculator">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.EasyCalculator">
<activity
android:name=".MainActivity"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAU
</intent-filter>
</activity>
</application>
</manifest>
manifest
Text
Merged Manifest
Android Emulator
Pixel 5 ...
Hold Shift to control camera
org.mozilla
.javascript.U
ndefined@0
C ( ) /
7 8 9 *
4 5 6 +
1 2 3 -
AC 0 . =
Android_CalculatorApp-main
app
src
main
AndroidManifest.xml
5:18
LF
UTF-8
4 spaces


EasyCalculator
Version control
app
Project
Android_CalculatorApp-mai
External Libraries
Scratches and Consoles
Extensions
Scratches
AndroidManifest.xml
local.properties
MainActivity.java
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="easy.tuto.easycalculator">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.EasyCalculator">
<activity
android:name=".MainActivity"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAU
</intent-filter>
</activity>
</application>
</manifest>
manifest
application
Text
Merged Manifest
Android Emulator
Pixel 5 ...
Hold Shift to control camera
1+6
7
C ( ) /
7 8 9 *
4 5 6 +
1 2 3 -
AC 0 . =
Android_CalculatorApp-main
app
src
main
AndroidManifest.xml
7:19
LF
UTF-8
4 spaces

Main code

```
package easy.tuto.easycalculator;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import com.google.android.material.button.MaterialButton;
import org.mozilla.javascript.Context;
import org.mozilla.javascript.Scriptable;
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    TextView resultTv,solutionTv;
    MaterialButton buttonC,buttonBrackOpen,buttonBrackClose;
    MaterialButton buttonDivide,buttonMultiply,buttonPlus,buttonMinus,buttonEquals;
    MaterialButton button0,button1,button2,button3,button4,button5,button6,button7,button8,button9;
    MaterialButton buttonAC,buttonDot;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
       resultTv = findViewById(R.id.result_tv);
       solutionTv = findViewById(R.id.solution_tv);
       assignId(buttonC,R.id.button_c);
       assignId(buttonBrackOpen,R.id.button_open_bracket);
       assignId(buttonBrackClose,R.id.button_close_bracket);
       assignId(buttonDivide,R.id.button_divide);
       assignId(buttonMultiply,R.id.button_multiply);
       assignId(buttonPlus,R.id.button_plus);
       assignId(buttonMinus,R.id.button_minus);
       assignId(buttonEquals,R.id.button_equals);
       assignId(button0,R.id.button_0);
       assignId(button1,R.id.button_1);
       assignId(button2,R.id.button_2);
```

```
    assignId(button3,R.id.button_3);
    assignId(button4,R.id.button_4);
    assignId(button5,R.id.button_5);
    assignId(button6,R.id.button_6);
    assignId(button7,R.id.button_7);
    assignId(button8,R.id.button_8);
    assignId(button9,R.id.button_9);
    assignId(buttonAC,R.id.button_ac);
    assignId(buttonDot,R.id.button_dot);
}

void assignId(MaterialButton btn,int id){
    btn = findViewById(id);
    btn.setOnClickListener(this);
}

@Override
public void onClick(View view) {
    MaterialButton button =(MaterialButton) view;
    String buttonText = button.getText().toString();
    String dataToCalculate = solutionTv.getText().toString();

    if(buttonText.equals("AC")){
        solutionTv.setText("");
        resultTv.setText("0");
        return;
    }
    if(buttonText.equals("=")){
        solutionTv.setText(resultTv.getText());
        return;
    }
    if(buttonText.equals("C")){
```

```
            dataToCalculate = dataToCalculate.substring(0,dataToCalculate.length()-1);
        }else{
            dataToCalculate = dataToCalculate+buttonText;
        }
        solutionTv.setText(dataToCalculate);

        String finalResult = getResult(dataToCalculate);

        if(!finalResult.equals("Err")){
            resultTv.setText(finalResult);
        }
    }

    String getResult(String data){
        try{
            Context context  = Context.enter();
            context.setOptimizationLevel(-1);
            Scriptable scriptable = context.initStandardObjects();
            String finalResult =  context.evaluateString(scriptable,data,"Javascript",1,null).toString();
            if(finalResult.endsWith(".0")){
                finalResult = finalResult.replace(".0","");
            }
            return finalResult;
        }catch (Exception e){
            return "Err";
        }
    }

}
```

Code: XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/solution_tv"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@id/result_tv"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:textAlignment="textEnd"
        android:textColor="@color/black"
        android:textSize="32dp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/result_tv"
        android:text="0"
        android:textSize="64dp"
        android:textAlignment="textEnd"
        android:textColor="@color/black"
```

```
    android:layout_above="@id/buttons_layout"
    android:layout_margin="16dp" />


<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_alignParentBottom="true"
    android:background="#F1F1F1"
    android:paddingVertical="16dp"
    android:id="@+id/buttons_layout">

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

        <com.google.android.material.button.MaterialButton
            android:layout_width="72dp"
            android:layout_height="72dp"
            app:cornerRadius="36dp"
            style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
            android:textSize="32dp"
            android:textColor="@color/white"
            android:layout_margin="12dp"
            android:id="@+id/button_c"
            android:backgroundTint="#F44336"
            android:text="C" />
```

```
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_open_bracket"
    android:backgroundTint="#A1A1A1"
    android:text="(" />
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:backgroundTint="#A1A1A1"
    android:id="@+id/button_close_bracket"
    android:text=")" />
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
```

```
        android:id="@+id/button_divide"
        android:backgroundTint="#FF9800"
        android:text="/" />

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

    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_7"
        android:text="7" />
    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_8"
```

```
        android:text="8" />
    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_9"
        android:text="9" />
    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_multiply"
        android:backgroundTint="#FF9800"
        android:text="*" />

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">
```

```
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_4"
    android:text="4" />
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_5"
    android:text="5" />
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_6"
    android:text="6" />
```

```
    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_plus"
        android:backgroundTint="#FF9800"
        android:text="+" />

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

    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_1"
        android:text="1" />
    <com.google.android.material.button.MaterialButton
```

```
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_2"
    android:text="2" />
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_3"
    android:text="3" />
<com.google.android.material.button.MaterialButton
    android:layout_width="72dp"
    android:layout_height="72dp"
    app:cornerRadius="36dp"
    style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
    android:textSize="32dp"
    android:textColor="@color/white"
    android:layout_margin="12dp"
    android:id="@+id/button_minus"
    android:backgroundTint="#FF9800"
    android:text="-" />
```

```
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">

    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="20dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_ac"
        android:backgroundTint="#F44336"
        android:text="AC" />
    <com.google.android.material.button.MaterialButton
        android:layout_width="72dp"
        android:layout_height="72dp"
        app:cornerRadius="36dp"
        style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
        android:textSize="32dp"
        android:textColor="@color/white"
        android:layout_margin="12dp"
        android:id="@+id/button_0"
        android:text="0" />
    <com.google.android.material.button.MaterialButton
```

```
            android:layout_width="72dp"
            android:layout_height="72dp"
            app:cornerRadius="36dp"
            style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
            android:textSize="32dp"
            android:textColor="@color/white"
            android:layout_margin="12dp"
            android:id="@+id/button_dot"
            android:text="." />
        <com.google.android.material.button.MaterialButton
            android:layout_width="72dp"
            android:layout_height="72dp"
            app:cornerRadius="36dp"
            style="@style/Widget.MaterialComponents.ExtendedFloatingActionButton"
            android:textSize="32dp"
            android:textColor="@color/white"
            android:layout_margin="12dp"
            android:id="@+id/button_equals"
            android:backgroundTint="#FF9800"
            android:text="=" />

    </LinearLayout>
  </LinearLayout>
</RelativeLayout>
```