

Ex. No. : 01A

Date : 31 Dec 2025

Register No. : 231701042

Name : Pugazhendhi B

GUI Components

Aim

Develop an application to change the font and color of the text and display toast message when the user presses the button.

Procedure:

Create a new Android project in Android Studio:

- Open Android Studio → Select “New Project” → Choose “Empty Activity”
→ Set Project Name and Package Name → Finish.

2. Design the layout:

- Open activity_main.xml.
- Use a LinearLayout with vertical orientation.
- Add a TextView to display the text.
- Add three Button elements:
 - Change Font Size
 - Change Font Color
 - Change Background

3. Configure TextView:

- Set android:text to the desired message, e.g., “★ THALAPATHY ★”.
- Align the text to the center using android:gravity="center".

4. Configure Buttons:

- Set appropriate android:id for each button.
- Set android:text to describe the action (Font Size, Font Color, Background).

5. Modify AndroidManifest.xml:

- Ensure the main activity is declared with the MAIN action and LAUNCHER category.
- Confirm the application theme and other attributes are set.

6. Write the MainActivity code:

- In MainActivity.kt, get references to all views (TextView, Buttons, LinearLayout) using findViewById.
- Implement **click listeners** for each button:
 - **Font Size Button:** Increment font size on each click (cycling back after a limit).
 - **Font Color Button:** Change the text color in a loop through multiple colors.
 - **Background Button:** Change the background color of the main layout in a loop through multiple colors.
- Optional: Display a **Toast message** each time a button is pressed to indicate the action performed.

7. Run the application:

- Connect an Android device or start an emulator.
- Click on each button to observe the changes:
 - Text size changes on “Change Font Size”.
 - Text color changes on “Change Font Color”.
 - Background color changes on “Change Background”.

8. Verify the result:

- Ensure that each button performs the expected action.
- Make sure the changes are visible immediately on the TextView or layout.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.FirstApp">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Activity_main.xml

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="509dp"
        android:gravity="center"
        android:scaleX="1"
        android:scaleY="1"
        android:text="★ THALAPATHY ★"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.313" />

    <Button
        android:id="@+id/btnFontSize"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Change Font Size"
        tools:layout_editor_absoluteX="125dp"
        tools:layout_editor_absoluteY="272dp" />

    <Button
        android:id="@+id/btnFontColor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Change Font Color"
        tools:layout_editor_absoluteX="122dp"
        tools:layout_editor_absoluteY="341dp" />

    <Button
        android:id="@+id/btnBgColor"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Change Background"
        tools:layout_editor_absoluteX="117dp"
        tools:layout_editor_absoluteY="405dp" />

    </LinearLayout>
```

MainActivity.kt

```
package com.example.firstapp

import android.graphics.Color
import android.os.Bundle
import android.text.Layout
import android.widget.Button
import android.widget.LinearLayout
import android.widget.TextView
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        val textView = findViewById<TextView>(R.id.textView)
        val btnFontSize = findViewById<Button>(R.id.btnFontSize)
        val btnFontColor = findViewById<Button>(R.id.btnFontColor)
        val btnBgColor = findViewById<Button>(R.id.btnBgColor)
        val main = findViewById<LinearLayout>(R.id.main)

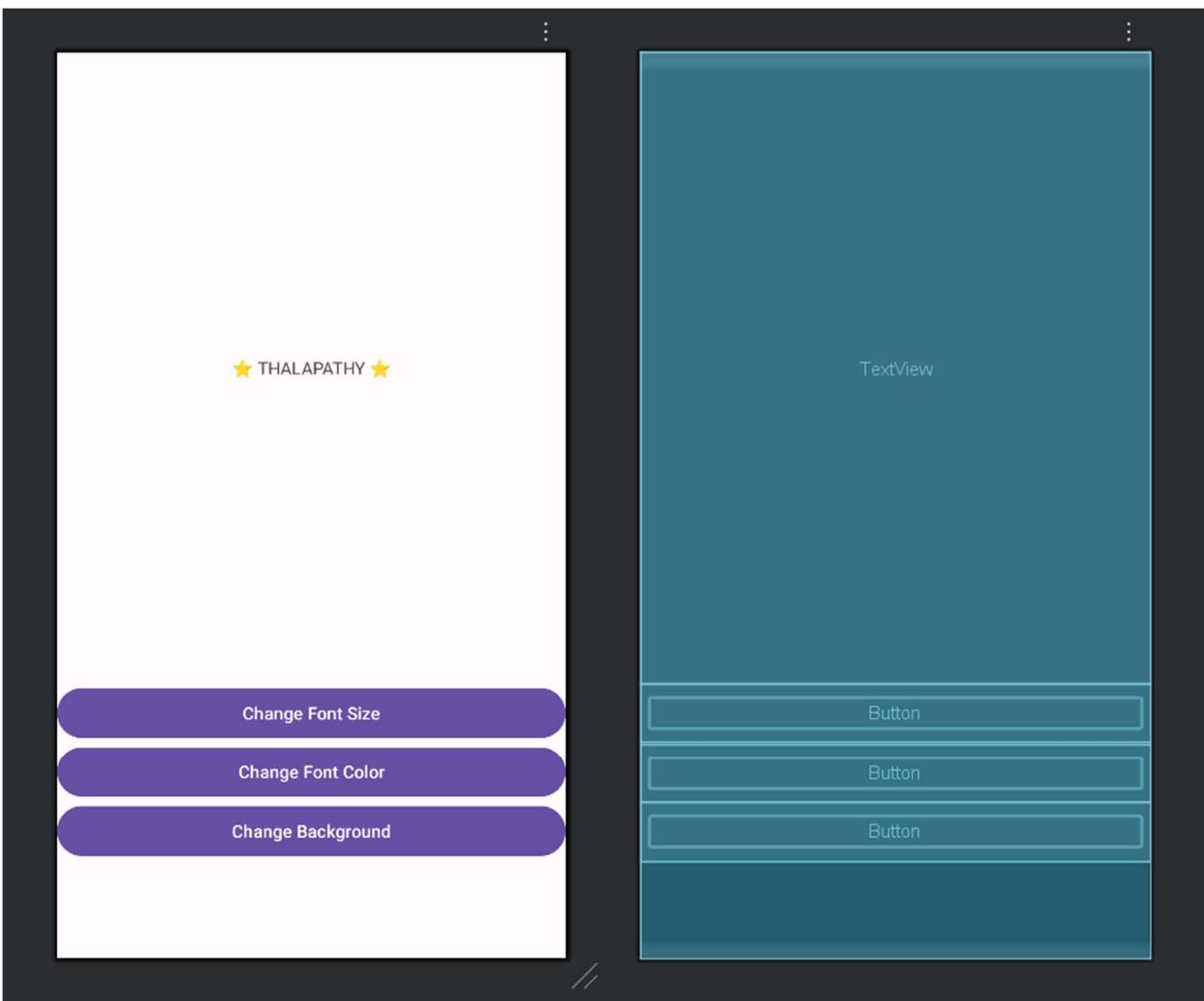
        // Change font size
        var fontSize: Float = 10f
        btnFontSize.setOnClickListener {
            textView.setTextSize(fontSize)
            fontSize = ((fontSize+10)%50)
        }

        // Change font color
        var fontColor: Int=5
        btnFontColor.setOnClickListener {
            when(fontColor%5){
                0 -> textView.setTextColor(Color.RED)
                1 -> textView.setTextColor(Color.YELLOW)
                2 -> textView.setTextColor(Color.BLUE)
                3 -> textView.setTextColor(Color.GREEN)
                4 -> textView.setTextColor(Color.MAGENTA)
            }
        }
    }
}
```

```
        }
        fontColor++
    }

    // Change background color
    var bgColor: Int=5
    btnBgColor.setOnClickListener {
        when(bgColor%5){
            1 -> main.setBackgroundColor(Color.RED)
            2 -> main.setBackgroundColor(Color.YELLOW)
            3 -> main.setBackgroundColor(Color.BLUE)
            4 -> main.setBackgroundColor(Color.GREEN)
            0 -> main.setBackgroundColor(Color.MAGENTA)
        }
        bgColor++
        /*if (isBgYellow) {
            textView.setBackgroundColor(Color.LTGRAY)
        } else {
            textView.setBackgroundColor(Color.YELLOW)
        }
        isBgYellow = !isBgYellow*/
    }
}
}
```

Output



Result:

The application successfully changes the text size, text color, and background color when the respective buttons are pressed. All changes are applied immediately on the screen.