***Создание контекстного меню***

**Цель работы**

Научиться создавать и использовать контекстное меню на отдельных элементах приложения

**Методические указания**

Контекстное меню вызывается в Андроид длительным нажатием на каком-либо экранном компоненте. Обычно оно используется в списках, когда на экран выводится список однородных объектов (например письма в почт.ящике) и, чтобы выполнить действие с одним из этих объектов, мы вызываем контекстное меню для него. Но т.к. списки мы еще не проходили, сделаем пример попроще и будем вызывать контекстное меню для TextView.

Откроем main.xml и нарисуем там два TextView:

<?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\_height="wrap\_content"

        android:textSize="26sp"

        android:layout\_width="wrap\_content"

        android:id="@+id/tvColor"

        android:layout\_marginBottom="50dp"

        android:layout\_marginTop="50dp"

        android:text="Text color">

    </TextView>

    <TextView

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:textSize="22sp"

        android:id="@+id/tvSize"

        android:text="Text size">

    </TextView>

</LinearLayout>

Для первого TextView мы сделаем контекстное меню, с помощью которого будем менять цвет текста. Для второго – будем менять размер текста.

Принцип создания контекстного меню похож на создание обычного меню. Но есть и отличия.

Метод создания onCreateContextMenu вызывается каждый раз перед показом меню. На вход ему передается:

- ContextMenu, в который мы будем добавлять пункты

- View - элемент экрана, для которого вызвано контекстное меню

- ContextMenu.ContextMenuInfo – содержит доп.информацию, когда контекстное меню вызвано для элемента списка. Пока мы это не используем, но, когда будем изучать списки, увидим, что штука полезная.

Метод обработки onContextItemSelected аналогичный методу onOptionsItemSelected для обычного меню. На вход передается MenuItem – пункт меню, который был нажат.

Также нам понадобится третий метод registerForContextMenu. На вход ему передается View и это означает, что для этой View необходимо создавать контекстное меню. Если не выполнить этот метод, контекстное меню для View создаваться не будет.

Давайте кодить, открываем MainActivity.java. Опишем и найдем TextView и укажем, что необходимо создавать для них контекстное меню.

TextView tvColor, tvSize;

  /\*\* Called when the activity is first created. \*/

  @Override

  public void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

      setContentView(R.layout.main);

      tvColor = (TextView) findViewById(R.id.tvColor);

      tvSize = (TextView) findViewById(R.id.tvSize);

      // для tvColor и tvSize необходимо создавать контекстное меню

      registerForContextMenu(tvColor);

      registerForContextMenu(tvSize);

  }

Теперь опишем создание контекстных меню. Используем константы для хранения ID пунктов меню.

final int MENU\_COLOR\_RED = 1;

final int MENU\_COLOR\_GREEN = 2;

final int MENU\_COLOR\_BLUE = 3;

final int MENU\_SIZE\_22 = 4;

final int MENU\_SIZE\_26 = 5;

final int MENU\_SIZE\_30 = 6;

И создаем

@Override

public void onCreateContextMenu(ContextMenu menu, View v,

    ContextMenuInfo menuInfo) {

  // TODO Auto-generated method stub

  switch (v.getId()) {

case R.id.tvColor:

  menu.add(0, MENU\_COLOR\_RED, 0, "Red");

  menu.add(0, MENU\_COLOR\_GREEN, 0, "Green");

  menu.add(0, MENU\_COLOR\_BLUE, 0, "Blue");

  break;

case R.id.tvSize:

  menu.add(0, MENU\_SIZE\_22, 0, "22");

  menu.add(0, MENU\_SIZE\_26, 0, "26");

  menu.add(0, MENU\_SIZE\_30, 0, "30");

  break;

}

}

Обратите внимание, что мы по ID определяем View, для которого вызвано контекстное меню и в зависимости от этого создаем определенное меню. Т.е. если контекстное меню вызвано для tvColor, то мы создаем меню с перечислением цветов, а если для tvSize – с размерами шрифта.

В качестве ID пунктов мы использовали константы. Группировку и сортировку не используем, поэтому используем нули в качестве соответствующих параметров.

Можно все сохранить и запустить. При долгом нажатии на TextView должны появляться контекстные меню.

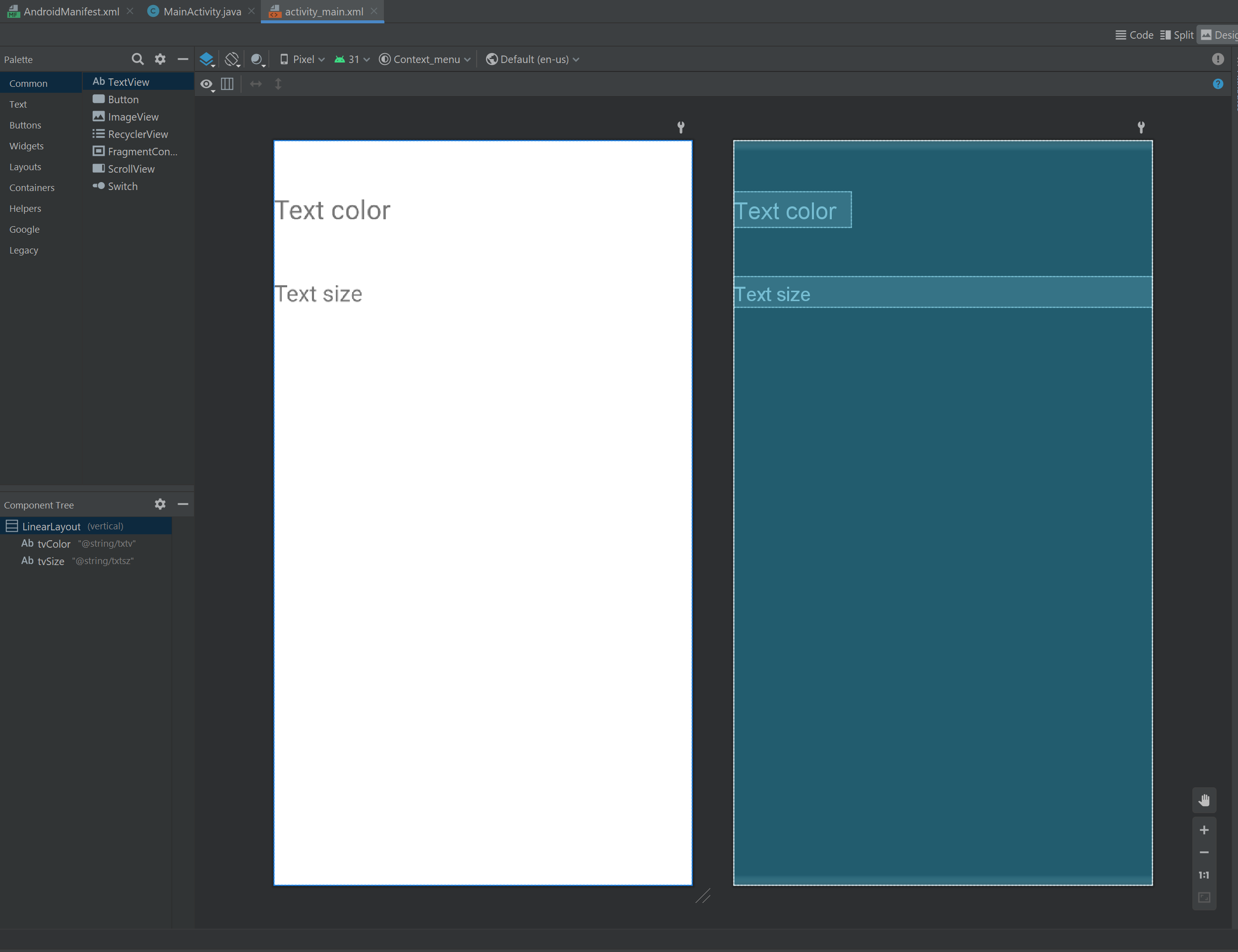
Листинг MainActivity.java

package com.example.context\_menu;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
import android.view.ContextMenu;  
import android.view.View;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 TextView tvColor, tvSize;  
  
 final int MENU\_COLOR\_RED = 1;  
 final int MENU\_COLOR\_GREEN = 2;  
 final int MENU\_COLOR\_BLUE = 3;  
  
 final int MENU\_SIZE\_22 = 4;  
 final int MENU\_SIZE\_26 = 5;  
 final int MENU\_SIZE\_30 = 6;  
  
  
 */\*\* Called when the activity is first created. \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 tvColor = findViewById(R.id.*tvColor*);  
 tvSize = findViewById(R.id.*tvSize*);  
  
 // для tvColor и tvSize необходимо создавать контекстное меню  
  
 registerForContextMenu(tvColor);  
 registerForContextMenu(tvSize);  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v,  
 ContextMenu.ContextMenuInfo menuInfo) {  
 // *TODO Auto-generated method stub* switch (v.getId()) {  
 case R.id.*tvColor*:  
 menu.add(0, MENU\_COLOR\_RED, 0, "Red");  
 menu.add(0, MENU\_COLOR\_GREEN, 0, "Green");  
 menu.add(0, MENU\_COLOR\_BLUE, 0, "Blue");  
 break;  
 case R.id.*tvSize*:  
 menu.add(0, MENU\_SIZE\_22, 0, "22");  
 menu.add(0, MENU\_SIZE\_26, 0, "26");  
 menu.add(0, MENU\_SIZE\_30, 0, "30");  
 break;  
 }  
 }  
}

Manifest

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.context\_menu">  
  
 <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.Context\_menu">  
 <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>

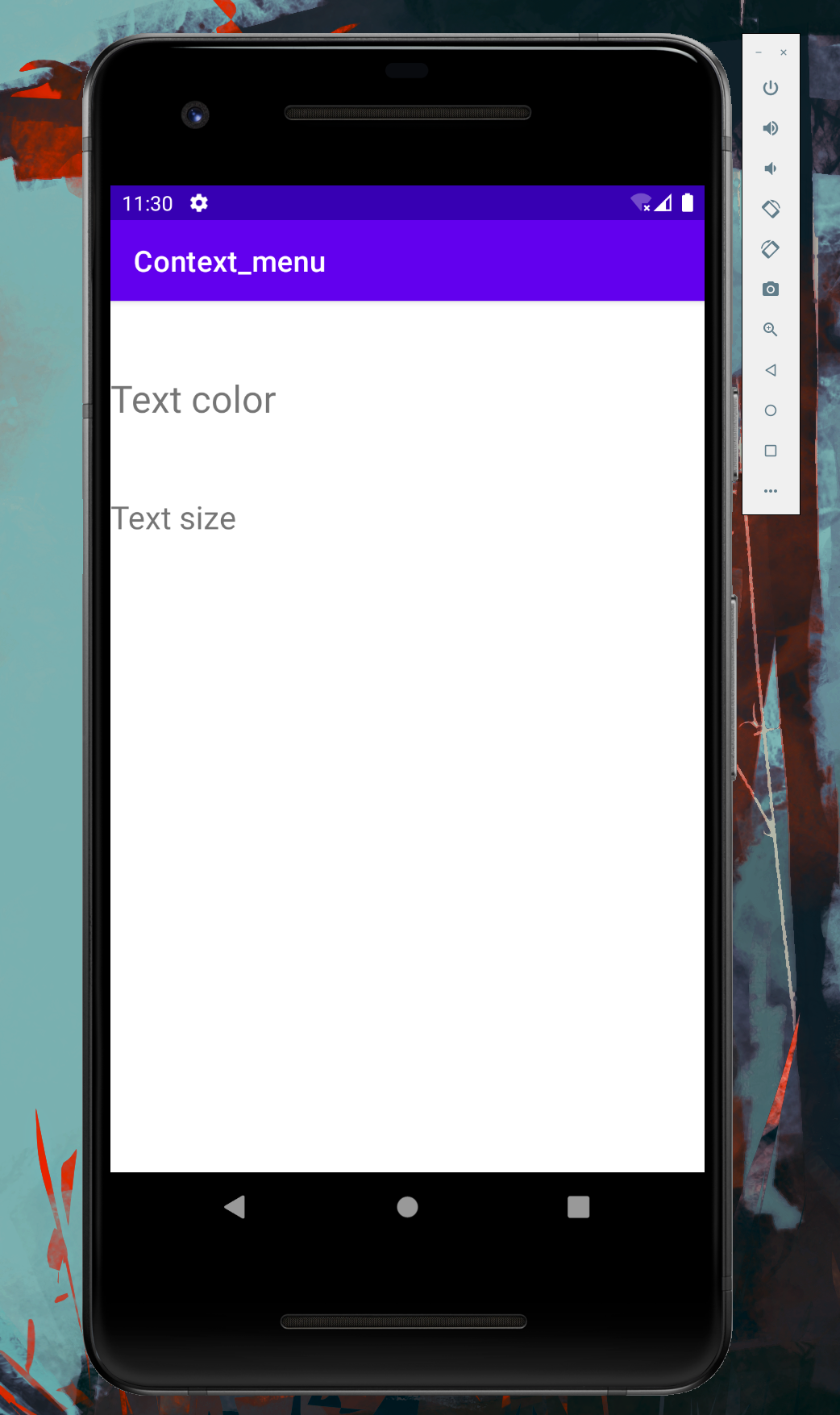
Как выглядит activity\_main.xml



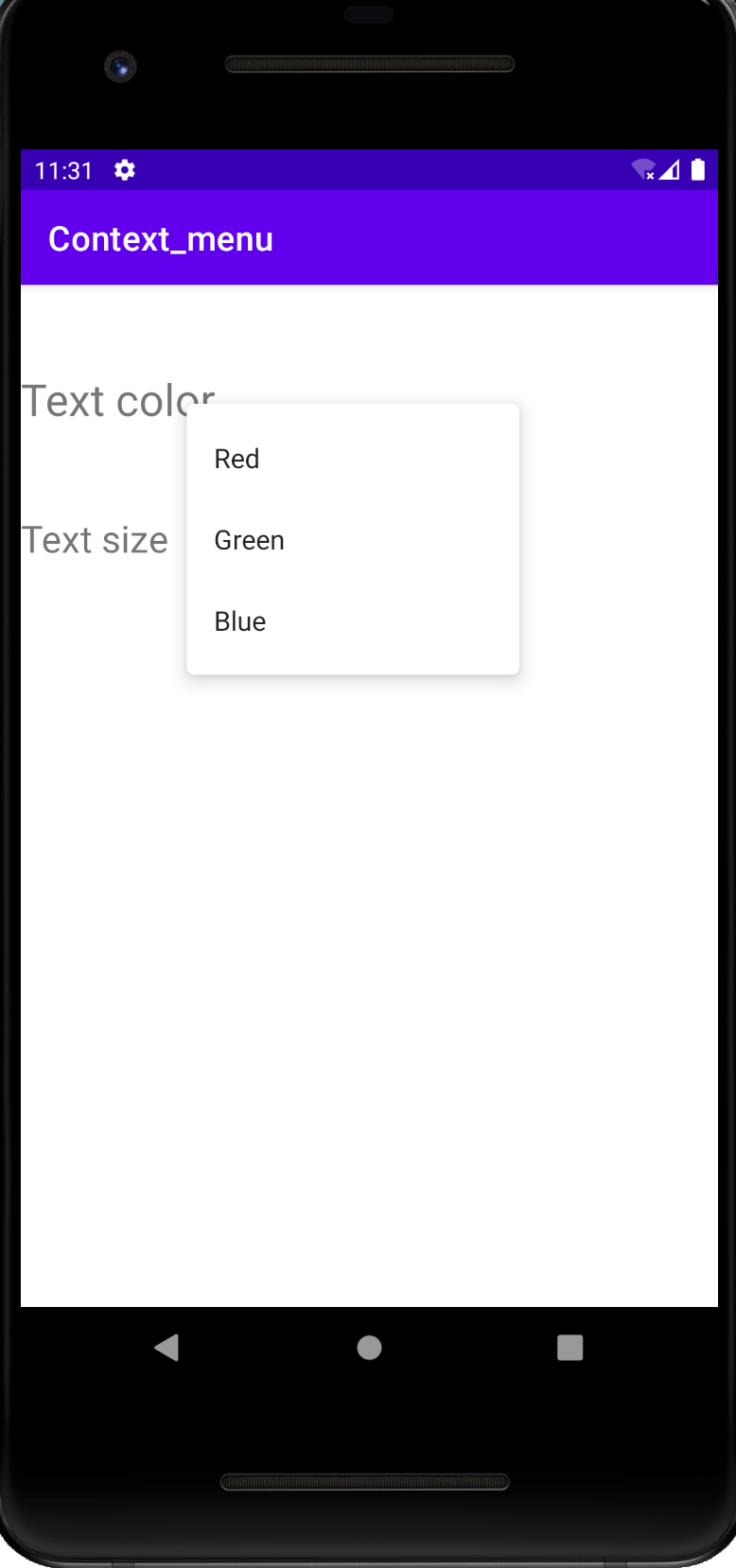
Код activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvColor"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginBottom="50dp"  
 android:text="@string/txtv"  
 android:textSize="26sp"></TextView>  
  
 <TextView  
 android:id="@+id/tvSize"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="22sp"  
 android:text="@string/txtsz">  
 </TextView>  
  
</LinearLayout>

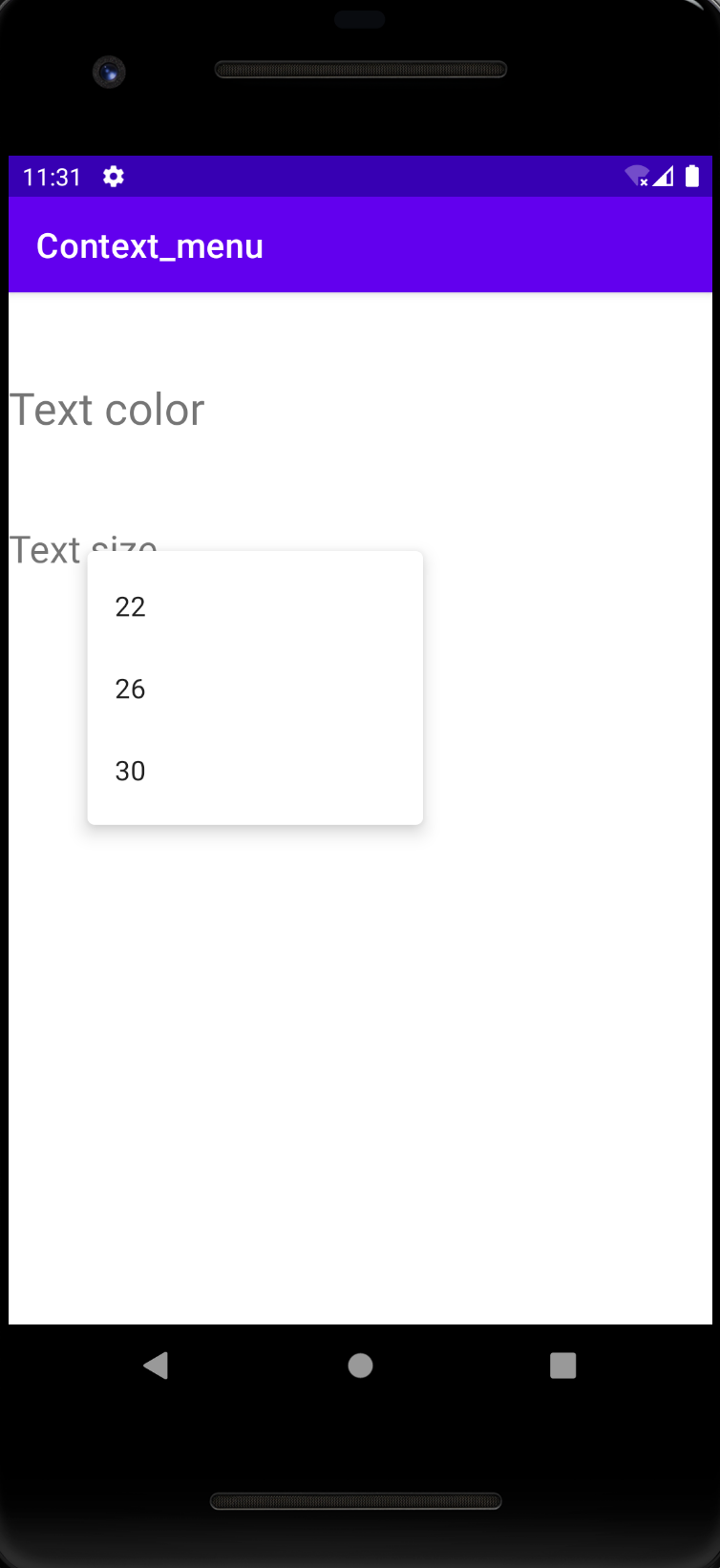
Работа программы:



Зажимаем на Textview – Text Color



Зажимаем на Textview – Text size



**Дополнительные задания**

1. Создайте приложение-текстовый редактор, состоящий из