МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Звіт

з лабораторної роботи № 2 з дисциплін

Виконав: Перевірив:

ст. гр. ПЗПІ-23-2 ст. викладач кафедри ПІ

Семьонов Олег Сокорчук Ігор Петрович

Харків 2024

ЗАВДАННЯ ДО РОБОТИ

Розробити редактор, у якому з допомогою трьох регуляторів SeekBar встановлюється колір панелі.

Регулятори визначають RGB компоненти кольору панелі. Зовнішній вигляд екрану редактора повинен залежати від орієнтації пристроя. У портретній орієнтації пристроя, регулятори повинні знаходитися знизу від панелі, займати 1/3 висоти і розташовуватися на всю ширину екрану. У ланшафтній – праворуч від панелі і займати 1/3 ширини. Решту екрану повинна займати кольорова панель.

ХІД РОБОТИ

Після створення нового проекту одразу створюємо ще одну папку в ресурсах під назвою layout-land і також створюємо туди ще одну розмітку activity\_main.xml для ландшафтної орієнтації екрану. Тепер можемо налаштовувати портретну розмітку. Я вибрав лінійну розмітку, бо для данної роботи її легко використовувати. За умовами наше View має займати 2/3 висоти, а регулятори 1/3. Для цього ще створимо всередині ще одну лінійну розмітку з властивістю layout\_weight та layout\_height = 0dp. Для View створимо такі ж самі атрибути для автоматичного визначення висоти. Ширина елементів дорівнєю контейнеру, тому використаємо значення match\_parent.

<View  
 android:id="@+id/view"  
 android:layout\_width="match\_parent"  
 android:layout\_weight="0.66"  
 android:layout\_height="0dp" />  
  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="0.33"  
 android:gravity="center"  
 android:orientation="vertical">

Після цього також створимо три SeekBar, які також розташуємо рівномірно по висоті, а також додамо властивість max для максимального значення нашого бару. Наприклад:

<SeekBar  
 android:id="@+id/seekR"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>

Тепер трішки переробимо ці ж самі елементи для ландшафтної розмітки. Фактично, треба поміняти місцями висоту та ширину у вищевизначених елементах.

<View  
 android:id="@+id/view"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.66" />  
<LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.33"  
 android:orientation="vertical">

Повний код обох розміток буде знаходитись у [ДОДАТОК А] та [ДОДАТОК В].

Після створення обох розміток перейдемо до логіки. У файлі MainActivity.java ініціалізуємо наші View і три SeekBar.

private View colorPanel;  
private SeekBar seekBarR, seekBarG, seekBarB;

А також зробимо метод getElements(), який в свою чергу буде викликатися у методі життєвого циклу onCreate(). Метод не є обов’язковим, адже він лише знаходить наші елементи за айді, що можно було реалізувати і без методу. Але для більшої компактності він є.

private void getElements(){  
 colorPanel = findViewById(R.id.view);  
 seekBarR = findViewById(R.id.seekR);  
 seekBarG = findViewById(R.id.seekG);  
 seekBarB = findViewById(R.id.seekB);  
}

Наступним методом є colorChanger(), який приймає кожний колір по одинці, потім передає до змінної finalColor у RGB вигляді і далі до нашої кольоровї панелі встановлюється вибраний колір.

private void colorChanger(){  
 int red = seekBarR.getProgress();  
 int green = seekBarG.getProgress();  
 int blue = seekBarB.getProgress();  
 int finalColor = Color.rgb(red,green,blue);  
 colorPanel.setBackgroundColor(finalColor);  
}

Останнім методом є setColor(). Метод setColor() використовується для налаштування слухача подій для кожного з трьох SeekBar у додатку. Метою цього методу є відстеження змін значень SeekBar, що контролюють інтенсивність червоного, зеленого та синього кольорів, і відповідно змінюють колір панелі.

private void setColor() {  
 SeekBar.OnSeekBarChangeListener seekBarChangeListener = new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 colorChanger();  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 };  
 seekBarR.setOnSeekBarChangeListener(seekBarChangeListener);  
 seekBarG.setOnSeekBarChangeListener(seekBarChangeListener);  
 seekBarB.setOnSeekBarChangeListener(seekBarChangeListener);  
}

Фактично всередені є методи,визначені інтерфейсом, адже треба використати три методи. Перший метод реагує на зміну нашого SeekBar, тому ми там і використаємо метод colorChanger(). Інші два методи у данному завданні не потрібні, але їх реалізація, хоч і пуста, вимагається цим самим інтерфейсом. Далі у цьому методі присвоємо до кожного бару свій слухач.

Код знаходиться у [ДОДАТОК С].

ВИСНОВКИ

Отже, під час виконання лабораторної роботи було закріплено вміння з зміни орієнтації, обробки подій та змін на екрані.

Відеозвіт. URL: <https://youtu.be/uiOKrHMAELA>

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SeekBar.OnSeekBarChangeListener <https://developer.android.com/reference/android/widget/SeekBar.OnSeekBarChangeListener>

2.SeekBar

<https://developer.android.com/reference/android/widget/SeekBar>

3. SeekBar (Слайдер)

<https://developer.alexanderklimov.ru/android/views/seekbar.php>

4.LinearLayout

<https://developer.android.com/reference/android/widget/LinearLayout>

ДОДАТОК А

<?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"  
 android:orientation="vertical"  
  
 tools:context=".MainActivity">  
  
 <View  
 android:id="@+id/view"  
 android:layout\_width="match\_parent"  
 android:layout\_weight="0.66"  
 android:layout\_height="0dp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="0.33"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <SeekBar  
 android:id="@+id/seekR"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>  
 <SeekBar  
 android:id="@+id/seekG"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>  
 <SeekBar  
 android:id="@+id/seekB"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>  
  
  
 </LinearLayout>  
  
</LinearLayout>

ДОДАТОК В

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/main"  
 android:orientation="horizontal">  
  
  
 <View  
 android:id="@+id/view"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.66" />  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.33"  
 android:orientation="vertical">  
  
 <SeekBar  
 android:id="@+id/seekR"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>  
 <SeekBar  
 android:id="@+id/seekG"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>  
 <SeekBar  
 android:id="@+id/seekB"  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:max="255"/>  
 </LinearLayout>  
  
</LinearLayout>

ДОДАТОК С

package com.nuresemonovoleh.android\_pzpi\_23\_2\_semonov\_oleh\_labtask2;  
  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.SeekBar;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
 private View colorPanel;  
 private SeekBar seekBarR, seekBarG, seekBarB;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
 getElements();  
 setColor();  
 }  
 private void getElements(){  
 colorPanel = findViewById(R.id.view);  
 seekBarR = findViewById(R.id.seekR);  
 seekBarG = findViewById(R.id.seekG);  
 seekBarB = findViewById(R.id.seekB);  
 }  
 private void colorChanger(){  
 int red = seekBarR.getProgress();  
 int green = seekBarG.getProgress();  
 int blue = seekBarB.getProgress();  
 int finalColor = Color.rgb(red,green,blue);  
 colorPanel.setBackgroundColor(finalColor);  
 }  
 private void setColor() {  
 SeekBar.OnSeekBarChangeListener seekBarChangeListener = new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 colorChanger();  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 };  
 seekBarR.setOnSeekBarChangeListener(seekBarChangeListener);  
 seekBarG.setOnSeekBarChangeListener(seekBarChangeListener);  
 seekBarB.setOnSeekBarChangeListener(seekBarChangeListener);  
 }  
  
}