

**Министерство науки и высшего образования Российской Федерации**  
**Федеральное государственное автономное образовательное**  
**учреждение высшего образования**  
**Национальный исследовательский**  
**Томский политехнический университет**

Инженерная школа информационных технологий и робототехники  
Отделение информационных технологий

Отчет по лабораторной работе №5 по дисциплине

**«Язык Kotlin и основы разработки»**

Жизненный цикл активности. Контейнер GridLayout

Выполнил:

Студент группы 1A22

  
\_\_\_\_\_

О.К. Кравцов

Проверил:

Ст. преп. ОИТ ИШИТР

\_\_\_\_\_

В.А. Дорофеев

Томск 2025

## Задание

Создайте приложение с цветными плитками.

Сетка, в которой будут находиться плитки, должна быть сделана с помощью контейнера `GridLayout`, и занимать всё свободное пространство в активности. Каждая цветная плитка представляет собой элемент `TextView`, растянутый на всю ширину и высоту ячейки. Цвет фона элемента задаётся с помощью метода `setBackgroundColor`.

Плитки должны менять свои цвета как при нажатии на любую плитку, так и при появлении приложения на экране (например, при запуске программы, или если переключиться на другое приложение и тут же вернуться обратно).

Придумайте свой уникальный алгоритм генерации цветов для перекрашивания элементов: цвета всех элементов должны быть каким-то образом связаны друг с другом. В примере у всех элементов одинаковый базовый цвет, но разный альфа-канал, от 25 с шагом 25:

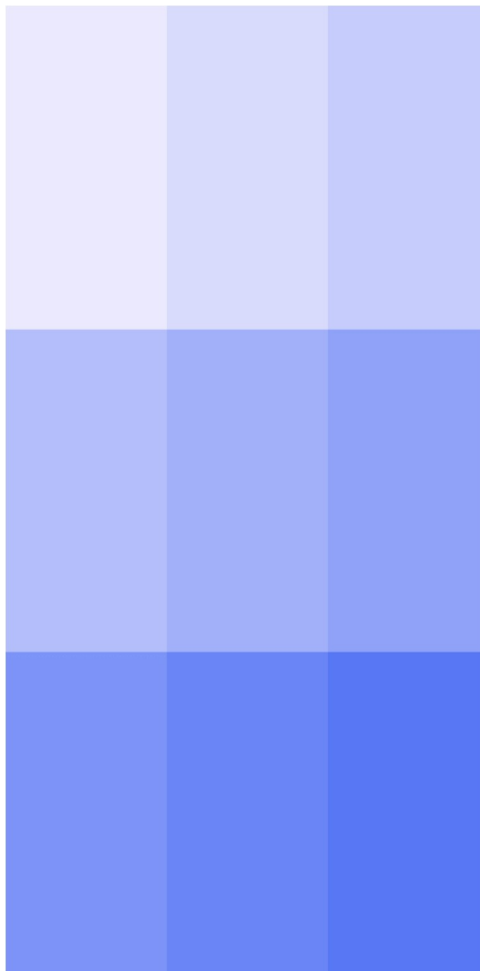


Рисунок 1 – Пример интерфейса

## Ход работы

1. Создан проект Lab5 на основе Empty Views Activity
2. Реализована разметка с контейнером GridLayout 4×3:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:rowCount="4"
    android:columnCount="3"
    android:background="@color/white">
</GridLayout>
```

3. Написан код для создания плиток и обработки событий:

```
package ru.olegkravtsov.lab5

import android.graphics.Color
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.GridLayout
import android.widget.TextView
import kotlin.random.Random

class MainActivity : AppCompatActivity() {

    private lateinit var gridLayout: GridLayout
    private var rows = 0
    private var cols = 0
    private var cellCount = 0
    private var currentColor = Color.RED

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

        gridLayout = findViewById(R.id.gridLayout)
        rows = gridLayout.rowCount
        cols = gridLayout.columnCount
        cellCount = rows * cols

        createTiles()
        updateAllTiles()
    }

    override fun onResume() {
        super.onResume()
    }
}
```

```

        generateNewColor()
        updateAllTiles()
    }

    private fun createTiles() {
        for (i in 0 until cellCount) {
            val textView = TextView(this).apply {
                text = ""
                textSize = 0f
                layoutParams = GridLayout.LayoutParams().apply {
                    width = 0
                    height = 0
                    rowSpec = GridLayout.spec(i / cols, 1f)
                    columnSpec = GridLayout.spec(i % cols, 1f)
                }
            }

            textView.setOnClickListener {
                generateNewColor()
                updateAllTiles()
            }

            gridLayout.addView(textView)
        }
    }

    private fun generateNewColor() {
        currentColor = Color.rgb(
            Random.nextInt(256),
            Random.nextInt(256),
            Random.nextInt(256)
        )
    }

    private fun updateAllTiles() {
        for (i in 0 until cellCount) {
            val textView = gridLayout.getChildAt(i) as TextView
            val transparency = 25 + (i * 15)
            val tileColor = Color.argb(
                transparency,
                Color.red(currentColor),
                Color.green(currentColor),
                Color.blue(currentColor)
            )
            textView.setBackgroundColor(tileColor)
        }
    }
}

```

4. Реализован алгоритм генерации цветов: все плитки имеют один основной цвет, но с разной прозрачностью (от 25 до 220 с шагом 15)

## Результат работы

Приложение представляет собой сетку 4×3 из цветных плиток. При нажатии на любую плитку или при возврате приложения на экран все плитки меняют цвет. Каждая плитка имеет разную прозрачность одного основного цвета.

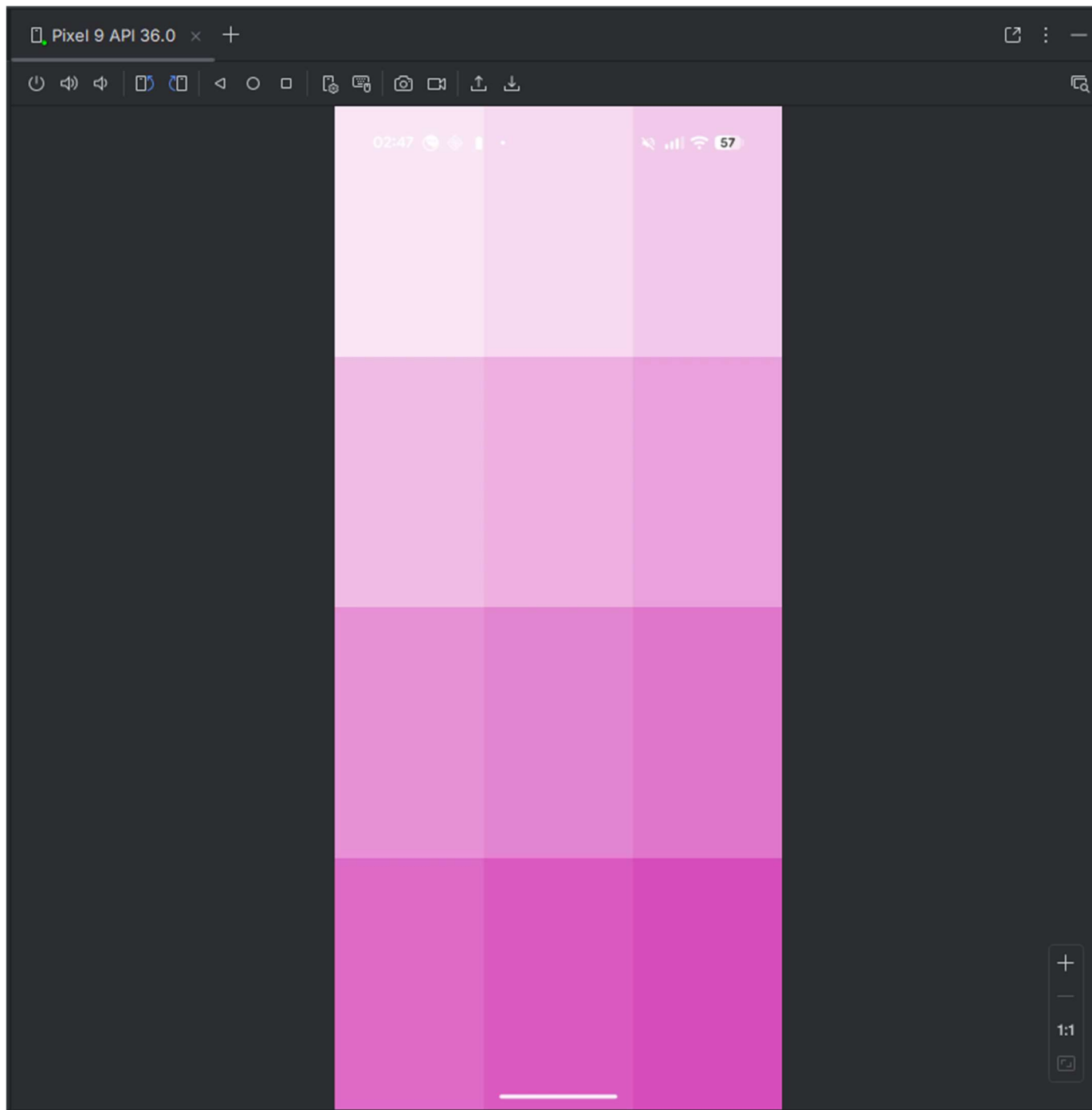


Рисунок 2 - Результат

## **Выводы**

Освоены GridLayout, жизненный цикл активности и обработка кликов.  
Реализована генерация цветов с разной прозрачностью.