**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине: **«Программирование мобильных информационных систем»**

на тему: **«Двухмерная и трехмерная графика. Мультимедиа-возможности в Android»**

Выполнил: студент группы ИТП-31

Пронуза М.Ю.

Проверил ст. преподаватель:

Стефановский И.Л.

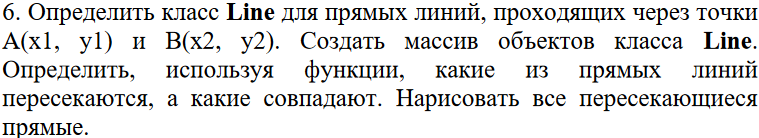
Гомель 2024

**Цель работы**: изучить программирование двухмерной и трехмерной графики в Android.

**Задание:**

Разработать приложение Android, выполняющее следующее задание. Выполнить запуск приложения на эмуляторе.

Вариант задания представлен на рисунке 1.

Рисунок 1 – вариант задания

**Ход выполнения и результаты работы:**

На рисунке 2 представлено получившееся изображение. Совпадающие линии выделены красным.

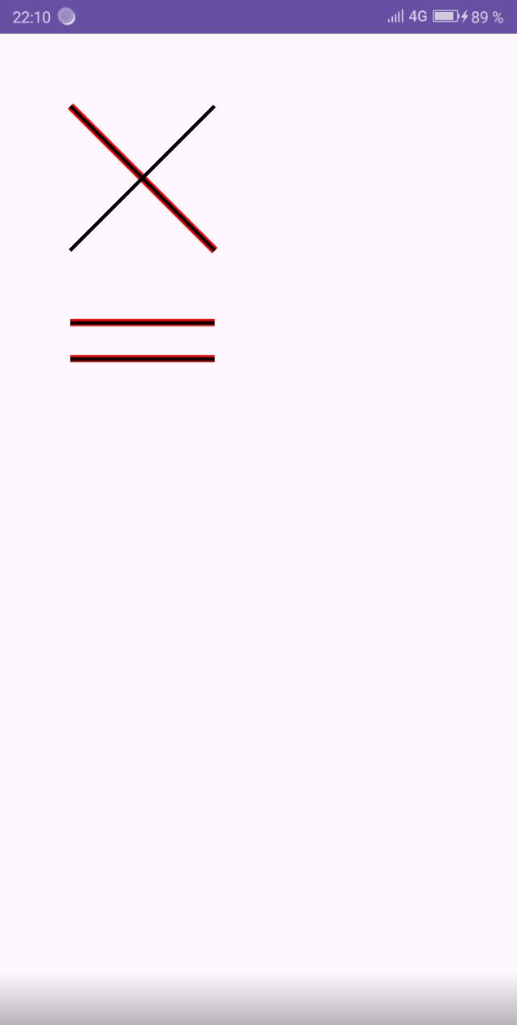


Рисунок 2 – получившееся изображение

На рисунке 3 представлен результат вычисления пересечений и совпадений.

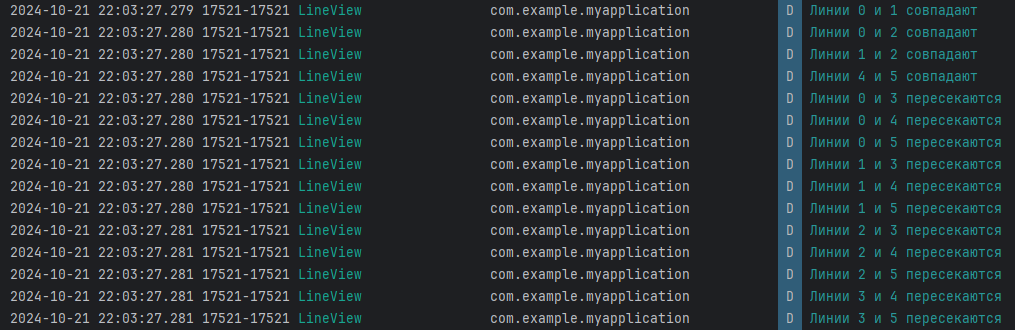


Рисунок 3 – результат вычисления пересечений и совпадений

Текст *MainActivity.java* представлен в приложении А.

Текст *Line.java* представлен в приложении Б.

Текст *LineView.java* представлен в приложении В.

**Вывод:** в ходе выполнения лабораторной работы изучено программирование двухмерной и трехмерной графики в Android.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст *MainActivity***

package com.example.myapplication;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.graphics.Insets;

import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;

import java.util.ArrayList;

import java.util.List;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Упрощенный пример массива линий для отладки

List<Line> lines = new ArrayList<>();

// Совпадающие линии

lines.add(new Line(100, 100, 200, 200)); // Прямая 1

lines.add(new Line(100, 100, 200, 200)); // Прямая 2 (точная копия Прямой 1)

// Пересекающиеся линии

lines.add(new Line(100, 100, 300, 300)); // Прямая 3

lines.add(new Line(100, 300, 300, 100)); // Прямая 4

// Параллельные линии

lines.add(new Line(100, 400, 300, 400)); // Прямая 5

lines.add(new Line(100, 450, 300, 450)); // Прямая 6

// Устанавливаем кастомное представление с линиями

setContentView(new LineView(this, lines));

}

}

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**Текст *Line***

package com.example.myapplication;

public class Line {

public float x1, y1, x2, y2;

public float A, B, C;

private static final float EPSILON = 0.0001f; // Погрешность для сравнения

public Line(float x1, float y1, float x2, float y2) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

// Вычисляем коэффициенты A, B, C уравнения прямой

A = y1 - y2;

B = x2 - x1;

C = x1 \* y2 - x2 \* y1;

}

// Метод для проверки пересечения линий

public boolean isIntersect(Line other) {

float det = this.A \* other.B - other.A \* this.B;

return Math.abs(det) > EPSILON; // Если определитель больше погрешности, линии пересекаются

}

// Метод для проверки совпадения линий с учетом погрешности

public boolean isCoincident(Line other) {

return Math.abs(this.A \* other.B - other.A \* this.B) < EPSILON &&

Math.abs(this.A \* other.C - other.A \* this.C) < EPSILON;

}

}

**ПРИЛОЖЕНИЕ В**

(обязательное)

**Текст *LineView***

package com.example.myapplication;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.util.Log;

import android.view.View;

import java.util.List;

public class LineView extends View {

public List<Line> lines;

public Paint linePaint;

public Paint coincidentLinePaint;

public LineView(Context context, List<Line> lines) {

super(context);

this.lines = lines;

linePaint = new Paint();

linePaint.setColor(Color.BLACK);

linePaint.setStrokeWidth(5);

coincidentLinePaint = new Paint();

coincidentLinePaint.setColor(Color.RED); // Совпадающие линии будем рисовать красным

coincidentLinePaint.setStrokeWidth(10);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

// Сначала рисуем совпадающие линии (красные)

for (int i = 0; i < lines.size(); i++) {

Line line1 = lines.get(i);

for (int j = i + 1; j < lines.size(); j++) {

Line line2 = lines.get(j);

if (line1.isCoincident(line2)) {

Log.d("LineView", "Линии " + i + " и " + j + " совпадают");

// Рисуем совпадающие линии красным цветом

canvas.drawLine(line1.x1, line1.y1, line1.x2, line1.y2, coincidentLinePaint);

canvas.drawLine(line2.x1, line2.y1, line2.x2, line2.y2, coincidentLinePaint);

}

}

}

// Затем рисуем пересекающиеся линии (чёрные)

for (int i = 0; i < lines.size(); i++) {

Line line1 = lines.get(i);

for (int j = i + 1; j < lines.size(); j++) {

Line line2 = lines.get(j);

if (line1.isIntersect(line2) && !line1.isCoincident(line2)) {

Log.d("LineView", "Линии " + i + " и " + j + " пересекаются");

// Рисуем только пересекающиеся линии

canvas.drawLine(line1.x1, line1.y1, line1.x2, line1.y2, linePaint);

canvas.drawLine(line2.x1, line2.y1, line2.x2, line2.y2, linePaint);

}

}

}

}

}