Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №6

по дисциплине "Технологии разработки мобильных приложений"

ТЕМА РАБОТЫ:

Работа с Canvas

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

преподаватель кафедры ИСПИ

Рощина А.И.

Владимир 2023 г.

Цель работы:

Разработать мобильное приложение, которое будет давать возможность

рисовать простейшие геометрические фигуры.

Выполнение тестовых примеров из методички

Тестовый пример 1.

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(new DrawView(this));  
 }  
  
 class DrawView extends SurfaceView implements SurfaceHolder.Callback {  
 private DrawThread drawThread;  
  
 public DrawView(Context context){  
 super(context);  
 getHolder().addCallback(this);  
 }  
  
 @Override  
 public void surfaceChanged(SurfaceHolder holder, int format, int width, int height){  
  
 }  
  
 @Override  
 public void surfaceCreated(SurfaceHolder holder){  
 drawThread = new DrawThread(getHolder());  
 drawThread.setRunning(true);  
 drawThread.start();  
 }  
  
 @Override  
 public void surfaceDestroyed(SurfaceHolder holder){  
 boolean retry = true;  
  
 drawThread.setRunning(false);  
 while (retry){  
 try {  
 drawThread.join();  
 retry = false;  
 } catch (InterruptedException e){  
  
 }  
 }  
 }  
 }  
  
 class DrawThread extends Thread {  
 private boolean running = false;  
 private SurfaceHolder surfaceHolder;  
  
 public DrawThread(SurfaceHolder surfaceHolder){  
 this.surfaceHolder = surfaceHolder;  
 }  
  
 public void setRunning(boolean running){  
 this.running = running;  
 }  
  
 @Override  
 public void run(){  
 Canvas canvas;  
 while (running){  
 canvas = null;  
 try{  
 canvas = surfaceHolder.lockCanvas(null);  
 if (canvas == null){  
 continue;  
 }  
 canvas.drawColor(Color.*GREEN*);  
 } finally {  
 if (canvas != null){  
 surfaceHolder.unlockCanvasAndPost(canvas);  
 }  
 }  
 }  
 }  
 }  
}

Тестирование примера 1:

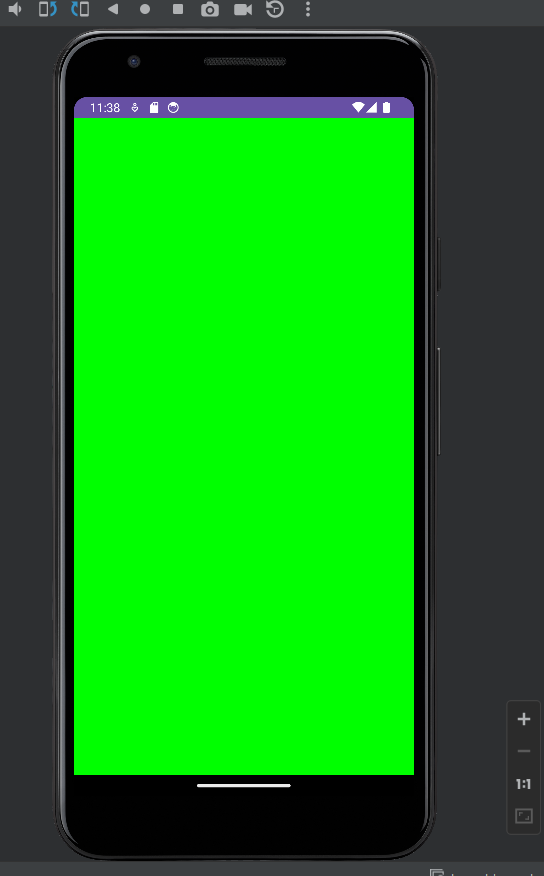


Рисунок 1. Тестирование примера 1

Тестовый пример 2.

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(new DrawView(this));  
 }  
  
 class DrawView extends View {  
 Paint p;  
 Rect rect;  
  
 public DrawView(Context context){  
 super(context);  
 p = new Paint();  
 rect = new Rect();  
 }  
  
 @Override  
 protected void onDraw(Canvas canvas){  
 canvas.drawARGB(80, 102, 204, 255);  
 p.setColor(Color.*RED*);  
 p.setStrokeWidth(10);  
 canvas.drawPoint(50, 50, p);  
 canvas.drawLine(100, 100, 500, 50, p);  
 canvas.drawCircle(100, 200, 50, p);  
 canvas.drawRect(200, 150, 400, 200, p);  
 rect.set(250, 300, 350, 500);  
 canvas.drawRect(rect, p);  
 }  
 }  
}

Тестирование примера 2:

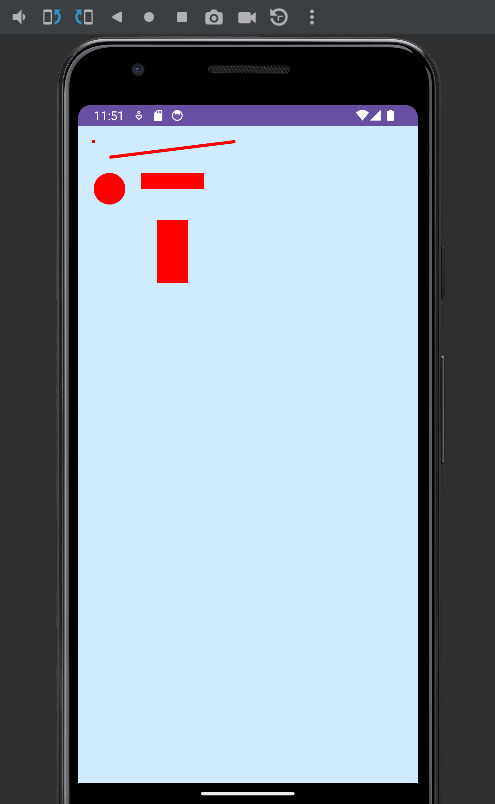


Рисунок 2. Тестирование примера 2

Пример 3.

class DrawView extends View {  
  
 Paint p;  
 RectF rectf;  
 float[] points;  
 float[] points1;  
  
 public DrawView(Context context) {  
 super(context);  
 p = new Paint();  
 rectf = new RectF(700,100,800,150);  
 points = new float[]{100,50,150,100,150,200,50,200,50,100};  
 points1 = new float[]{300,200,600,200,300,300,600,300,400,100,400,400,500,100,500,400};  
 }  
  
 @Override  
 protected void onDraw(Canvas canvas) {  
 canvas.drawARGB(80, 102, 204, 255);  
  
 p.setColor(Color.*RED*);  
 p.setStrokeWidth(10);  
  
 // рисуем точки их массива points  
 canvas.drawPoints(points,p);  
  
 // рисуем линии по точкам из массива points1  
 canvas.drawLines(points1,p);  
  
 // перенастраиваем кисть на зеленый цвет  
 p.setColor(Color.*GREEN*);  
  
 // рисуем закругленный прямоугольник по координатам из rectf  
 // радиусы закругления = 20  
 canvas.drawRoundRect(rectf, 20, 20, p);  
  
 // смещаем коорднаты rectf на 150 вниз  
 rectf.offset(0, 150);  
 // рисуем овал внутри прямоугольника rectf  
 canvas.drawOval(rectf, p);  
  
 // смещаем rectf в (900,100) (левая верхняя точка)  
 rectf.offsetTo(900, 100);  
 // увеличиваем rectf по вертикали на 25 вниз и вверх  
 rectf.inset(0, -25);  
 // рисуем дугу внутри прямоугольника rectf  
 // с началом в 90, и длиной 270  
 // соединение крайних точек через центр  
 canvas.drawArc(rectf, 90, 270, true, p);  
  
 // смещаем коорднаты rectf на 150 вниз  
 rectf.offset(0, 150);  
 // рисуем дугу внутри прямоугольника rectf  
 // с началом в 90, и длиной 270  
 // соединение крайних точек напрямую  
 canvas.drawArc(rectf, 90, 270, false, p);  
  
 // перенастраиваем кисть на толщину линии = 3  
 p.setStrokeWidth(3);  
 // рисуем линию (150,450) - (150,600)  
 canvas.drawLine(150, 450, 150, 600, p);  
  
 // перенастраиваем кисть на синий цвет  
 p.setColor(Color.*BLUE*);  
  
 // настраиваем размер текста = 30  
 p.setTextSize(30);  
 // рисуем текст в точке (150,500)  
 canvas.drawText("text left", 150, 500, p);  
  
 // настраиваем выравнивание текста на центр  
 p.setTextAlign(Paint.Align.*CENTER*);  
 // рисуем текст в точке (150,525)  
 canvas.drawText("text center", 150, 525, p);  
  
 // настраиваем выравнивание текста на левое  
 p.setTextAlign(Paint.Align.*RIGHT*);  
 // рисуем текст в точке (150,550)  
 canvas.drawText("text right", 150, 550, p);  
 }  
}

Тестирование примера 3:

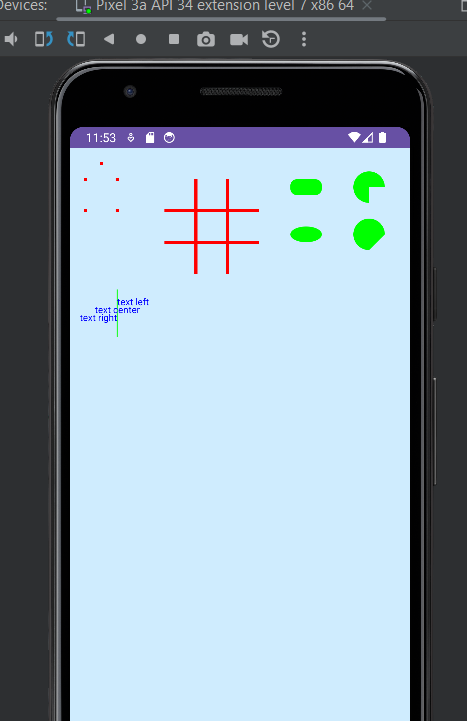


Рисунок 3. Пример 3

Пример 4.

@Override  
 protected void onDraw(Canvas canvas) {  
 canvas.drawARGB(80, 102, 204, 255);  
  
 p.setColor(Color.*BLUE*);  
 p.setStrokeWidth(10);  
  
 p.setTextSize(30);  
  
 // создаем строку с значениями ширины и высоты канвы  
 sb.setLength(0);  
 sb.append("width = ").append(canvas.getWidth())  
 .append(", height = ").append(canvas.getHeight());  
 canvas.drawText(sb.toString(), 100, 100, p);  
  
 // перенастраивам кисть на заливку  
 p.setStyle(Paint.Style.*FILL*);  
 canvas.drawRect(rect, p);  
  
 // перенастраивам кисть на контуры  
 p.setStyle(Paint.Style.*STROKE*);  
 rect.offset(150, 0);  
 canvas.drawRect(rect, p);  
  
 // перенастраивам кисть на заливку + контуры  
 p.setStyle(Paint.Style.*FILL\_AND\_STROKE*);  
 rect.offset(150, 0);  
 canvas.drawRect(rect, p);  
 }  
}

Тестирование примера 4:

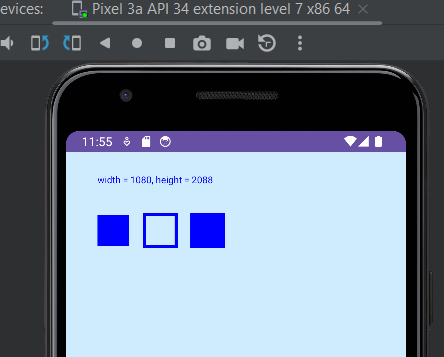


Рисунок 4. Тестирование примера 4

Выполнение задания (Вариант 1)

1) Нарисовать пятиконечную звезду

2) Нарисовать пятиугольник

3) Нарисовать шестиугольник

Был добавлен метод drawPolygon для создания различных полигонов. Взяв за основу код из примера, был создан проект и нарисованы с помощью объекта Canvas две фигуры: пятиугольник и шестиугольник (так как это объективно проще рисования пятиконечной звезды).

Код класса MainActivity:

package com.example.lab6\_parakhin;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(new DrawView(this));  
 }  
}

Код класса DrawView (который отвечает за рисование):

package com.example.lab6\_parakhin;  
  
import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Rect;  
import android.view.View;  
  
class DrawView extends View {  
 Paint p;

public DrawView(Context context) {  
 super(context);  
 p = new Paint();  
 }  
  
 @Override  
 public void onDraw(Canvas canvas){  
 canvas.drawARGB(80, 100, 200, 255);  
 p.setColor(Color.*BLUE*);  
 p.setStrokeWidth(15);  
 drawPolygon(canvas, 6, 780, 1200);  
 drawPolygon(canvas, 5 , 720, 900);  
 }  
 public void drawPolygon(Canvas canvas, int countOfSides, float Xstart, float Ystart){  
 int size = 500;  
 for (int i = 1; i <= countOfSides; i++){  
 canvas.drawLine(  
 Xstart, Ystart,  
 (float) (Xstart + size \* Math.*cos*(i \* 2 \* Math.*PI* /countOfSides)),  
 (float) (Ystart + size \* Math.*sin*(i \* 2 \* Math.*PI* / countOfSides)), p);  
 Xstart = (float) (Xstart + size \* Math.*cos*(i \* 2 \* Math.*PI* / countOfSides));  
 Ystart = (float) (Ystart + size \* Math.*sin*(i \* 2 \* Math.*PI* / countOfSides));  
 }  
 }  
}

Тестирование графического приложения:

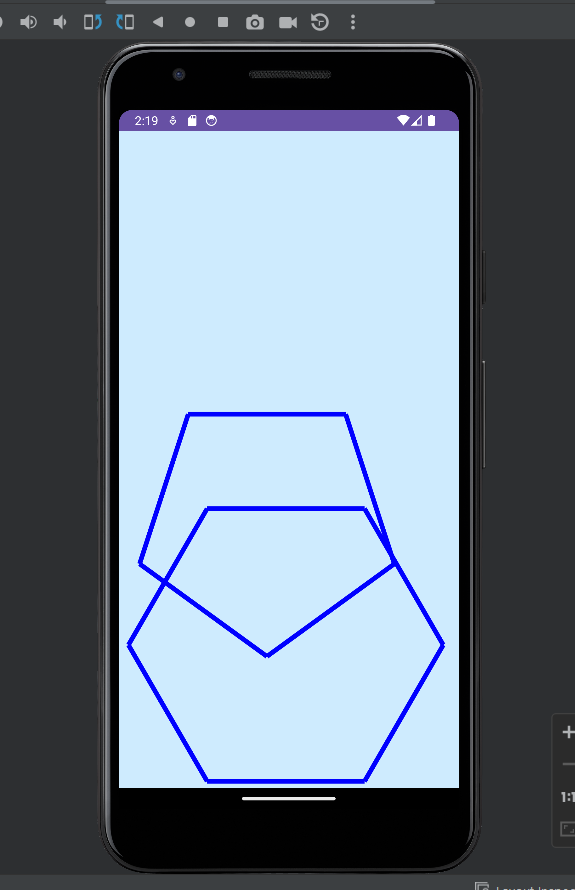


Рисунок 5. Рисование пятиугольника и шестиугольника на Canvas

Вывод

В результате выполнения работы, я разработал мобильное приложение, которое будет давать возможность рисовать простейшие геометрические фигуры.