Kpi-best

Міністерство освіти та науки України

Національний технічний університет України “Київський політехнічний інститут”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота № 2

з курсу "Графічне та географічне моделювання"

Виконав: студент 4 курсу

групи ІП-32

Ковтун А.В.

Київ – 2016

**Завдання:**

Геометрические и графические модели

1. Построить геометрическую модель M1 согласно варианту заданий. Для этого определить конкретные значения параметров R, , N, A, B, D, m, K, H и построить соответствующую ей графическую модель.

2. Разработать модель орнамента M2(М1,P2, … , PN, где М1- геометрическая модель по варианту, а P2, … , PN – параметры модели, определяющие конфигурацию, количество, размер, шаг тиражирования графической модели и др.

3. На основе разработанной модели М2, изменяя ее параметры добиться визуального спецэффекта (муар, пульсирование и т.п.).

**Код програми**

**package** com.example.andrew.lab2\_model;  
  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.graphics.Point;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.Display;  
**import** android.view.View;  
**import** android.widget.RelativeLayout;  
  
  
**import** java.util.LinkedList;  
**import** java.util.List;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 */\*  
 \* X=(A/B)\*R\*cos(t/B)+(1/B)\*R\*cos(A\*t/B)  
 Y=(A/B)\*R\*sin(t/B)-(1/B)\*R\*sin(A\*t/B)  
  
 \*/* **private final float a** = 40;  
 **private final float b** = 1;  
 **private final float r** = 2;  
 **private float n** = 7;  
 **private final float pi** = (**float**) 3.1415926535897932384626433832795;  
 **int height**;  
 **int width**;  
 Paint **paint**;  
 Paint **linePaint**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 Display display = getWindowManager().getDefaultDisplay();  
 Point size = **new** Point();  
 display.getSize(size);  
 **width** = size.**x**;  
 **height** = size.**y**;  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 DataPoint[] data = getData();  
 **paint** = **new** Paint();  
 **paint**.setTextSize(48f);  
 **paint**.setStrokeWidth(5);  
 **linePaint** = **new** Paint();  
 **linePaint**.setStrokeWidth(3);  
 **linePaint**.setColor(Color.***RED***);  
 **for** (DataPoint dataPoint : data) {  
 Log.*d*(**"dataString"**, dataPoint.toString());  
 }  
 RelativeLayout relativeLayout = (RelativeLayout) findViewById(R.id.***relativeLayout***);  
 relativeLayout.addView(**new** GraphView(getBaseContext()));  
  
 }  
  
 **private class** GraphView **extends** View {  
 **public** GraphView(Context context) {  
 **super**(context);  
 }  
  
 @Override  
 **protected void** onDraw(Canvas canvas) {  
 drawCoordinates(canvas);  
 drawGraphic(canvas);  
 }  
  
 **private void** drawGraphic(Canvas canvas) {  
 DataPoint[] dataPoints = getData();  
 DataPoint firstDataPoint = convertDataToPX(dataPoints[0]);  
 **float** lastX = firstDataPoint.**x**;  
 **float** lastY = firstDataPoint.**y**;  
 **for** (DataPoint dataPoint : dataPoints) {  
 DataPoint convertDataPoint = convertDataToPX(dataPoint);  
 Log.*d*(**"DataPoint"**, convertDataPoint.**x** + **" | "** + convertDataPoint.**y**);  
 canvas.drawLine(lastX, lastY, convertDataPoint.**x**, convertDataPoint.**y**, **linePaint**);  
 lastX = convertDataPoint.**x**;  
 lastY = convertDataPoint.**y**;  
 }  
 }  
  
 **private** DataPoint convertDataToPX(DataPoint functionDataPoint) {  
 **float** y = ((functionDataPoint.**y** / 5)) \* 50;  
 **float** x = ((**float**) (functionDataPoint.**x** / 5)) \* 50;  
 **return new** DataPoint(x, y);  
 }  
  
 **private void** drawCoordinates(Canvas canvas) {  
 canvas.drawLine(0, **height** / 2, **width**, **height** / 2, **paint**);  
 canvas.drawLine(50, **height** / 2 + 50, 50, 50, **paint**);  
 **for** (**int** i = 50; i < **width**; i += 100) {  
 canvas.drawLine(i, **height** / 2 + 5, i, **height** / 2 - 5, **paint**);  
 canvas.drawText(String.*valueOf*((i / 100) \* 5), i - 15, **height** / 2 + 45, **paint**);  
 }  
 **for** (**int** i = **height** / 2; i >= 0; i -= 100) {  
 canvas.drawLine(45, i, 55, i, **paint**);  
 canvas.drawText(String.*valueOf*(((**height** / 2 - i) / 100) \* 5), 0, i, **paint**);  
 }  
 }  
 }  
  
 **private class** DataPoint {  
 **public float x**;  
 **public float y**;  
  
 **public** DataPoint() {  
 }  
  
 **public** DataPoint(**float** x, **float** y) {  
 **this**.**x** = x;  
 **this**.**y** = y;  
 }  
 }  
 */\*  
 X=(A/B)\*R\*cos(t/B)+(1/B)\*R\*cos(A\*t/B)  
 Y=(A/B)\*R\*sin(t/B)-(1/B)\*R\*sin(A\*t/B)  
 \* \*/* **private** DataPoint[] getData() {  
 List<DataPoint> dataPoints = **new** LinkedList<>();  
 **for** (**float** t = 0; t < **n** \* **pi**; t += 0.1) {  
 **float** x = (**float**) ((**a** / **b**) \* **r** \* (**float**) Math.*cos*(t / **b**) + (1 / **b**) \* **r** \* (**float**) Math.*cos*(**a** \* (t / **b**)));  
 **float** y = (**float**) ((**a** / **b**) \* **r** \* (**float**) Math.*sin*(t / **b**) + (1 / **b**) \* **r** \* (**float**) Math.*sin*(**a** \* (t / **b**)));  
 DataPoint dataPoint = **new** DataPoint(x, y);  
 dataPoints.add(dataPoint);  
 *// Log.d("Point", dataPoint.x + " | " + dataPoint.y);* }  
 **return** dataPoints.toArray(**new** DataPoint[dataPoints.size()]);  
 }  
}

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:id="@+id/relativeLayout"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.andrew.lab2\_model.MainActivity"**>  
  
</**RelativeLayout**>

apply **plugin**: **'com.android.application'**android {  
 compileSdkVersion 24  
 buildToolsVersion **"24.0.2"** defaultConfig {  
 applicationId **"com.example.andrew.lab2\_model"** minSdkVersion 15  
 targetSdkVersion 24  
 versionCode 1  
 versionName **"1.0"** }  
 buildTypes {  
 release {  
 minifyEnabled **false** proguardFiles getDefaultProguardFile(**'proguard-android.txt'**), **'proguard-rules.pro'** }  
 }  
}  
  
dependencies {  
 compile fileTree(**dir**: **'libs'**, **include**: [**'\*.jar'**])  
 testCompile **'junit:junit:4.12'** compile **'com.android.support:appcompat-v7:24.2.1'** compile **'com.jjoe64:graphview:4.2.0'**}

**Результат виконання:**

