Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Комп’ютерна графіка

Лабораторна робота № 2

Виконав:
студент групи КП-51
Волощенко Олександр

Київ 2017

Тема: Побудова та анімація зображень за допомогою Java2D

Мета: Ознайомитися з можливостями побудови зображень та їх анімації у Java2D

Теоретичні відомості:

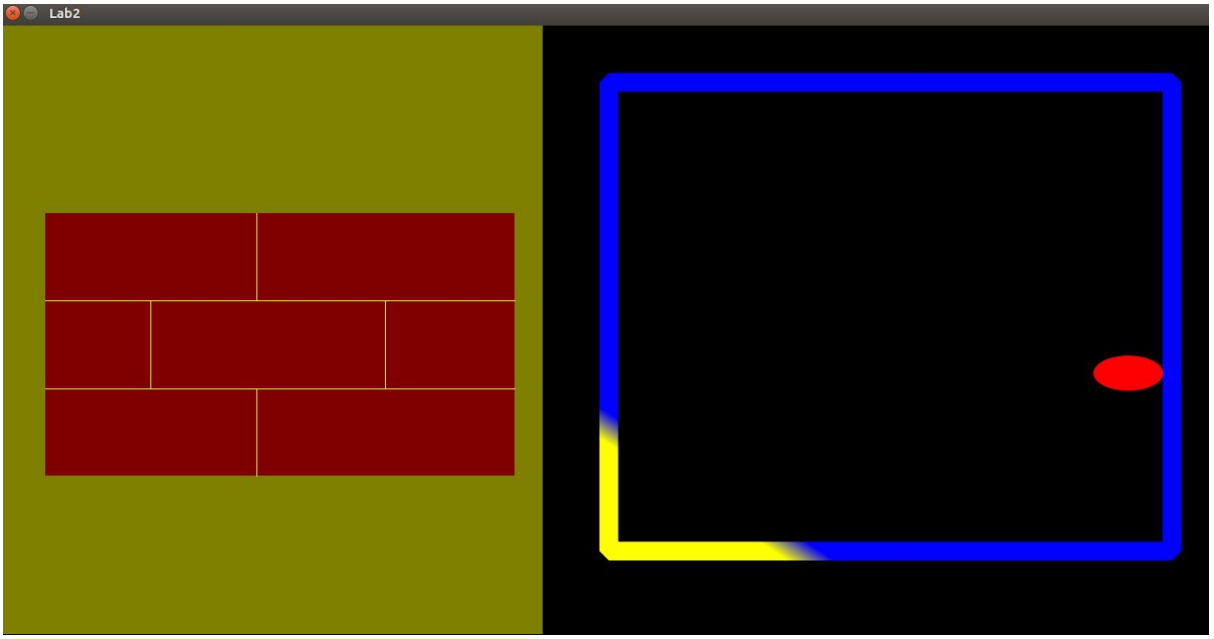
Є два види комп'ютерної графіки – растрова та векторна. Растрова графіка представляє зображення як набір пікселів. Векторна графіка використовує геометричні примітиви – точки, лінії, полігони, дуги. Java 2D надає можливість працювати як з растровою, так і з векторною графікою.

Java 2D це API для створення двовимірних зображень за допомогою мови програмування Java. Java 2D API має наступні властивості:

- широкий вибір геометричних примітивів
- визначення перетину фігур, тексту, зображень
- контроль якості рендерингу
- друк документів
- уніфікована модель рендерингу для дисплеїв та принтерів

Java 2D досить потужна технологія, за допомогою якої можна створювати потужні та красиві інтерфейси користувача, ігри, анімації, мультимедійні програми.

Виконання:



Код

```
package lab2;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

public class Window extends JPanel implements ActionListener {


    private final double points[][] = {
            { 275, 200 }, { 275, 293 }, { 550, 293 },
            { 412, 293 }, { 412, 387 }, { 550, 387 },
            { 275, 387 }, { 275, 480 }, { 275, 387 },
            { 50, 387  }, { 162, 387 }, { 162, 293 },
            { 50, 293  }, { 275, 293 }
    };

    private double scale = 1;
    private double delta = 0.1;

    private final double speed = 7;
```

```
    private double dx = speed, tx, dy, ty;

    private static int maxWidth;
    private static int maxHeight;

    private Window() {
        Timer timer = new Timer(100, this);
        timer.start();
    }

    public void paint(Graphics g) {
        super.paint(g);
        Graphics2D g2d = (Graphics2D)g;
        g2d.setBackground(Color.black);
        g2d.clearRect(0, 0, maxWidth+1, maxHeight+1);
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
                RenderingHints.VALUE_RENDER_QUALITY);

        //region From Lab1
        g2d.setColor(new Color(128,128,0));
        g2d.fillRect(0, 0, (int) (0.45*maxWidth), maxHeight);

        g2d.setColor(new Color(128,0,0));
        g2d.fillRect(50, 200, 500,280);


        g2d.setColor(Color.YELLOW);
        GeneralPath line = new GeneralPath();

        line.moveTo(points[0][0], points[0][1]);

        for (double[] point : points) {
            line.lineTo(point[0], point[1]);
        }

        line.closePath();
        g2d.draw(line);
        //endregion

        GradientPaint gp = new GradientPaint(5,25,new
Color(255,255,0),20,2,new Color(0,0,255), false);
        g2d.setPaint(gp);
        BasicStroke stroke = new BasicStroke(20, BasicStroke.CAP_BUTT,
                BasicStroke.JOIN_BEVEL);
        g2d.setStroke(stroke);

        g2d.drawRect(650,60,600,500);
```

```
        g2d.translate(tx, ty);
        g2d.scale(0.99,scale);

        g2d.setColor(Color.RED);

        g2d.fillOval(620,60,75, 75);

    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Lab2");
        frame.add(new Window());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1300, 680);
        frame.setResizable(false);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
        Dimension size = frame.getSize();
        Insets insets = frame.getInsets();
        maxWidth = size.width - insets.left - insets.right - 1;
        maxHeight = size.height - insets.top - insets.bottom - 1;
    }

    public void actionPerformed(ActionEvent e) {
        if ( scale < 0.5 || scale > 0.9) {
            delta = -delta;
        }

        if (ty < 0) {
            dy = tx = ty = 0;
            dx = speed;
        }

        if (tx > 550) {
            dx = 0;
            dy = speed;
        }

        if (ty > 480) {
            dy = 0;
            dx = -speed;
        }

        if (tx < 0) {
            dy = -speed;
            dx = 0;
        }

        scale += delta;
        tx += dx;
```

```
        ty += dy;
        repaint();
    }
}
```