Nama : Nur Cahyani Astika Fauzi

Nim : 1201220043

Class : SE-05-01

Mata Kuliah : Grafika Computer

3.2 Exercise 3.2 Apply the structural algorithm in section 3.3 to draw the line in figure 3.6.

Jawab :

a.) **Membangun Pola Awal**

- Diberikan dua titik ujung (x0, y0) dan (x1, y1) dari garis dengan kemiringan antara nol dan satu, nilai dx = x1 - x0 dan dy = y1 - y0 dihitung.
- Piksel awal dx, lebih banyak piksel yang harus digambar. Untuk piksel dx ini, diperlukan langkah diagonal dy. Sisanya (dx - dy) harus berupa langkah horizontal.
- Masalah yang harus dipecahkan terdiri dari menemukan urutan langkah diagonal dan horizontal yang benar.
- Barisan 3 Hdx-dyDdy, yang berisi jumlah langkah horizontal dan diagonal yang benar tetapi mungkin dalam urutan yang salah, digunakan sebagai perkiraan pertama untuk pola gambar garis.
- Permutasi yang sesuai dari urutan awal ini akan menghasilkan urutan yang benar untuk menggambar garis.

b.) **Menggambar Garis dan Kurva**

Pertimbangan untuk algoritma struktural juga akan dibatasi pada garis dengan kemiringan antara nol dan satu. Algoritma struktural membangun pola berulang untuk menggambar piksel sebagai urutan langkah horizontal (H) dan diagonal (D), berdasarkan prinsip-prinsip berikut.

c.) **Algoritma Brons**

Algoritma Brons membangun permutasi yang benar dari urutan awal Hdx-dyDdy dengan cara berikut:

- Jika dx dan dy (dan karenanya juga (dx - dy)) memiliki pembagi persekutuan terbesar lebih dari satu, yaitu g = gcd(dx, dy) > 1, maka garis piksel dapat digambar dengan pengulangan g dari urutan panjang dx/g.
- Oleh karena itu, dapat diasumsikan tanpa kehilangan keumuman bahwa dx dan dy tidak memiliki pembagi persekutuan.
- Biarkan P dan Q menjadi dua kata (urutan) di atas alfabet {D, H}.
- Dari urutan awal PpQq dengan frekuensi p dan q yang tidak memiliki pembagi persekutuan dan dengan asumsi tanpa kehilangan keumuman p > q, langkah selanjutnya adalah:
- (Pk+1Q)r(PkQ)q-r jika r > (q - r).

- Terapkan prosedur yang sama secara rekursif ke sub-urutan dengan panjang r dan (q - r), masing-masing, hingga r = 1 atau (q - r) = 1 berlaku.

Contoh

Langkah-langkah:

1. Tentukan titik awal dan akhir:

- Titik awal (x0, y0) = (0, 0)
- Titik akhir (x1, y1) = (82, 34)

2. Hitung dx, dy, dan gcd(dx, dy):

- dx = x1 - x0 = 82
- dy = y1 - y0 = 34
- gcd(dx, dy) = 2

3. Bagi dx dan dy dengan gcd(dx, dy):

- dx' = dx / gcd(dx, dy) = 41
- dy' = dy / gcd(dx, dy) = 17

4. Gunakan algoritma Bresenham untuk menggambar garis:

- Mulai dari (0, 0).
- Ulangi langkah berikut hingga mencapai (41, 17):
- Jika p >= 0, pilih H (langkah horizontal).
- Jika p < 0, pilih D (langkah diagonal).
- Perbarui p: p = p + 2dy' - dx'.
- Ulangi langkah 4 untuk menggambar garis dari (41, 17) ke (82, 34).

3.3 Extend the program GeneralPathCar.java for drawing the car of figure 2.10. Show the control points for the quadratic and cubic curves and connect the endings of the curves with their corresponding control points by dashed lines.

```java
import java.awt.*;
import java.awt.geom.*;

/**
* An example for the use of a GeneralPath to draw a car.
*
* @author Frank Klawonn
* Last change 07.01.2005
*/
public class GeneralPathCar extends Frame
{
  //Constructor
  GeneralPathCar()
  {
    //Enables the closing of the window.
    addWindowListener(new MyFinishWindow());
  }


  public void paint(Graphics g) {
    Graphics2D g2d = (Graphics2D) g;

    // Use antiallasing to have nicer lines.
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);

    // The lines should have a thickness of 3.0 instead of 1.0.
    BasicStroke bs = new BasicStroke(3.0f, BasicStroke.CAP_ROUND, BasicStroke.JOIN_ROUND);
    g2d.setStroke(bs);

    // Define the dash pattern for the stroke
    float[] dashPattern = {10, 5}; // 10 pixels drawn, 5 pixels skipped, repeated
    bs = new BasicStroke(3.0f, BasicStroke.CAP_ROUND, BasicStroke.JOIN_ROUND, 10, dashPattern, 0);
    g2d.setStroke(bs);

    // The GeneralPath to describe the car.
    GeneralPath gp = new GeneralPath();

    // Start at the lower front of the car.
    gp.moveTo(60, 120);
    // front underbody
    gp.lineTo(80, 120);
    // front wheel
    gp.quadTo(90, 140, 100, 120);
    // middle underbody
    gp.lineTo(160, 120);
    // rear wheel
    gp.quadTo(170, 140, 180, 120);
    // rear underbody
    gp.lineTo(200, 120);
    // rear
    gp.curveTo(195, 100, 200, 80, 160, 80);
    // roof
    gp.lineTo(110, 80);
    // windscreen
    gp.lineTo(90, 100);
    // bonnet
    gp.lineTo(60, 100);
    // front
    gp.lineTo(60, 120);

    // Draw the car.
    g2d.draw(gp);

    // Reset the stroke to solid line for the coordinate system
    g2d.setStroke(new BasicStroke(1.0f));

    // Draw a coordinate system.
    drawSimpleCoordinateSystem(200, 150, g2d);
}




  /**
  * Draws a coordinate system (according to the window coordinates).
  *
  * @param xmax     x-coordinate to which the x-axis should extend.
  * @param ymax     y-coordinate to which the y-axis should extend.
  * @param g2d      Graphics2D object for drawing.
  */
  public static void drawSimpleCoordinateSystem(int xmax, int ymax,
                                                Graphics2D g2d)
  {
    int xOffset = 30;
    int yOffset = 50;
    int step = 20;
    String s;
    //Remember the actual font.
    Font fo = g2d.getFont();
    //Use a small font.
    g2d.setFont(new Font("ARIAL",Font.PLAIN,9));
    //x-axis.
    g2d.drawLine(xOffset,yOffset,xmax,yOffset);
    //Marks and labels for the x-axis.
    for (int i=xOffset+step; i<=xmax; i=i+step)
    {
      g2d.drawLine(i,yOffset-2,i,yOffset+2);
      g2d.drawString(String.valueOf(i),i-7,yOffset-7);
    }

    //y-axis.
    g2d.drawLine(xOffset,yOffset,xOffset,ymax);

    //Marks and labels for the y-axis.
    s="  "; //for indention of numbers < 100
    for (int i=yOffset+step; i<=ymax; i=i+step)
    {
      g2d.drawLine(xOffset-2,i,xOffset+2,i);
      if (i>99){s="";}
      g2d.drawString(s+String.valueOf(i),xOffset-25,i+5);
    }

    //Reset to the original font.
    g2d.setFont(fo);
  }


   public static void main(String[] argv)
  {
    GeneralPathCar f = new GeneralPathCar();
    f.setTitle("General Path Car");
    f.setSize(500,500);
    f.setVisible(true);
    f.setLocationRelativeTo(null);
  }
}
```

OUTPUT