Computer Graphic

Java : Draw a Bezier curve

Starting code:

```
package tpimagereader;

import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.*;

public class TPImageReader {

   public static void main(String args[]) throws IOException {
      TPImageReader I = new TPImageReader();
   }

   public TPImageReader() throws IOException {
      init();
   }

   public void init() throws IOException {
      JFrame jf = new JFrame("Image Reader");
      SCanvas sc = new SCanvas();
      sc.setPreferredSize(new java.awt.Dimension(500, 500));

      jf.getContentPane().add(sc);
      jf.pack();
      jf.setResizable(true);
      jf.setVisible(true);
      jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      sc.setFocusable(true);
      sc.requestFocusInWindow();

   }

   private class SCanvas extends JComponent   {


      //Font font;
      public SCanvas() {
         super();

      }

      @Override
      public void paint(Graphics g_) {
         Graphics2D g = (Graphics2D) g_;

         // Fill the back ground
         g.setColor(Color.white);
         g.fillRect(0, 0, getWidth(), getHeight());

      }

   }
}
```

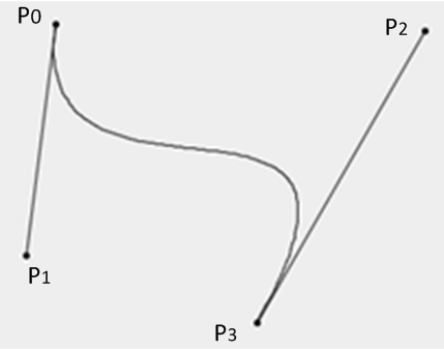
Draw a circle with connected lines thanks to the following parametric equation:

$$\begin{cases} f_x(t) = \mathrm{R}\cos t \\ f_y(t) = \mathrm{R}\sin t \end{cases} \quad , \mathrm{t} \in [0; 2\pi[$$

Extensions: draw an ellipse, a spiral

Draw a Bezier curve with connected lines thanks to the following parametric equation:

$$\begin{cases} f_x(t) = (1-t)^3 P_{0x} + 3t(1-t)^2 P_{1x} + 3t^2(1-t)P_{2x} + t^3 P_{3x} \\ f_y(t) = (1-t)^3 P_{0y} + 3t(1-t)^2 P_{1y} + 3t^2(1-t)P_{2y} + t^3 P_{3y} \end{cases} \quad , t \in [0;1[$$



Extensions: change the width of the curve

Use the mouse to change the location of the reference points

```
public class CourbeBezier extends  JComponent implements MouseListener, MouseMotionListener,
MouseWheelListener, KeyListener
```

Extensions: change the strength of the Bezier curve with the mouse wheel (changing the parameter 3 is not a good idea!)

Change the width of the line

```
  //Set the width of the line
     BasicStroke s = new BasicStroke(10);
     g2.setStroke(s);
```

add some antialiasing… google it to understand it…

```
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
```

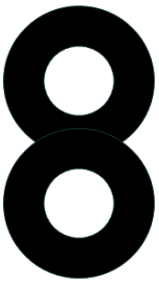
Draw a circle with a solid color thanks to small triangles (tessellation)

```
     Polygon p = new Polygon();
```

```
p.addPoint(0, 0);

p.addPoint(100, 0);

p.addPoint(100, 100);

g2.fillPolygon(p);
```

Draw a circle with a hole

Extensions: we can use some shape overlapping. Why is it harmful?



Draw a width Bezier curve with small triangles