



Trabalho 1: Implementação de Traçado de Retas
Professor Guido Militão

Orientações:

- Use a linguagem Java;
- Ao final deverá criar um .jar independente nomeado como TR1-P3-NOMES_DA_DUPLA.jar
- Compactar código fonte + jar com o nome TR1-P3-NOMES_DA_DUPLA.zip

Crie um novo projeto CG_2010

TRABALHO:

1. Crie uma classe *FrameDesenho*;
2. Crie uma classe *JPanelDesenho*;
3. Altere a classe *JPanelDesenho*:

```
public class JPanelDesenho extends javax.swing.JPanel {

    private static Integer MaxX;
    private static Integer MaxY;

    static void Pintaponto(Graphics g, int x, int y) {
        g.setColor(Color.black);
        g.fillRect(x, MaxY-y, 1, 1);
    }

    public JPanelDesenho() {
        MaxX = 300; // Pegar o tamanho maximo do Panel
        MaxY = 300;
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        setPreferredSize(new java.awt.Dimension(500, 500));

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 400, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 300, Short.MAX_VALUE)
        );
    } // </editor-fold>

    @Override
    protected void paintComponent(Graphics g) {
        g.setColor(Color.white);
        g.fillRect(0, 0, 400, 300);

        Pintaponto(g, 100, 100);
        Pintaponto(g, 30, 30);
        Pintaponto(g, 0, 0);

        for (int i=0; i<MaxX; i++)
        {
            Pintaponto(g, i, 10);Pintaponto(g, 10, i);
        }
    }
}
```

```
    }  
  
    }  
}
```

4. Altere o construtor da classe FrameDesenho:

```
public FrameDesenho() {  
    panel = new JPanelDesenho();  
    panel.setSize(300, 300);  
    initComponents();  
    this.add(panel);  
    pack();  
  
}
```

5. Implemente o algoritmo Scanline com dois parâmetros: Lista de vértices e cor . Utilize o algoritmo:

1. Obter a intersecção da linha de varredura com todos os lados do polígono;
2. Ordenar os pontos de intersecção (em x crescente);
3. Preencher os pixels entre pares de pontos de intersecção do polígono que são internos a ele, usando a regra de paridade

6. Implemente o algoritmo BoundaryFill com dois parâmetros: x,y. X,Y deverá ser interior a um polígono previamente desenhado.

Execute para o seguinte conjunto de vertices:

Para mais informações:

- <http://www.cse.unsw.edu.au/~cs3421/slides/bres/ScanLine.html>
- <http://www.netgraphics.sk/scan-line-algorithm>
- <http://www.netgraphics.sk/areas>