

- Pode ser necessário criar novos componentes para:
 - Exibição de dados (jogos) com formato específico;
 - Entrada de informações especializadas;
 - Exibir comportamento diferente dos componentes já existentes.
- Duas abordagens:
 - Criar componentes que herdam de outros, já existentes, com funções similares.
 - Criar novos componentes a partir de um componente genérico.

- Passos (nem todos são obrigatórios):
 - Herdar de classe que tem comportamento semelhante.
 - No construtor, chamar construtor ancestral, inicializar atributos relevantes e modificar comportamento através de métodos.
 - Sobreescrever métodos
`get{Maximum,Minimum,Preferred}Size()`.
 - Sobreescrever `paintComponent()`.

- Exemplo: peça para Reversi.
 - Botão com aparência e comportamento diferente.
 - Existem várias maneiras de implementar...

```
package reversi;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;

import javax.swing.JButton;

public class Peca extends JButton
{
    private static int tamanho = 64;
    private Estado estado;

    public Peca()
    {
        super();
        estado = Estado.VAZIO;
    }
}
```

```
package reversi;

public enum Estado { VAZIO, PRETO, BRANCO }
```

```
public Dimension getMaximumSize() { return getPreferredSize(); }
public Dimension getMinimumSize() { return getPreferredSize(); }
public Dimension getPreferredSize() { return new Dimension(tamanho,tamanho); }

protected void paintComponent(Graphics g)
{
    super.paintComponent(g);
    Graphics2D g2d = (Graphics2D)g;
    // Não preenchemos botões vazios.
    if (estado != Estado.VAZIO)
    {
        if (estado == Estado.BRANCO) g2d.setColor(Color.WHITE);
        else if (estado == Estado.PRETO) g2d.setColor(Color.BLACK);
        g2d.fillOval(6,6,getWidth()-12,getHeight()-12);
    }
    // Pintamos a borda da peça independente do estado.
    g2d.setColor(Color.GRAY);
    g2d.drawOval(6,6,getWidth()-12,getHeight()-12);
}
```

- Para mostrar a interface gráfica precisamos de:
 - Classe Peca, que representa botões para o jogo.
 - Classe Tabuleiro, que é um conjunto 8x8 de peças.
 - Classe Jogo, que é a aplicação que usa Tabuleiro.
- Um jogo funcional precisaria ainda de...
 - Regras do jogo, que possivelmente devem ser implementadas com código em todas as classes.
 - Mecanismo do adversário: rede (para adversário humano), inteligência artificial (para computador).

```
package reversi;

import java.awt.GridLayout;
import javax.swing.JPanel;

public class Tabuleiro extends JPanel
{
    private Peca[][] tabuleiro;

    public Tabuleiro()
    {
        setLayout(new GridLayout(8,8));
        tabuleiro = new Peca[8][8];
        for(int l=0;l<8;l++)
            for(int c=0;c<8;c++)
            {
                tabuleiro[c][l] = new Peca();
                add(tabuleiro[c][l]);
            }
        tabuleiro[3][3].setEstado(Estado.BRANCO);
        tabuleiro[4][4].setEstado(Estado.BRANCO);
        tabuleiro[3][4].setEstado(Estado.PRETO);
        tabuleiro[4][3].setEstado(Estado.PRETO);
    }
}
```

Criando Novos Componentes Gráficos

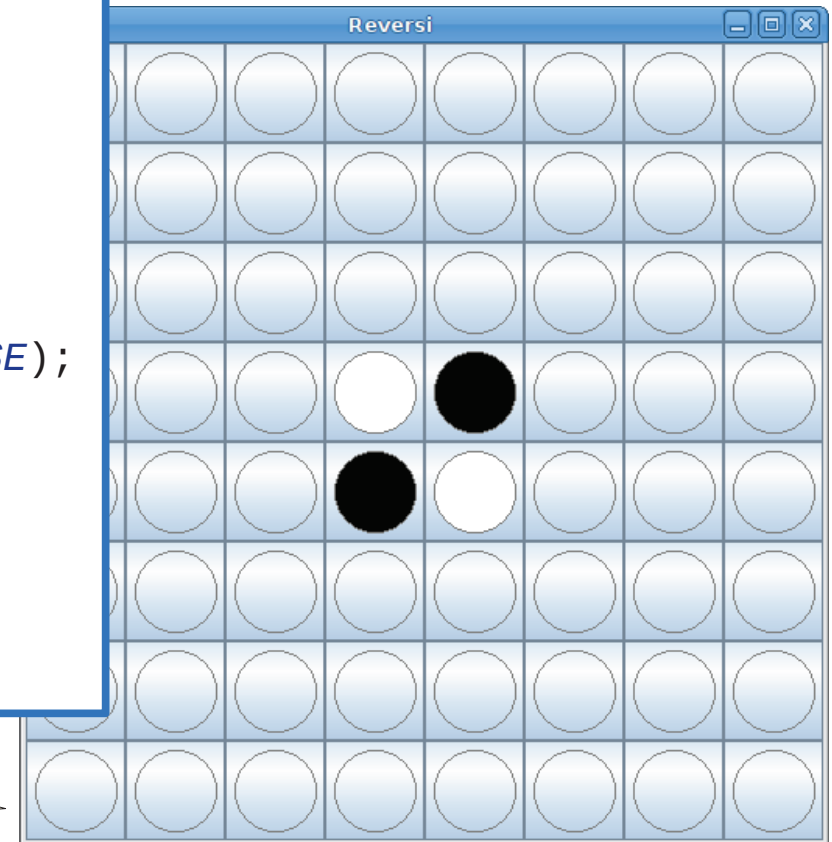


```
package reversi;

import javax.swing.JFrame;

public class Jogo extends JFrame
{
    public Jogo()
    {
        super("Reversi");
        getContentPane().add(new Tabuleiro());
        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args)
    {
        new Jogo();
    }
}
```



Dá pra melhorar a aparência?



Criando Novos Componentes Gráficos



```
package reversi;

import java.awt.*;
import java.awt.geom.Point2D;

import javax.swing.JButton;

public class PecaMelhor extends JButton
{
    private static int tamanho = 64;
    private Estado estado;

    public PecaMelhor()
    {
        super();
        setBackground(new Color(40,200,0));
        estado = Estado.VAZIO;
    }

    public void setEstado(Estado e) { estado = e; }

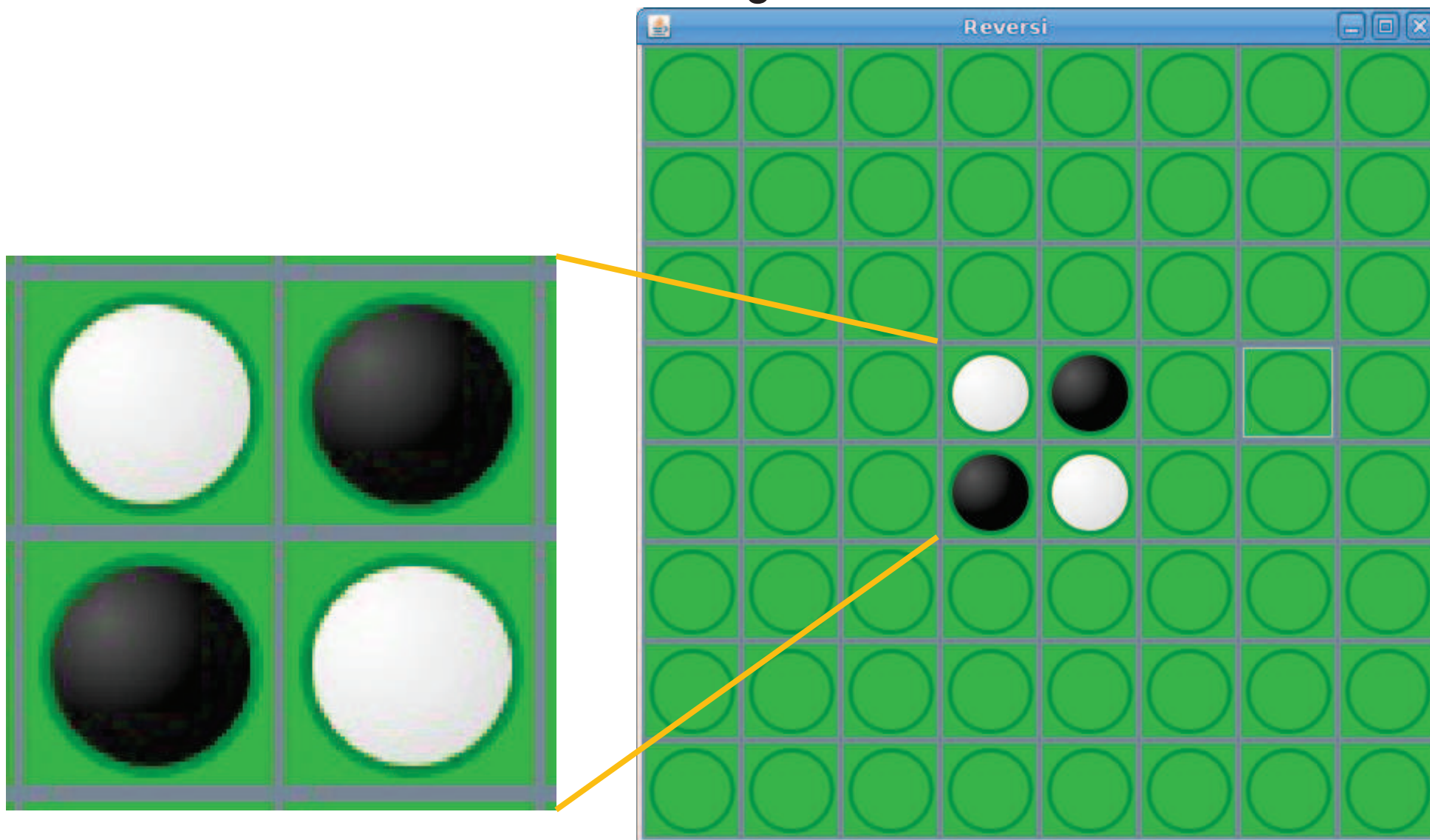
    public Dimension getMaximumSize() { return getPreferredSize(); }
    public Dimension getMinimumSize() { return getPreferredSize(); }
    public Dimension getPreferredSize() { return new Dimension(tamanho,tamanho); }
```




```
protected void paintComponent(Graphics g)
{
    super.paintComponent(g);
    Graphics2D g2d = (Graphics2D)g;
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);
    // Não preenchemos botões vazios.
    if (estado != Estado.VAZIO)
    {
        Color[] cores = new Color[2];
        if (estado == Estado.BRANCO)
            { cores[0] = Color.WHITE; cores[1] = new Color(220,220,220); }
        else if (estado == Estado.PRETO)
            { cores[0] = new Color(100,100,100); cores[1] = Color.BLACK; }
        RadialGradientPaint paint =
            new RadialGradientPaint(new Point2D.Double(tamanho/3,tamanho/3),
                2*tamanho/3,new float[]{0f,1f},cores);

        g2d.setPaint(paint);
        g2d.fillOval(6,6,getWidth()-12,getHeight()-12);
    }
    // Pintamos a borda da peça independente do estado.
    g2d.setColor(new Color(20,150,0));
    g2d.setStroke(new BasicStroke(3f));
    g2d.drawOval(6,6,getWidth()-12,getHeight()-12);
}
}
```

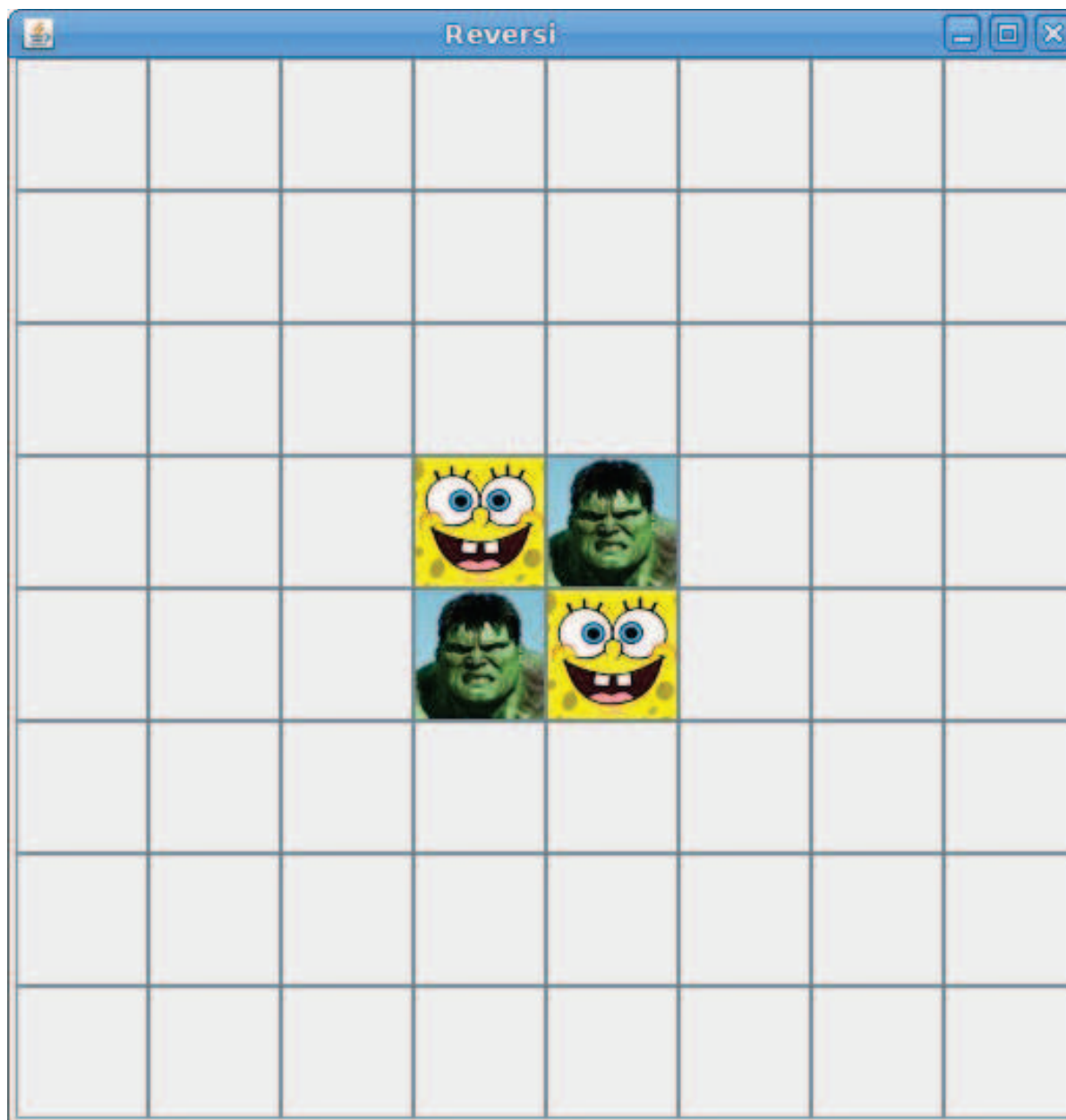
- Basta usar PecaMelhor no lugar de Peca em Tabuleiro.



```
...
public class PecaIcône extends JButton
{
    private static int tamanho = 64;
    private Estado estado;
    private Image i1,i2;

    public PecaIcône()
    {
        super();
        setContentAreaFilled(false);
        estado = Estado.VAZIO;
        i1 = new ImageIcon(getClass().getResource("/Sprites/sbob.jpg")).getImage();
        i2 = new ImageIcon(getClass().getResource("/Sprites/hulk.jpg")).getImage();
    } ...
    protected void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D)g;
        // Não preenchemos botões vazios.
        if (estado != Estado.VAZIO)
        {
            if (estado == Estado.BRANCO) g2d.drawImage(i1,0,0,null);
            else if (estado == Estado.PRETO) g2d.drawImage(i2,0,0,null);
        }
    }...
}
```

Criando Novos Componentes Gráficos



- Outro exemplo: componente que processa seus próprios eventos.

```
package rabisco;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import javax.swing.JComponent;

public class ComponenteParaRabiscos extends JComponent
    implements MouseListener, MouseMotionListener, KeyListener
{
    private ArrayList<Point> pontos;
    private int size = 8; private int halfsize = size/2;
    private Color cor;

    public ComponenteParaRabiscos(Color cor)
    {
        this.cor = cor;
        pontos = new ArrayList<Point>(1024);
        addMouseListener(this);
        addMouseMotionListener(this);
        addKeyListener(this);
        setFocusable(true);
        requestFocus();
    }
}
```

Pula?

```
protected void paintComponent(Graphics g)
{
    Graphics2D g2d = (Graphics2D)g;
    g2d.setColor(Color.WHITE);
    g2d.fillRect(0,0,getWidth(),getHeight());
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                          RenderingHints.VALUE_ANTIALIAS_ON);
    g2d.setColor(cor);
    for(Point p:pontos)
    {
        g2d.fillOval(p.x-halfsize,p.y-halfsize,size,size);
    }
}

public void mousePressed(MouseEvent e)
{
    pontos.add(e.getPoint());
    repaint();
}

public void mouseDragged(MouseEvent e)
{
    pontos.add(e.getPoint());
    repaint();
}
```

```
public void mouseEntered(MouseEvent e)
{
    requestFocus();
}

public void mouseReleased(MouseEvent e) { } // NOP
public void mouseClicked(MouseEvent e) { } // NOP
public void mouseExited(MouseEvent e) { } // NOP
public void mouseMoved(MouseEvent e) { } // NOP

public void keyPressed(KeyEvent e)
{
    System.out.println(e.getKeyCode());
    if (e.getKeyCode() == KeyEvent.VK_C)
    {
        pontos.clear();
        repaint();
    }
}

public void keyReleased(KeyEvent e) { } // NOP
public void keyTyped(KeyEvent e) { } // NOP
}
```

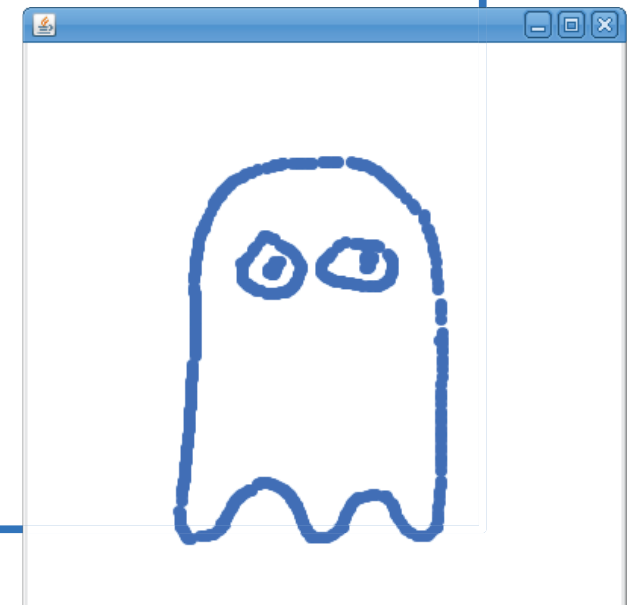
- Aplicação que usa o componente

```
package rabisco;

import java.awt.Color;
import java.awt.Container;
import javax.swing.JFrame;

public class AppRabiscos extends JFrame
{
    public AppRabiscos()
    {
        ComponenteParaRabiscos c = new ComponenteParaRabiscos(new Color(50,120,250));
        Container cp = getContentPane();
        cp.add(c);
        setSize(400,400);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args)
    {
        new AppRabiscos();
    }
}
```



- Outro exemplo: componente que produz e processa seus próprios eventos.

```
package exemplos;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ComponenteLuzVermelha extends JComponent implements ActionListener
{
    private int nível, passo;
    private Timer timer;

    public ComponenteLuzVermelha(int passo)
    {
        this.passo = passo;
        nível = 0;
        setPreferredSize(new Dimension(200, 200));
        timer = new Timer(50, this);
        timer.setCoalesce(true);
        timer.start();
    }
}
```

Pula?

```
protected void paintComponent(Graphics g)
{
    g.setColor(Color.WHITE);
    g.fillRect(0,0,getWidth(),getHeight());
    // Calculamos a cor de acordo com o passo.
    g.setColor(new Color(nível/100,0,0));
    g.fillArc(0,0,getWidth(),getHeight(),0,360);
}

public void actionPerformed(ActionEvent e)
{
    if (nível < 25500) nível += passo;
    repaint();
}
```

- Aplicação que usa o componente

```
package exemplos;

import java.awt.*;
import javax.swing.*;

public class LuzVermelhaApp extends JFrame
{
    public LuzVermelhaApp()
    {
        super("Luz Vermelha");
        getContentPane().setLayout(new GridLayout(2,2));
        ComponenteLuzVermelha c1,c2,c3,c4;
        c1 = new ComponenteLuzVermelha(10);    c2 = new ComponenteLuzVermelha(50);
        c3 = new ComponenteLuzVermelha(100);   c4 = new ComponenteLuzVermelha(250);
        getContentPane().add(c1); getContentPane().add(c2);
        getContentPane().add(c3); getContentPane().add(c4);
        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args)
    { new LuzVermelhaApp(); }
}
```

Criando Novos Componentes Gráficos

