

```

1 /*
2  * Copyright (C) 2016 redxef.
3  *
4  * This library is free software; you can redistribute it and/or
5  * modify it under the terms of the GNU Lesser General Public
6  * License as published by the Free Software Foundation; either
7  * version 2.1 of the License, or (at your option) any later version.
8  *
9  * This library is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
12 * Lesser General Public License for more details.
13 *
14 * You should have received a copy of the GNU Lesser General Public
15 * License along with this library; if not, write to the Free Software
16 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
17 * MA 02110-1301 USA
18 */
19 package powerspy.client;
20
21 import java.awt.*;
22 import java.awt.geom.*;
23 import javax.swing.*;
24
25 /**
26  *
27  * @author redxef
28  */
29 class ProgressCircleUI extends javax.swing.plaf.basic.BasicProgressbarUI {
30
31     private static final int ALPHA = 80;
32     private static final double ALPHA_RAD = ALPHA / 180.0 * Math.PI;
33     private static final int STARTANGLE = (180 - ALPHA) / 2 + 180;
34     private static final int ENDANGLE = -(180 - ALPHA) / 2;
35     private static final int RANGE = 360 - ALPHA;
36
37     private final int boldness;
38     private final Color bg;
39     private final Color fg;
40
41     /**
42      * Returns a new ProgressCircleUI Object in the PowerSpy design with the
43      * correct colours.
44      *
45      * @return the ProgrssCircleUI
46      */
47     public static ProgressCircleUI getPSDesign()
48     {
49         return new ProgressCircleUI(20, Color.LIGHT_GRAY, Defs.PS_ORANGE);
50     }
51
52     /**
53      * Constructs a new ProgrssCircleUI with the boldness of the arc, a
54      * foreground colour and a background colour.
55      *
56      * @param boldness the boldness of the arc

```

```

57     * @param bg         the background colour
58     * @param fg         the foreground colour
59     */
60     public ProgressCircleUI(int boldness, Color bg, Color fg)
61     {
62         this.boldness = boldness;
63         this.bg = bg;
64         this.fg = fg;
65     }
66
67     @Override
68     public Dimension getPreferredSize(JComponent c)
69     {
70         Dimension d = super.getPreferredSize(c);
71         int v = Math.max(d.width, d.height);
72         d.setSize(v, v);
73         return d;
74     }
75
76     @Override
77     public void paint(Graphics g, JComponent c)
78     {
79         Graphics2D g2d;
80
81         int width = progressBar.getWidth()
82             - progressBar.getInsets().right
83             - progressBar.getInsets().left;
84         int heigh = progressBar.getHeight()
85             - progressBar.getInsets().top
86             - progressBar.getInsets().bottom;
87
88         heigh += (int) (heigh / 2 * (1 - Math.cos(ALPHA_RAD / 2)));
89
90         int ratio = (progressBar.getValue() * RANGE) / 100;
91
92         if (width <= 0 || heigh <= 0)
93             return;
94
95         g2d = (Graphics2D) g.create();
96
97         Shape o0 = new Arc2D.Double(0, 0, width, heigh, STARTANGLE, -ratio);
98         Shape o1 = new Arc2D.Double(0, 0, width, heigh, STARTANGLE, -RANGE);
99         Shape i = new Ellipse2D.Double(boldness, boldness, width - 2 * boldness, heigh);
100        Area a0 = new Area(o0);
101        Area a1 = new Area(o1);
102        a0.subtract(new Area(i));
103        a1.subtract(new Area(i));
104
105        g2d.setColor(Color.LIGHT_GRAY);
106        g2d.fill(a1);
107        g2d.setColor(getSelectionForeground());
108        g2d.fill(a0);
109
110        if (progressBar.isStringPainted()) {
111            paintString(g, progressBar.getInsets().left,
112                progressBar.getInsets().top,
113                width, heigh, 0, progressBar.getInsets());

```

```
114         }
115     }
116
117     @Override
118     protected Color getSelectionForeground()
119     {
120         return Defs.PS_ORANGE;
121     }
122
123     @Override
124     protected Color getSelectionBackground()
125     {
126         return Defs.PS_ORANGE;
127     }
128 }
```