

```
1 package ticTacToe.component;
2
3 import java.awt.Color;
14
15 public class Button implements Paintable{
16
17     private Point position = null;
18     private Dimension dimension = null;
19     private boolean mouseOver = false;
20
21     public Button() {
22         this.position = new Point(0,0);
23         this.dimension = new Dimension(20,20);
24     }
25
26     public Button(int x, int y) {
27         this();
28         setPosition(x,y);
29     }
30
31     public Button(int x, int y,int width, int height) {
32         this(x,y);
33         setSize(width,height);
34     }
35
36     public void setPosition(int x, int y) {
37         this.position = new Point(x,y);
38     }
39
40     public Point getPosition() {
41         return (Point) this.position.clone();
42     }
43
44     public void setSize(int width, int height) {
45         this.dimension = new Dimension(width,height);
46     }
47
48     public Dimension getSize() {
49         return (Dimension) this.dimension.clone();
50     }
51
52     public int width() {
53         return this.dimension.width;
54     }
55
56     public int height() {
57         return this.dimension.height;
58     }
59
60     @Override
61     public void paint(Graphics g) {
62
63
64         Color corOriginal = g.getColor();
65
66         g.drawRect(position.x, position.y,
67                     dimension.width, dimension.height);
68
69         if (mouseOver == true) {
70             g.setColor(Color.RED);
71             g.drawRect(position.x + 2, position.y + 2, dimension.width - 4,
dimension.height - 4);
```

```
72         g.setColor(corOriginal);
73     }
74 }
75
76 public boolean isOver(Point point) {
77
78     int xLeft = this.position.x;
79     int yTop = this.position.y;
80     int xRight = xLeft + this.width();
81     int yBottom = yTop + this.height();
82
83     return ( (point.x > xLeft && point.x < xRight) &&
84             (point.y > yTop && point.y < yBottom) );
85 }
86
87 //--reacting to mouse
88 click-----
89
90 public MouseMotionListener mouseMotionListener() {
91     return new MouseMotionAdapter() {
92         @Override
93         public void mouseMoved(MouseEvent me) {
94             boolean dentro = isOver(me.getPoint());
95             if (mouseOver != dentro) {
96                 mouseOver = dentro;
97                 me.getComponent().repaint();
98             }
99         };
100     };
101
102     public MouseListener mouseMouseListener() {
103         return new MouseListenerAdapter() {
104             @Override
105             public void mouseEntered(MouseEvent e) {
106                 mouseOver = true;
107                 e.getComponent().repaint();
108             }
109
110             @Override
111             public void mouseExited(MouseEvent e) {
112                 mouseOver = false;
113                 e.getComponent().repaint();
114             }
115         };
116     }
117 }
118
```