

FlipFlop.java

```

1 // FlipFlop.java
2 // Raja Sooriamurthi
3 //
4 // An applet using the JDK 1.1 event model
5 // to play the game of "Lights Out"
6 //
7 // 1999/11/03 Initial version
8
9 /*
10 <applet code=FlipFlop.class height=500 width=500></applet>
11 */
12
13 import java.applet.*;
14 import java.awt.*;
15 import java.awt.event.*;
16
17 public class FlipFlop extends Applet {
18
19     int gameSize = 6;
20     int numButtons = gameSize*gameSize;
21     int buttonSize = 65;
22
23     private boolean[] state;
24     private Button[] buttons;
25
26     public void init() {
27
28         // all initialized to false
29         state = new boolean[numButtons];
30         buttons = new Button[numButtons];
31
32         setLayout (new BorderLayout());
33
34         Panel p0 = new Panel();
35         p0.setLayout(new FlowLayout(FlowLayout.CENTER));
36         Button startButton = new Button("Start");
37         startButton.setBackground(Color.blue);
38         startButton.setForeground(Color.yellow);
39         startButton.addActionListener (new StartButtonL());
40         p0.add(startButton);
41         // *** "North" is case sensitive !!
42         add("North", p0);
43
44         Panel p1 = new Panel();
45         p1.setLayout(new GridLayout(gameSize, gameSize));
46         for (int i = 0; i < numButtons; i++) {
47             buttons[i] = new Button(Integer.toString(i));
48             buttons[i].setSize (buttonSize, buttonSize);
49             buttons[i].addActionListener (new ButtonL());
50             p1.add(buttons[i]);
51             show(i);
52         }
53         // *** "Center" is case sensitive !!
54         add("Center", p1);
55
56         setSize (gameSize*buttonSize,
57                 gameSize*buttonSize + p0.getSize().height);
58         setBackground (Color.cyan);
59         repaint();
60     }
61
62     // *****
63     // A listener for the "start" button
64     // *****
65
66     class StartButtonL implements ActionListener {

```

FlipFlop.java

```

67     public void actionPerformed (ActionEvent e) {
68         for (int i=0; i<numButtons; i++)
69             if (Math.random() > 0.5) click(i);
70     }
71
72
73     // *****
74     // Listeners for each button
75     // *****
76
77     class ButtonL implements ActionListener {
78         public void actionPerformed (ActionEvent e) {
79             int i = Integer.parseInt (e.getActionCommand());
80             click (i);
81         }
82     }
83
84     // *****
85     // Click
86     // *****
87     // Click toggles button i as well as each of its
88     // neighbors on its four sides -if- they exist
89
90     private void click (int i) {
91         // toggle the clicked button
92         System.out.println ("Clicked " + i);
93         toggle(i);
94         // toggle the left
95         if (i%gameSize != 0) toggle(i-1);
96         // toggle the right
97         if (i%gameSize < gameSize-1) toggle(i+1);
98         // toggle the below
99         if (i ≥ gameSize) toggle(i-gameSize);
100        // toggle the above
101        if (i < numButtons-gameSize) toggle(i+gameSize);
102        repaint();
103    }
104
105    // *****
106    // toggle
107    // *****
108
109    private void toggle (int i) {
110        state[i] = -state[i];
111        show (i);
112    }
113
114    // *****
115    // show
116    // *****
117
118    private void show (int i) {
119        buttons[i].setBackground (state[i] ? Color.red: Color.green);
120    }
121 }

```