





```

1 import java.awt.*;
2 import java.awt.Color.*;
3 import javax.swing.*;
4 import java.awt.event.*;
5
6
7 public class KittyRun extends JFrame implements ActionListener{
8
9     private final int ROWS = 8;
10    private final int COLS = 8;
11    private final int GAP = 2;
12
13    private int xProg = 0;
14
15    private int kittyX = 0;
16    private int kittyY = 0;
17
18    private String kitty = ("^.^=");
19
20    private JButton up = new JButton("UP");
21    private JButton down = new JButton("down");
22    private JButton left = new JButton("LEFT");
23    private JButton right = new JButton("RIGHT");
24
25    private JPanel screen = new JPanel(new BorderLayout());
26    private JPanel pane = new JPanel(new GridLayout(ROWS, COLS, GAP, GAP))
27
28    private JPanel[][] panel = new JPanel[ROWS][COLS];
29    private JLabel[][] label = new JLabel[ROWS][COLS];
30    private Color color1 = Color.CYAN;
31    private Color color2 = Color.WHITE;
32    private Color tempColor;
33
34    public KittyRun() {
35
36        super("Run Kitty Run");
37        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38        screen.add(pane, BorderLayout.CENTER);
39        screen.add(up, BorderLayout.NORTH);
40        screen.add(down, BorderLayout.SOUTH);
41        screen.add(left, BorderLayout.WEST);
42        screen.add(right, BorderLayout.EAST);
43        add(screen);
44
45        up.addActionListener(this);
46        down.addActionListener(this);
47        left.addActionListener(this);
48        right.addActionListener(this);
49
50

```

```

51     for(int y = 0; y < ROWS; ++y) {
52         for(int x = 0; x < COLS; ++x) {
53             panel[x][y] = new JPanel();
54             label[x][y] = new JLabel();
55
56
57             panel[x][y].add(label[x][y]);
58             pane.add(panel[x][y]);
59
60             if(xProg != x) {
61                 ++ xProg;
62                 tempColor = color1;
63                 color1 = color2;
64                 color2 = tempColor;
65             }
66
67             if(y % 2 == 0) {
68                 panel[x][y].setBackground(color1);
69             }
70             else {
71                 panel[x][y].setBackground(color2);
72             }
73
74         }
75     }
76
77     label[kittyY][kittyX].setText(kitty);
78     //panel[kittyX][kittyY].remove(kitty);
79
80 }
81
82 @Override
83 public void actionPerformed(ActionEvent e) {
84     Object source = e.getSource();
85
86     if(source == up) {
87         if(kittyY > 0) {
88             label[kittyX][kittyY].setText("");
89             -- kittyY;
90             label[kittyX][kittyY].setText(kitty);
91         }
92     }
93     else if(source == down) {
94         if(kittyY < ROWS - 1) {
95             label[kittyX][kittyY].setText("");
96             ++ kittyY;
97             label[kittyX][kittyY].setText(kitty);
98         }
99     }

```

```

100         else if(source == left) {
101             if(kittyX > 0) {
102                 label[kittyX][kittyY].setText("");
103                 -- kittyX;
104                 label[kittyX][kittyY].setText(kitty);
105             }
106         }
107         else if(source == right) {
108             if(kittyX < COLS - 1) {
109                 label[kittyX][kittyY].setText("");
110                 ++ kittyX;
111                 label[kittyX][kittyY].setText(kitty);
112             }
113         }
114     }
115 }
116
117
118 public static void main(String[] args) {
119     KittyRun frame = new KittyRun();
120     frame.setSize(500,400);//width then height
121     frame.setVisible(true);
122 }
123
124
125
126 }

```