

LandHex.java

```
1 package catan;
2
3 import java.awt.Color;
4
5 /**
6  * This class holds the methods and variables available to Land
7  * Hexes for the Catan board.
8  * Contains a built in GUI, unusable without the user interface.
9  *
10 *
11 * @author Eddie Gurnee
12 * @author Nicole Downer
13 * @version 0.0.08 10/28/2013
14 *
15 */
16 public class LandHex extends Hex {
17     /**
18      * This class holds the methods and variables available to the
19      * number tokens on Land Hexes for the Catan board.
20      *
21      * @author Eddie Gurnee
22      * @version 0.0.05 10/22/2013
23      *
24      */
25     private class NumberToken extends BoardSpace {
26         private String theNum;
27
28         private final int HEIGHT = 30;
29         private final int WIDTH = 30;
30
31         private NumberToken (int numProduces) {
32             super();
33             setSize(WIDTH, HEIGHT);
34
35             this.theNum = "" + numProduces;
36         }
37         @Override
38         public void onLanding() {
39             BoardGame.board.theRobber.place(LandHex.this);
40         }
41         @Override
42         public void paintComponent(Graphics g) {
43             super.paintComponent(g);
```

LandHex.java

```
43         g.setColor(new Color(223, 180, 160));
44         g.fillOval(0, 0, WIDTH, HEIGHT);
45
46         g.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 12));
47         g.setColor(Color.BLACK);
48         g.drawString(theNum, 10, 20);
49     }
50     @Override
51     public void actionPerformed(ActionEvent e) {
52         if (!BoardGame.board.theRobber.isPlaced()) {
53             this.onLanding();
54         }
55     }
56 }
57
58 private int produceOn;
59 private boolean produces;
60 private int resource;
61 private Color bgColor;
62 private NumberToken numObject;
63
64 public LandHex(String type, int produceOn) {
65     super(type);
66     this.produceOn = produceOn;
67     this.produces = true;
68
69     this.numObject = new NumberToken(produceOn);
70
71     numObject.setBounds(
72         ((this.getWidth() - numObject.getWidth()) / 2),
73         ((this.getHeight() - numObject.getHeight()) / 2),
74         numObject.getWidth(),
75         numObject.getHeight());
76
77     this.add(numObject);
78
79     switch(type) {
80     case "Wood":
81         bgColor = new Color(150, 111, 51);
82         setResource(0);
83         break;
```

LandHex.java

```
84         case "Sheep":
85             bgColor = new Color(102, 166, 102);
86             setResource(1);
87             break;
88         case "Wheat":
89             bgColor = new Color(245, 222, 179);
90             setResource(2);
91             break;
92         case "Ore":
93             bgColor = new Color(212, 215, 217);
94             setResource(3);
95             break;
96         case "Brick":
97             bgColor = new Color(142, 35, 35);
98             setResource(4);
99             break;
100        case "Desert":
101            bgColor = new Color(254, 232, 214);
102            this.remove(numObject);
103            break;
104        }
105    }
106    @Override
107    public void paintComponent(Graphics g) {
108        super.paintComponent(g);
109
110        g.setColor(bgColor);
111        g.fillPolygon(this.getHex());
112    }
113    public boolean isProduces() {
114        return produces;
115    }
116    public void setProduces(boolean produces) {
117        this.produces = produces;
118    }
119    public void produce() {
120        if (produces) {
121            for (Properties a : this.getProperties()) {
122                a.getOwner().addTempResource(resource,
123                a.getValue());
124            }
125        }
126    }
```

LandHex.java

```
124     }
125 }
126 public int getProduceOn() {
127     return produceOn;
128 }
129 public int getResource() {
130     return resource;
131 }
132 private void setResource(int resource) {
133     this.resource = resource;
134 }
135 }
```