

BoardPanel.java

```
1 import java.awt.*;
7
8 public class BoardPanel extends JPanel implements KeyListener {
9     private Player player;
10    private ArrayList<Monster> monsters;
11    private Grid grid;
12    private final int cellWidth = 35;
13    private final int cellHeight = 35;
14    private final int Lmargin = 100;
15    private final int Tmargin = 40;
16
17    public BoardPanel(Grid grid, Player player,
18        ArrayList<Monster> monsters) {
19        this.player = player;
20        this.grid = grid;
21        this.monsters = monsters;
22    }
23
24    // reset game
25    public void reset(Grid grid, Player player,
26        ArrayList<Monster> monsters) {
27        this.player = player;
28        this.grid = grid;
29        this.monsters = monsters;
30    }
31    /* responds to various Keyboard pressed */
32    @Override
33    public void keyPressed(KeyEvent ke) {
34        if (ke.getKeyCode() == KeyEvent.VK_LEFT) {
35            if (player.getDirection() != 'L') {
36                player.clearPress();
37            }
38            player.setDirection('L');
39            player.addPress();
40        }
41        if (ke.getKeyCode() == KeyEvent.VK_RIGHT) {
42            if (player.getDirection() != 'R') {
43                player.clearPress();
44            }
45            player.setDirection('R');
```

BoardPanel.java

```
46         player.addPress();
47     }
48     if (ke.getKeyCode() == KeyEvent.VK_UP) {
49         if (player.getDirection() != 'U') {
50             player.clearPress();
51         }
52         player.setDirection('U');
53         player.addPress();
54     }
55     if (ke.getKeyCode() == KeyEvent.VK_DOWN) {
56         if (player.getDirection() != 'D') {
57             player.clearPress();
58         }
59         player.setDirection('D');
60         player.addPress();
61     }
62     if (ke.getKeyCode() == KeyEvent.VK_Z) {
63         player.putTrap();
64     }
65     if (ke.getKeyCode() == KeyEvent.VK_X) {
66         player.putBlock();
67     }
68 }
69
70 @Override
71 public void keyReleased(KeyEvent ke) {
72 }
73
74 @Override
75 public void keyTyped(KeyEvent e) {
76 }
77
78 /* returns the x coordinate based on left margin and cell
width */
79 private int xCor(int col) {
80     return Lmargin + col * cellWidth;
81 }
82
83 /* returns the y coordinate based on top margin and cell
height */
84 private int yCor(int row) {
85     return Tmargin + row * cellHeight;
```

BoardPanel.java

```
86     }
87
88     /*
89     * Redraws the board and the pieces Called initially and in
    response to
90     * repaint()
91     */
92     protected void paintComponent(Graphics graphics) {
93         super.paintComponent(graphics);
94         Cell cells[] = grid.getAllCells();
95         Cell cell;
96         for (int i = 0; i < cells.length; i++) {
97             cell = cells[i];
98             if (cell.col % 5 == 0 && cell.row % 5 == 0)
99                 graphics.setColor(Color.cyan);
100             else
101                 graphics.setColor(Color.white);
102             graphics.fillRect(xCor(cell.col), yCor(cell.row),
    cellWidth, cellHeight);
103             graphics.setColor(Color.black);
104             graphics.drawRect(xCor(cell.col), yCor(cell.row),
    cellWidth, cellHeight);
105             if (cell.gotGold) {
106                 graphics.setColor(Color.MAGENTA);
107                 graphics.fillArc(xCor(cell.col) + cellWidth / 8,
    yCor(cell.row) + cellHeight / 8, cellWidth * 3 / 4,
108                 cellHeight * 3 / 4, 45, 45);
109                 graphics.setColor(Color.white);
110                 graphics.drawString("G", xCor(cell.col) +
    cellWidth / 3, yCor(cell.row) + 2 * cellWidth / 3);
111             }
112         }
113         cell = player.getCell();
114         graphics.setColor(Color.red);
115         graphics.fillOval(xCor(cell.col) + cellWidth / 8,
    yCor(cell.row) + cellHeight / 8, cellWidth * 3 / 4,
116             cellHeight * 3 / 4);
117         graphics.setColor(Color.white);
118         graphics.drawString("P", xCor(cell.col) + cellWidth / 3,
    yCor(cell.row) + 2 * cellWidth / 3);
119
120         for (Trap trap : player.getTrap()) {
```

BoardPanel.java

```
121         if (trap.getState()) {
122             cell = trap.getCell();
123             graphics.setColor(Color.green);
124             graphics.fillRect(xCor(cell.col),
125 yCor(cell.row), cellWidth, cellHeight);
125             graphics.setColor(Color.white);
126             graphics.drawString("T", xCor(cell.col) +
127 cellWidth / 3, yCor(cell.row) + 2 * cellWidth / 3);
128         }
129
130         for (Roadblock roadblock : player.getBlock()) {
131             if (roadblock.getState()) {
132                 cell = roadblock.getCell();
133                 graphics.setColor(Color.blue);
134                 graphics.fillRect(xCor(cell.col),
135 yCor(cell.row), cellWidth, cellHeight);
135                 graphics.setColor(Color.white);
136                 graphics.drawString("B", xCor(cell.col) +
137 cellWidth / 3, yCor(cell.row) + 2 * cellWidth / 3);
138             }
139
140             for (Monster monster : monsters) {
141                 cell = monster.getCell();
142                 if (monster.viewable() && !monster.isBaby()) {
143                     graphics.setColor(Color.black);
144                     graphics.fill3DRect(xCor(cell.col) + cellWidth /
145 8, yCor(cell.row) + cellHeight / 8, cellWidth * 3 / 4,
146 cellHeight * 3 / 4, true);
145                     graphics.setColor(Color.white);
146                     graphics.drawString("M", xCor(cell.col) +
147 cellWidth / 3, yCor(cell.row) + 2 * cellWidth / 3);
147                 } else if (monster.viewable() && monster.isBaby()) {
148                     graphics.setColor(Color.yellow);
149                     graphics.fill3DRect(xCor(cell.col) + cellWidth /
150 8, yCor(cell.row) + cellHeight / 8, cellWidth * 3 / 4,
151 cellHeight * 3 / 4, true);
151                     graphics.setColor(Color.white);
152                     graphics.drawString("B", xCor(cell.col) +
153 cellWidth / 3, yCor(cell.row) + 2 * cellWidth / 3);
154                 }
155             }
156         }
157     }
158 }
```

BoardPanel.java

```
155     }  
156 }  
157
```