

CISC 275: Introduction to Software Engineering

Layered Panes

Lab 8

Charlie Greenbacker
University of Delaware
Fall 2011

Overview

- What are Layered Panes?
- Quick demo
- How to use Layered Panes
- Going forward...

Material adapted from:

<http://download.oracle.com/javase/tutorial/uiswing/components/layeredpane.html>

What are Layered Panes?

- Swing container with 3rd dimension:
 - depth (aka “Z order”)
- When adding components to layered pane, specify depth as integer
 - higher integer = closer to “top”
- Use class: JLayeredPane

Quick demo

- LayeredPaneDemo
 - LayeredPaneDemo.java
 - dukeWaveRed.gif
 - (keep in same src folder in Eclipse)
- Mouse drags image of Duke in front of & behind colored labels
- Use combo box to change Duke's depth

How to use Layered Panes

- (code from LayeredPaneDemo.java)
- Creating the layered pane:

```
layeredPane = new JLayeredPane();  
layeredPane.setPreferredSize(new Dimension(300, 310));  
layeredPane.setBorder(BorderFactory.createTitledBorder(  
    "Move the Mouse to Move Duke"));  
layeredPane.addMouseMotionListener(this);
```

How to use Layered Panes

- Adding components & setting depth:

```
for (int i = 0; i < layerStrings.length; i++) {  
    JLabel label = createColoredLabel(layerStrings[i],  
                                      layerColors[i], origin);  
    layeredPane.add(label, new Integer(i));  
    origin.x += offset;  
    origin.y += offset;  
}
```

How to use Layered Panes

- Setting a component's position within its depth:

```
final ImageIcon icon = createImageIcon("dukeWaveRed.gif");  
...  
dukeLabel = new JLabel(icon);  
...  
dukeLabel.setBounds(15, 225,  
                    icon.getIconWidth(),  
                    icon.getIconHeight());  
...  
layeredPane.add(dukeLabel, new Integer(2), 0);
```


How to use Layered Panes

- Laying out components in a layered pane...
- No layout manager by default; manually:

```
dukeLabel.setBounds(15, 225,  
                    icon.getIconWidth(),  
                    icon.getIconHeight());  
...  
label.setBounds(origin.x, origin.y, 140, 140);
```

- Moving mouse calls setPosition():

```
dukeLabel.setLocation(e.getX()-XFUDGE, e.getY()-YFUDGE);
```


Going forward...

- Making layered panes transparent:

```
label.setOpaque(false);
```

- Changing layered pane color:

```
label.setBackground(Color.green);
```

Going forward...

- More ideas:
- Drawing arbitrary shapes (possibly in individual layered panes?)
<http://download.oracle.com/javase/tutorial/2d/geometry/arbitrary.html>
- Using maps with JXMapView
<http://today.java.net/pub/a/today/2007/10/30/building-maps-into-swing-app-with-jxmapviewer.html>
- Using SVG images in Java:
<http://java.sun.com/developer/technicalArticles/GUI/svg/>