25 Oct 2005 GUIS —Graphical User Interfaces

Read Chap. 17 of the text. The ProgramLive CD is a better way to learn about GUIs. See the CD for examples of code.

Their mouse had a mean time between failure of … a week, at which time it wou jam up irreparably, or ... It would jam up on the table-- ... It had a flimsy cord whose wires would break. Steve Jobs said "... Xerox says it can't be built for less than $400, but I want a $10 mouse that will never fail and can be mass-produced because it's going to be the primary interface of the computer of the future."

... Dean Hovey ... came back and said, "I've got some good news and some bad news. The good news is, we've got a new project with Apple. The bad news is, I told Steve we'd design him a mouse for ten bucks."

... A year … later ... we had a design, filed a patent, and were granted a patent, c the electro-mechanical-optical mouse of today, which is still the reference design for PC mice. ... and ... we ended up ... [making] the mouse as invisible to people as it is today.

Interview with Steve Sachs on Apple and the Mouse in 1979 and the first computer with a GUI, the Apple Lisa (about $9,999 in about 1982).
http://library.stanford.edu/mac/primary/interviews/sachs/trans.html

Basic Components

Component
- Button, Canvas
- Checkbox, Choice
- Label, List, Scrollbar
- TextComponent
  - TextField, TextArea
- Container
  - JComponent
  - AbstractButton
    - JButton
    - JToggleButton
      - JCheckBox
      - RadioButton
  - JLabel, JList
  - JOptionPane, JPanel
  - JPopupMenu, JScrollBar, JSlider
  - JTextComponent
  - JTextField, JTextArea

Component: Something that can be placed in a GUI window. These are the basic ones that one uses in a GUI

Note the use of subclasses to provide structure and efficiency. For example, there are two kinds of JToggleButtons, so that class has two subclasses.

Components that can contain other components

Component
- Box
- Container
  - JComponent
  - JPanel
  - Panel
    - Applet
- Window
  - Frame
    - JFrame
  - JWindow

java.awt is the old GUI package.

javax.swing is the new GUI package. When they wanted to use an old name, they put J in front of it.

(e.g. Frame and JFrame)

When constructing javax.swing, the attempt was made to rely on the old package as much as possible.

So, JFrame is a subclass of Frame.

But they couldn't do this with Jpanel.

What components can go in a JFrame

Packages that contain classes that deal with GUIs:
java.awt: Old package. **javax.swing: New package.**

Javax.swing has a better way of listening to buttons, text fields, etc. It's components are more flexible.

Component: Something that can be placed in a GUI window. They are instances of certain classes, e.g.

JButton, Button:	Clickable button
JLabel, Label:	Line of text
JTextField, TextField:	Field into which the user can type:
JTextArea, TextArea:	Many-row field into which user can type
JPanel, Panel:	Used for graphics; to contain other components
JCheckBox:	Checkable box with a title
JComboBox:	Menu of items, one of which can be checked
JRadioButton:	Same functionality as JCheckBox
Container:	Can contain other components
Box:	Can contain other components

Putting components in a JFrame

```
import java.awt.*;
import javax.swing.*;
/** Demonstrate placement of components in a JFrame. Use BorderLayout.
   It places five components in the five possible areas:
     (1) a JButton in the east,
     (2) a JLabel in the west,
     (3) a JLabel in the south,
     (4) a JTextField in the north, and
     (5) a JTextArea in the center.   */

public class ComponentExample extends JFrame {
   /** Constructor: a window with title t and 5 components */
   public ComponentExample(String t) {
      super(t);
      Container cp=  getContentPane();
      cp.add(new JButton("click me"), BorderLayout.EAST);
      cp.add(new JTextField("type here", 22), BorderLayout.NORTH);
      cp.add(new JLabel("label 1"), BorderLayout.SOUTH);
      cp.add(new JLabel("label 2"), BorderLayout.WEST);
      cp.add(new JTextArea("type\nhere", 4, 10), BorderLayout.CENTER);
      pack();
   }
```

JFrame's content pane

Layout manager: An instance controls the placement of components.

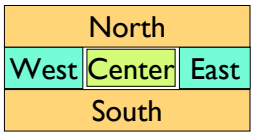
JFrame layout manager default: BorderLayout.

BorderLayout layout manager: Can place 5 components:

```
Container cp= getContentPane();
JButton jb= new JButton("Click here");
JButton jl= new JLabel( "label 2");

cp.add(jb, BorderLayout.EAST);
cp.add(jl, BorderLayout.WEST);

pack();
setVisible(true);
```



Jpanel as a container

```
/** Instance has labels in north /south, JPanel with four
    buttons in center.  */
public class PanelDemo extends JFrame {
    JPanel p= new JPanel();
    /** Constructor: an invisible frame with title t, two labels in
        north and south and a JPanel of 4 buttons in the center */
    public PanelDemo(String t) {
        super(t);
    p.add(new JButton("0"));    p.add(new JButton("1"));
    p.add(new JButton("2"));    p.add(new JButton("3"));
    Container cp= getContentPane();
    cp.add(new JLabel("north"), BorderLayout.NORTH);
    cp.add(new JLabel("south"), BorderLayout.SOUTH);
    cp.add(p, BorderLayout.CENTER);
    pack();  show();
    }
}
```

JPanel layout manager default: FlowLayout

FlowLayout layout manager: Place any number o components. They appear in the order in which the were added, taking as many rows as necessary

Class Box: a container

```
/** Demo class Box. Commen on constructor says how frame is laid out */
public class BoxDemo extends JFrame {
    /** Constructor: frame with title t, labels in the east / west, blank label in
        south, horizontal Box with three buttons in center. */
    public BoxDemo(String t) {
      super(t);
      Box b= new Box(BoxLayout.X_AXIS);
      b.add(new JButton("first"));     b.add(new JButton("second"));
      b.add(new JButton("third"));
      Container cp= getContentPane();
      cp.add(new JLabel("WEST Label"), BorderLayout.WEST);
      cp.add(new JLabel("EAST Label"), BorderLayout.EAST);
      cp.add(new JLabel(" "), BorderLayout.SOUTH);
      cp.add(b, BorderLayout.CENTER);
      pack();
    }
}
```

Box layout manager default: BoxLayou

BoxLayout layout manager: Place any numb of components. They appear in the order which they were added, taking only one ro

Boxes within a Box

3 vertical boxes, each column of buttons, are placed in a horizontal box

```
public class BoxDemo2 extends JFrame {
 /** Constructor: frame with title t and 3 columns with n, n+1, and n+2 buttons. */
 public BoxDemo2(String t, int n) {
    super(t);
    // Create Box b1 with n buttons.
        Box b1= new Box(BoxLayout.Y_AXIS);
        for (int i= 0;  i != n;  i= i+1)
            b1.add(new JButton("1 " + i));
    // Create Box b2 with n+1 buttons.
        Box b2= …
    // Create Box b3 with n+2 buttons.
        Box b3= …
    // Create horizontal box b containing b1, b2, b3
        Box b= new Box(BoxLayout.X_AXIS);
        b.add(b1);
        b.add(b2);
        b.add(b3);
    Container cp= getContentPane();
    cp.add(b, BorderLayout.CENTER);
    pack();  show();
}
```

BoxLayout layout manager: Place any number of components. They appear in the order in which they were added, taking only one row.

Changing Layout Managers

You can change the layout manager of a JFrame --but not to a BoxLayout manager.

To change it to a flow layout, use:

```
getContentPane.setLayout(new FlowLayout());
```

To simulate using a BoxLayout manager for a JFrame, create a Box and place it as the sole component of the JFrame:

```
JFrame jf= new JFrame("title");
Box b= new Box(BoxLayout.X_AXIS);
Add components to b;
jf.add(b,BorderLayout.CENTER);
```