GUI: Graphical User Interfaces

Their mouse had a mean time between failure of ... a week ... it would jam up irreparably, or ... jam up on the table -- ... It had a flimsy cord whose wires would break. Steve Jobs: "... Xerox says it can't be built for < $400, 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 ..."

... Dean Hovey ... came back, "I've got some good and some bad news. Good news: we've got a new project with Apple. Bad news: I told Steve we'd design a mouse for 10 bucks."

... year later ... we ... filed ... and were granted a patent, on the electro-mechanical-optical mouse of today; ... we ended up ... [making] the mouse as invisible to people as it is today.


GUI (Graphical User Interface)

• Provides a friendly interface between user and program
• Allows event-driven or reactive programming: The program reacts to events such as button clicks, mouse movement, keyboard input
• Often is multi-threaded: Different threads of execution can be executing simultaneously. We study concurrency and threads in April.

Two aspects to making a GUI:
1. Placing components (buttons, text, etc.) in it.
2. Listening/responding to events

Lecture notes page of course website. rows for GUI lectures:
Contains guiDemo.zip. It’s filled with short demos of GUI features including demos for today and next lecture. Download it and look at demos in DrJava or Eclipse.

GUI (Graphical User Interface)

There are three GUI packages in Java:
• AWT (Abstract or Awful Window Toolkit) — first one. Some parts are implemented not in Java but in code that depends on the platform. Came with first Java.
• Swing — a newer one, which builds on AWT as much as possible. It is "lightweight": all code written as Java classes/interfaces. Released in 97-98.
• JavaFX — completely new! Much more functionality, flexibility, but far too complicated to teach in CS2110. (Released first in 2008)

We use Swing (and parts of AWT)

Class JFrame

JFrame object: associated with a window on your monitor.

Generally, a GUI is a JFrame object with various components placed in it

Some methods in a JFrame object
hide() show() setVisible(boolean)
getWidth() getHeight() getLocation(int, int)
getTitle() setLocation(String)

Over 100 methods in a JFrame object!

Class JFrame is in package javax.swing

Placing components in a JFrame

Layout manager: Instance controls placement of components.

JFrame layout manager default: BorderLayout.

BorderLayout layout manager: Can place 5 components:

```java
public class C extends JFrame {
    public C() {
        JButton jb = new JButton("Click here");
        JLabel jl = new JLabel("west");
        add(jb, BorderLayout.WEST);
        add(jl, BorderLayout.CENTER);
        add(new JLabel("south"), BorderLayout.SOUTH);
        add(new JLabel("center"), BorderLayout.CENTER);
        add(new JLabel("north"), BorderLayout.NORTH);
        pack();
    }
}
```

JFrameDemo.java
Putting components in a JFrame

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

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

Packages -- Components

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

javax.swing has a better way of listening to buttons, text fields, etc. Components are more flexible.

Component: Something that can be placed in a GUI window. They are instances of certain classes, e.g.
- JButton: Clickable button
- JLabel: Line of text
- JTextField, TextArea: Field into which user can type
- JCheckBox: Checkable box with a title
- JComboBox: Menu of items, one of which can be checked
- JPanel: Used for graphics; to contain other components

Put scrollbars around JTextArea
- JScrollPane: Scrolls around a JTextArea
- JScrollBar: Vertical or horizontal scrollbars

Containers: Can contain other components
- Container
- Box: Can contain other components

Basic Components

- Component: Something that can be placed in a GUI window. These are the basic ones used in GUIs

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

Components that can contain other components

- java.awt is the old GUI package.
- javax.swing is the newer GUI package.

When they wanted to use an old name, they put J in front of it.
- JFrame is a subclass of Frame.

Put scrollbars around JTextArea
- JScrollPane: Scrolls around a JTextArea
- JScrollBar: Vertical or horizontal scrollbars

Containers: Can contain other components
- Container
- Box: Can contain other components

JPanel as a container

PanelDemo extends JFrame {  
    // panelDemo is a frame with title "Panel demo", labels in east/west, 
    // blank label in south, JPanel of 4 buttons in the center
    public PanelDemo() {
        super("Panel demo");
        p.add(new JButton("0"));  p.add(new JButton("1"));
        p.add(new JButton("2"));  p.add(new JButton("3"));
        add(new JLabel("east"), BorderLayout.EAST);
        add(new JLabel("west"), BorderLayout.WEST);
        add(new JLabel("s"), BorderLayout.SOUTH);
        add(p, BorderLayout.CENTER);
        pack();
    }
}
```
import java.awt.*;
import javax.swing.*;

/** Demo class Box. Comment on constructor says how frame is laid out. */
public class BoxDemo extends JFrame {
    /** Constructor: frame with title "Box demo", labels in the east/west,
     * blank label in south, horizontal Box with 4 buttons in center. */
    public BoxDemo() {
        super("Box demo");
        Box b = new Box(BoxLayout.X_AXIS);
        b.add(new JButton("0"));
        b.add(new JButton("1"));
        b.add(new JButton("2"));
        b.add(new JButton("3"));
        add(new JLabel("east"), BorderLayout.EAST);
        add(new JLabel("west"), BorderLayout.WEST);
        add(new JLabel(" "), BorderLayout.SOUTH);
        add(b, BorderLayout.CENTER);
        pack(); show();
    }
}

Class Box: a container

Box layout manager default: BoxLayout.

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

Simulate BoxLayout Manager in a JFrame

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);

1. Start developing a GUI by changing an already existing one. A lot
   of details. Hard to get all details right when one starts from scratch and
   has little idea about the Java GUI package.

1. Showed how to place components in a GUI. Next time: how to
   "listen" to things like button clicks in a GUI.