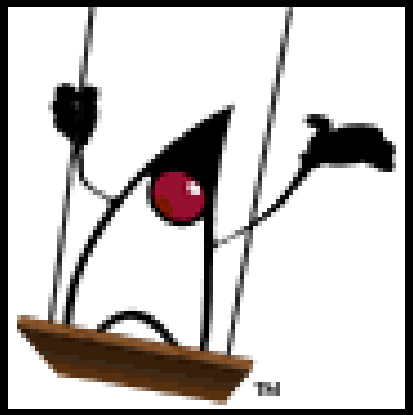
Java & Swing

Application development



Brian Amento, Gary Zamchick, NYU Spring ’06

Resources

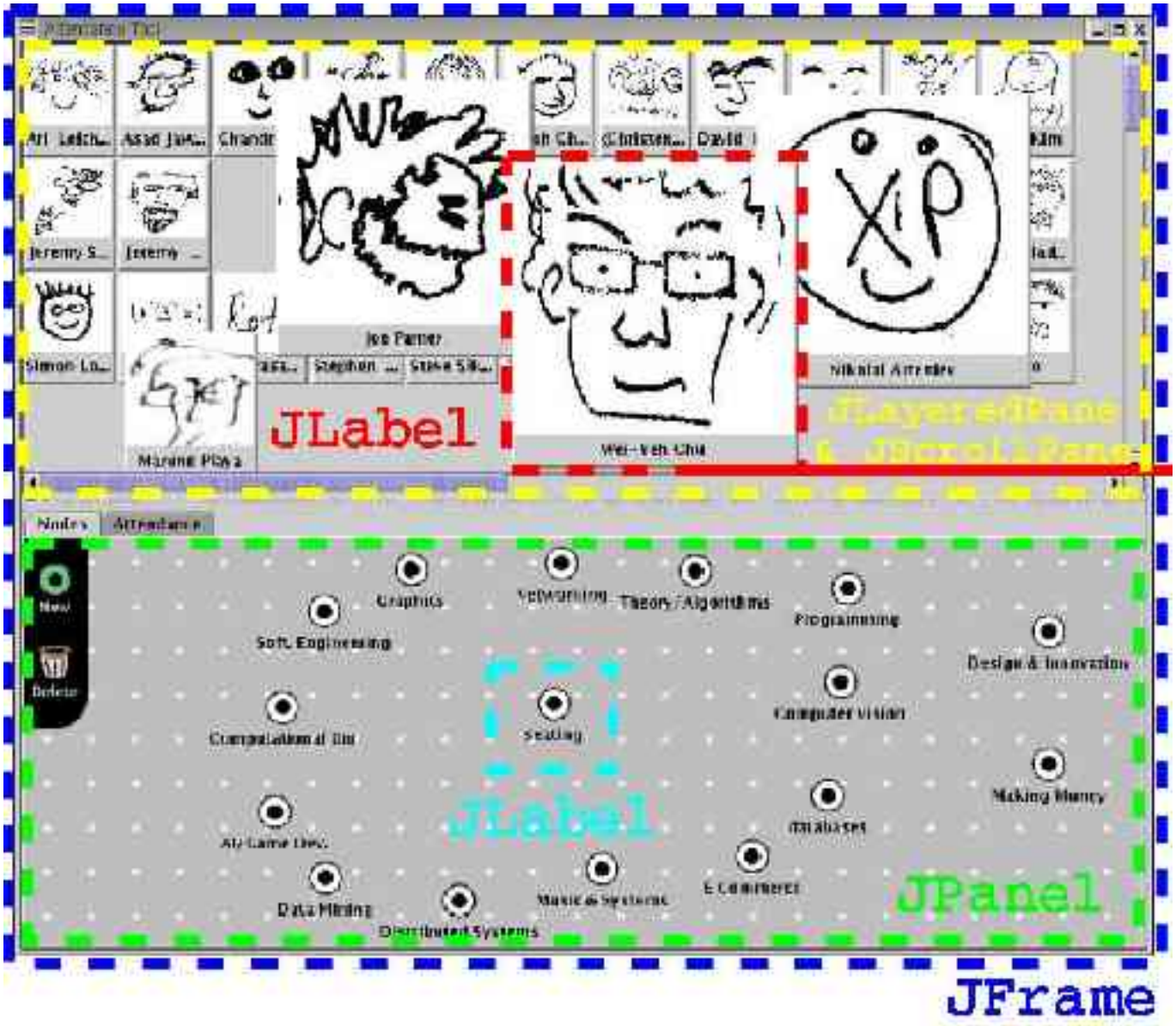
Important Online Resources

- Java API Documentation
  - http://java.sun.com/j2se/1.5.0/docs/api
- Java Swing Tutorial
  - http://java.sun.com/docs/books/tutorial/uiswing

Outline

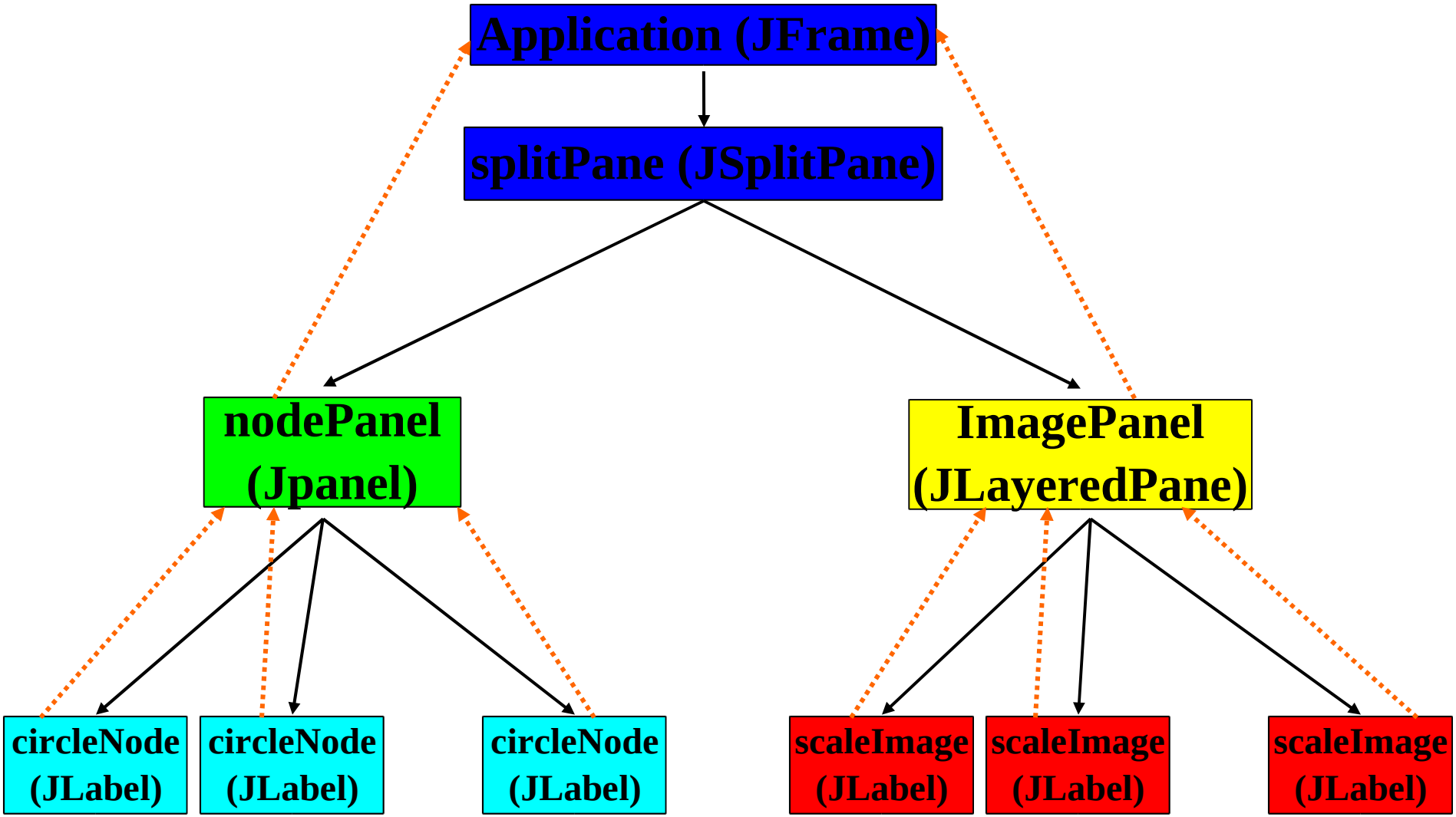
- Application Structure
- Packaging
- Java Swing
  - Containers
    - JFrame
    - JPanel
    - JSplitPane
    - JTabbedPane
    - JLayeredPane
  - Components
    - ImageIcon
    - JLabel
    - JMenu
    - JTable
    - JCheckBox
    - JTextField
    - JButton
  - Events
    - MouseEvent
    - MouseMotionEvent
    - ActionEvent
    - ItemEvent
    - ListSelectionEvent

App Breakdown



JSplitPane

App Structure



Packages

- Package edu.nyu.attendance
  - Class source located in path edu/nyu/attendance
  - Access using edu.nyu.attendance.Attendance

Jar Files

- MANIFEST.MF
  - Main-Class: edu/nyu/attendance/Attendance
  - Class-Path: jars/attendance.jar
- Creating the Jar File
  - jar cvmf <manifest> <jar file> <dirs...>
  - jar cvmf MANIFEST.MF attendnace.jar edu imgs
- Running jar
  - java -jar attendance.jar

JFrame

- Window with decorations
  - Border
  - Title
  - Buttons for closing, iconifying, etc.

```
JFrame frame = new JFrame(“A Window”);
frame.getContentPane().add(....);
frame.pack();
frame.setVisible(true);
```

JPanel

- Generic Lightweight Container

```
JPanel p = new JPanel();
p.add(....)
```

JLayeredPanel

- Container that adds depth to components
  - Integer value specifies Z-Order (highest valued components sit “on-top”)

```
JLayeredPane layeredPane = new JLayeredPane();
layeredPane.add(...,new Integer(depth));
```

```
layeredPane.getComponent(i).moveToFront();
```

Demo 1

- Empty Application
  - JFrame with Two Panels
    - JLayeredPane
    - JPane

ImageIcon

- Implements an Icon from an Image
  - Image can be from:
    - File
    - URL
  - Makes images easy to manipulate

```
ImageIcon icon = new ImageIcon("/home/brian/img.jpg");
int w=icon.getWidth();
int h = icon.getHeight();

ImageIcon bigIcon = icon.getImage().
   getScaledInstance(w*4,h*4,SCALE_SMOOTH);
```

Accessing resources in Jars

- Images (or other resources) can be located inside jar files
- Access using getClass().getResource(...)

If img is located in /imgs/ directory in jar file:

```
ImageIcon icon = new ImageIcon("/imgs/img.jpg");
```

Demo 2

- Add Background Image

JSplitPane

- Used to divide two components
  - Allows interactive resize by user

```
JSplitPane splitPane = new
   JSplitPane(JSplitPane.HORIZONTAL_SPLIT, pane1, pane2);

splitPane.setOneTouchExpandable(true);
splitPane.setDividerLocation(150);
```

JScrollPane

- Provides a scrollable view of any lightweight component

```
JScrollPane jsp = new JScrollPane(layeredPanel);
```

JTabbedPane

- Component that lets the user switch between a group of components by clicking on tabs

```
JTabbedPane tabPane = new JTabbedPane();

tabPane.add(“Panel 1”,panel1);
tabPane.add(“Panel 2”,panel2);
tabPane.add(“Panel 3”,panel3);
```

Demo 3

- Complete Shell of Application
  - JSplitPane
  - JScrollPane
  - JTabPane

JLabel

- Display area for text, image, or both

```
JLabel label = new JLabel("Image and text",imageIcon,
                                        JLabel.CENTER);

label.setVerticalTextPosition(JLabel.BOTTOM);
label.setHorizontalTextPosition(JLabel.CENTER);
```

JMenu

- Popup window that contains menu items
  - Menu is essentially a button with a popup menu

```
JPopupMenu popup = new JPopupMenu();

JMenuItem menuItem1 = new JMenuItem(“Item 1”);
popup.add(menuItem1);

popup.addSeparator();

JMenuItem menuItem2 = new JMenuItem(“Item 2”);
popup.add(menuItem2);
```

ActionEvent

- Semantic event that indicates a component level event has occurred

```
menuItem1.addActionListener(this);
menuItem1.setActionCommand(“item1”);
menuItem2.addActionListener(this);
menuItem2.setActionCommand(“item2”);
```

Demo 4

- Add Objects and Menus
  - JMenu in both panes
  - Create Initial Objects (based on JLabel)
    - ScaleImage
    - JCircle

ActionListener

```
public class Test implements ActionListener
.....................

public void actionPerformed(ActionEvent e) {
  String cmd = e.getActionCommand();
  if (“item1”.equals(cmd)) {
    // do something for item 1
  } else if (“item2”.equals(cmd)) {
    // do something for item 2
  }
```

MouseEvents

- An event which indicates that a mouse action has occurred
- Mouse Events
  - A mouse button is pressed, released or clicked
  - A mouse button enters/exits a component's geometry
- Mouse Motion Events
  - Mouse move
  - Mouse drag

MouseListener

```
public class Test implements MouseListener
.....................

public void mousePressed(MouseEvent e) {
}
public void mouseReleased(MouseEvent e) {
}
public void mouseClicked(MouseEvent e) {
}
public void mouseEntered(MouseEvent e) {
}
public void mouseExited(MouseEvent e) {
}
```

MouseMotionListener

```
public class Test implements MouseMotionListener
.....

public void mouseMoved(MouseEvent e) {
}
public void mouseDragged(MouseEvent e) {
}
```

Demo 5

- Add Mouse Events for ScaleImage
  - Movement
  - Resizing

Demo 6

- Node Control Data
  - Add control data for nodes in NodePanel
  - Add MouseEvents for JCircle movement

JTable

- Display and edit two dimensional tables

```
String colNames = {“Name”,”ID #”,”Present”};

Object[][] data = {{“Brian”,new Integer(23),new Boolean(true)},
        {“Gary”,new Integer(11),new Boolean(true)},
        {“Phillip”,new Integer(35),new Boolean(false)}};

JTable table = new JTable(data,colNames);
```

ListSelectionListener

```
public class Test implements ListSelectionListener
.....

public void valueChanged(ListSelectionEvent e) {
}
```

JCheckBox

- An item that can selected or deselected

```
JCheckBox sketchCB = new JCheckBox("Initial Sketch");
JCheckBox protoCB = new JCheckBox("Prototype");
JCheckBox finalCB = new JCheckBox("Final Project");

sketchCB.setSelected(true);
sketchCB.addItemListener(this);
protoCB.addItemListener(this);
finalCB.addItemListener(this);
```

ItemListener

```
public class Test implements ItemListener
.....

public void itemStateChanged(ItemEvent e) {
  Object source = e.getItemSelectable();
  if (source == sketchCB) {
  } else if (source == protoCB) {
  } else if (source == finalCB) {
  }
}
```

Demo 7

- Linked Views
  - Add JTable list
  - Allow selection in either view