

CSSE 220 Day 16

Event Based Programming

Questions?

CSSE Faculty Candidate Talk

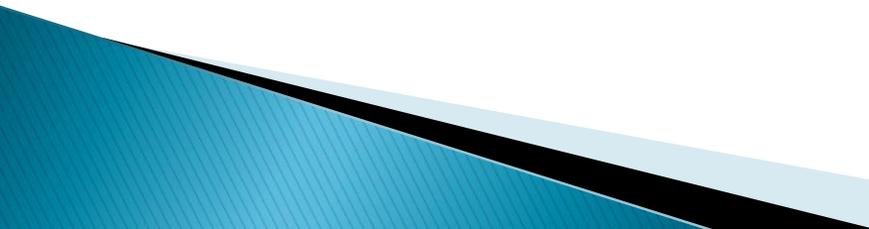
THE CHALLENGE OF QoS IN CONFIGURABLE MESH NETWORKS

Nadine Shillingford

**Department of Computer Science and
Engineering**

University of Notre Dame

Thursday, 4:20 PM, Olin 169



Today

- ▶ SwingDemo2: Implementing a Graphical User Interface (GUI)
 - Adding components to containers
 - Layout Managers
 - Event-driven programming
 - Buttons, Mouse
 - Drawing on a component (review)
 - Applets
- ▶ Detailed instructions, lots of interaction with partner and me
- ▶ Brief words about halfway through the class
- ▶ Due Thursday

Key Layout Ideas

- ▶ Containers like **JFrame** and **JPanel** have an **add(Component c)** method
 - Adds a new component to be drawn
 - **JFrame** for the top-level container, **JPanel** to organize subcomponents
- ▶ You control how the components are placed on the window, and how they change when the window is resized, with a **LayoutManager**
 - You will experience **FlowLayout** and **BorderLayout** today

SwingDemo2 Teams – Boutell

n	Team
01	duganje,popenhjc
02	kominet,davidsac
03	krachtkq,bugshank
04	lemmersj,beaversr
05	carvers
06	weavergg,wanstrnj
07	walthagd,amanb
08	cheungkt,woodhaal
09	pedzindm,foltztm
10	shinnsn,parasby

n	Team
11	hugheyjm,hannumed
12	labarpr,eatonmi
13	smebaksg,mcgeevsa
14	correlbn,sheetsjr
15	breenjw,macshake
16	moravemj,ngop
17	runchemr

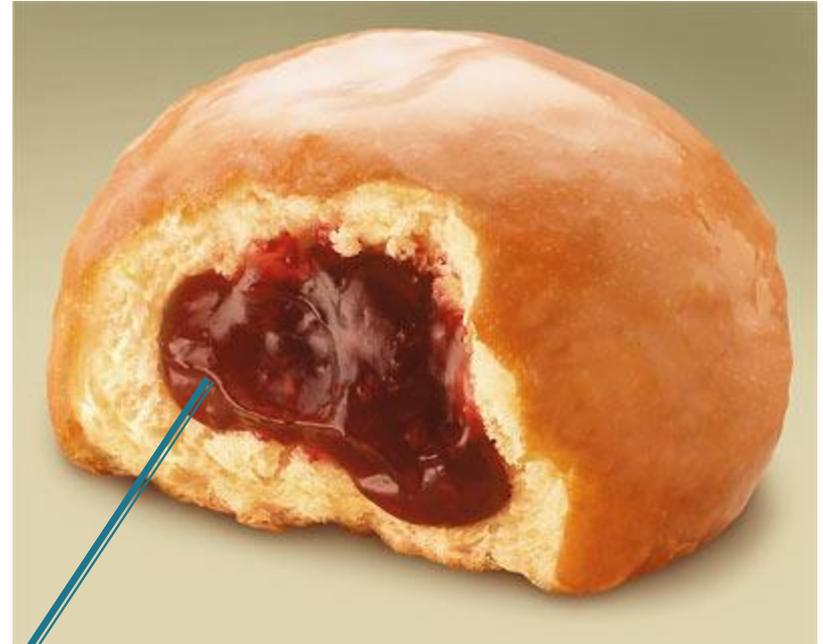
Check out *SwingDemo2* from SVN

Team number used in repository name:

<http://svn.csse.rose-hulman.edu/repos/csse220-201030-swingdemo2-teamXX>

Graphical User Interfaces in Java

- ▶ We say what to draw
- ▶ Java windowing library:
 - Draws it
 - Gets user input
 - **Calls back** to us with events
- ▶ We **handle** events



Hmm, donuts

Goody

Handling Events

- ▶ Many kinds of events:
 - Mouse pressed, mouse released, mouse moved, mouse clicked, button clicked, key pressed, menu item selected, ...
 - ▶ We tell which event source we will listen to and **add** our listener
 - Sources: buttons, menu items, graphics area, ...
 - ▶ We create **event listener objects**
 - that implement the right **interface**
 - that handle the event as we wish
- 

JButton example

- ▶ Three key steps:
 1. The JButton says which object(s) will respond when the JButton is pressed.
 2. The responding object(s) *implements ActionListener*.
 3. This means that there is an *actionPerformed* method that specifies what is to happen when the JButton is pressed

JButton example

2. Responder (this JButton) declares that it implements *ActionListener*

```
public class ExampleButton extends JButton  
                                implements ActionListener {  
    private ButtonAndMouseFrame frame;
```

```
    public ExampleButton(ButtonAndMouseFrame frame) {  
        this.frame = frame;  
        this.setText("Grow");  
        this.addActionListener(this);  
    }
```

1. JButton says that it will respond to its own button presses

Who responds to them?

Who is generating events?

@Override

```
    public void actionPerformed(ActionEvent buttonEvent) {  
        this.frame.grow();  
    }
```

3. Responder (this JButton) implements the required *actionPerformed* method, that says what to do when the JButton is pressed

A JButton often refers to one or more other objects (here, the ButtonAndMouseFrame) that it receives in its constructor and stores in a field. Or we could write a separate void setFrame(frame) method instead! (See buttonAndMouseExample in SwingDemo2 for the complete example.)

Another example: Button in a Panel

- ▶ Button is the event source
- ▶ Panel has to respond to the event and therefore can easily listen for events.

```
public TopPanel extends JPanel implements ActionListener {
    private JButton changeColor;
    ...
    public TopPanel(){
        this.changeColor = new JButton("Click to change color");
        this.changeColor.addActionListener(this);
        this.add(changeColor);
    }

    public void actionPerformed(ActionEvent e){
        //Change the background color of the panel
    }
}
```

Next time: inner classes

- ▶ Can save some work
- ▶ You are free to try them based on your past reading, but I'll demo tomorrow