GUI Programming: Events

Slides derived from the work of
Dr. Amy McGovern and Dr. Deborah Trytten
How do we make our GUIs DO something?

• What does it mean to click a button?
• How do you know when the user clicks somewhere?
• How do you grab text that a user types?
• What does clicking a button, a radio check box, entering text, and moving your mouse all have in common?
  • EVENTS!
What does it mean to click a button?
What does it mean to click a button?

User clicks button → Button generates an Event → Event listener handles the event
ActionListener Interface

• Your event listener must provide an implementation that handles the event
• This event listener is registered using `component.addActionListener(listener)`

-> Java API: Examine ActionListener
Examples

• First: Make our two button interface print out different text when we click on the two buttons
• Second: Make the buttons change the color inside the window somehow
Example

Hello my name is ___________
OK

Have it print out “Hello NAME” where name is what you enter
More event driven programming
Inner Classes

- Inner class is defined inside another class
- Can access variables in outer class
- Can be very useful in handling events
Anonymous Inner Classes

```java
button.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        doSomething();
    }
});
```

- Implicitly defines a new class that extends ActionListener
- We do not give it a class name
- Created inline
Example

• Convert button example to use inner classes
• Convert our button printout example to use anonymous inner classes
Other Things Listeners Can Do

Ask for the source:
• ActionEvent.getSource
  • See API
Example

Rewrite button example to only have one listener for all buttons
  • Use an inner class
Types of Events

• JEvent:
  • JButton, JTextField, JComboBox, JRadioButton, JCheckBox, ...

• Mouse events:
  • mouse button being pressed
  • mouse moving

• Keyboard presses
Mouse Events

• Examine MouseEvent API
  • Look at InputEvent (parent of MouseEvent) also

• Often use Listener Interface Adapters
  • Saves you from having to implement every event
  • Default implementation is empty
  • -> Examine MouseAdaptor API
Example

• Adapt the 😊 example to follow the mouse
  • Dragged or movement?

• Advanced example:
  • Give 😊 eyes that track the mouse
  • Won’t do this in class but it is good work for you to ensure you understand graphics and events!
Keyboard Events

• Examine KeyListener API
• Can use to make games!
• Example in lab
Animation: another option

• Examine the Timer class API
  • Javax.swing.Timer

• Can be used to implement a regular timer
Example

• Adapt the bouncing smiley demo to use a timer