§18.2: Graphics

13 Feb 2008
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

JFrame.dispose()

- Window listeners: do actions on close, iconify, etc.
- myJFrame.setDefaultCloseOperation()
  - Sets what happens when close button is clicked
    - In addition to any window listeners
  - Default: JFrame.HIDE_ON_CLOSE
    - Doesn't actually close the window!
  - JFrame.DISPOSE_ON_CLOSE makes more sense
- myJFrame.dispose()
  - Closes window and deallocates memory
  - Does not end program

Scroll bars

- Swing components can be contained inside scroll panes: show only a viewport of the whole widget
- e.g., a text area:
  - JTextArea blogEntry = new JTextArea(10, 40)
    - Only shows 10 lines, 40 characters of text
  - JScrollPane scrolledBlog = new JScrollPane(blogEntry);
    - Wrap in a scroll pane
  - myJPanel.add(scrolledBlog);
    - Add to a panel or window
- Scroll bar policy: whether to show a scrollbar or not
  - .setVerticalScrollBarPolicy(JScrollPane.
    VERTICAL_SCROLLBAR_AS_NEEDED);

Swing graphics: .paint()

- JFrames have a .paint() method, which draws the window on the screen
  - To do our own drawing, override paint()
  - Make sure to call super.paint() first to get the normal behaviour, then do our own drawing on top
- paint() takes a Graphics context as its argument
  - All drawing routines are methods of the Graphics

```
public class SmileyFace extends JFrame {
    public void paint( Graphics g ) {
        super.paint( g );
        g.drawOval( .... );
```

paint() vs. paintComponent()

- JFrames: use paint() method
- JPanels and other JComponents: use paintComponent()

- paint() and paintComponent() are only called when the window or component needs to be redrawn
  - e.g., expose after being covered by something
- If you make a change and want to request a redraw,
  - repaint() (method of JFrame or JComponents)

Lines and rectangles

- import java.awt.Graphics;

- g.drawLine(int x1, int y1, int x2, int y2);
  - Coordinates in pixels from top-left of component
- drawRect(x, y, w, h), fillRect
  - (x,y) is top-left corner of rectangle
- draw3DRect(x, y, w, h, boolean raised)
  - Border-shading so it looks raised or sunken
- drawRoundRect(x, y, w, h, arcW, arcH)
  - Specify diameter of rounded corners

Ovals and arcs

- g.drawOval(x, y, w, h), fillOval
  - Circles are ovals with equal width and height
- drawArc(x, y, w, h, angle, sweep), fillArc
  - Specify starting angle (0 points to right)
  - Specify how far the arc should go
  - Angle and sweep are both in integer degrees

Colours (colors)

- import java.awt.Color;
- Set the current drawing colour before drawing the object:
  - g.setColor(Color.BLUE);
  - g.drawArc(50, 50, 100, 100, 200, 140);
  - g.setColor(Color(0.7, 0.9, 0.1));
  - g.drawOval(80, 80, 40, 40);
- A few named colours, or use an RGB triple
- JColorChooser: dialog to select and return a Color
  - JColorChooser.showDialog(this, “title”, defaultColor);