Introduction to Graphics

Chapter 3G
Graphics and Graphical Interfaces

Java graphics API

- **Java 1.0 AWT (abstract windows toolkit)**
  - AWT graphics are based on native peers
  - Sun trademarked the slogan “write once, run everywhere”
  - Platform look and feel
  - Lowest common denominator capability (equally mediocre everywhere)
  - Subtle differences between platforms lead to inconsistent results
  - Different bugs in the AWT on different platforms lead to the slogan “write once, debug everywhere”

- **Java 1.1 AWT**
  - Still peer based (i.e., still lowest common denominator)
  - More consistent naming and features
  - Completely redesigned event model
  - More object-oriented (modular) design
Graphics and Graphical Interfaces

Continued

- **Java 1.2**: AWT (peer based) and Swing (non-peer based)
- **In 1996 Netscape creates the IFC (Internet Foundation Classes)**
  - GUI components are painted on a blank screen
  - The only peer functionality used is creating and painting a window
  - GUI components look and feel the same on all platforms
  - Sun & Netscape perfected the project code named Swing
- **Swing is the official name for the non-peer-based GUI portion of the JFC (Java Foundation Classes)**
  - Swing uses the Java 1.1 event delegation model
  - Platform independent
    - Less sensitive to platform-specific bugs
    - Provides components not available on all platforms
    - Consistent look and feel across all platforms
Graphics Overview

The basics

- Graphics and Graphical User Interfaces (GUIs)
  - Use A LOT of API classes (java.awt, java.awt.event, and javax.swing)
  - Use the full spectrum of the object-oriented paradigm
    - Encapsulation (chap 8)
    - Inheritance (chap 9)
    - Polymorphism (chap 9)
  - Is based on an event-driven interaction model
    - Event handlers are based on interfaces (chap 9)
    - One object is the source of an event
    - Another object listens for and responds to events
Major Classes

More basics

- Graphics (java.awt)
  - Provides drawing methods

- JFrame (javafx.swing)
  - Title bar with buttons and all borders
  - The basic window in which everything takes place

- JPanel
  - A canvas on which things are drawn
  - A organizing container to arrange components
Frame Format

Java frame coordinate system

- Measured in pixels
- Everything in a Frame is drawn on a Graphics context
Swing and the “X” Button

Introduced at Java 1.3

- JFrame method: setDefaultCloseOperation(int closeAction)
  - WindowConstants.DO NOTHING_ON_CLOSE
  - WindowConstants.HIDE ON_CLOSE
    - the default action for a JFrame
    - hide the frame after invoking any registered WindowListener objects
  - WindowConstants.DISPOSE ON_CLOSE
  - JFrame.EXIT ON_CLOSE
    - close with System.exit()
    - use only in applications (i.e., not in applets)
Lines and Shapes

Graphics methods (call from paint or paintComponent: g.drawLine ...)

- void drawLine(int x0, int y0, int x1, int y1)
- void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
- void drawPolygon(Polygon p)
- void drawPolygon(int xPoints[ ], int yPoints[ ], int numPoints)
- void drawRect(int x, int y, int width, int height)
- void drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)
- void draw3DRect(int x, int y, int width, int height, boolean raised)
- void drawOval(int x, int y, int width, int height)
More About Shapes

The meaning of parameters

\( (x, y) \)

\[ \text{drawRect}(x, y, \text{width}, \text{height}) \]

\[ \text{drawOval}(x, y, \text{width}, \text{height}) \]

\[ \text{drawArc}(x, y, \text{width}, \text{height}, \text{startAngle}, \text{arcAngle}) \]

\[ \text{drawRoundRect}(x, y, \text{width}, \text{height}, \text{arcWidth}, \text{arcHeight}) \]
Color

Foreground, background, lines, and fill (see colorDemo.java)

- **Color class**
  - Color(int red, int green, int blue)  // all values are [0-255]
  - Color(int RGB)  // RGB: 0xrrggbb
    - RGB is in hexadecimal: blue bits 0-7; green bits 8-15; red bits 16-23
  - Predefined colors
    - Color.black (0,0,0)  Color.green (0,255,0)  Color.red (255,0,0)
    - Color.blue (0,0,255)  Color.lightGray
    - Color.cyan  Color.magenta  Color.white (255, 255,255)
    - Color.darkGray  Color.orange
    - Color.gray  Color.pink

- **Graphics and Frame methods**
  - void g.setColor(Color c)  // call in paint or paintComponent
  - void setForeground(Color c)  // call in any method
  - void setBackground(Color c)  // call in any method
  - Effects all subsequent drawing operations