Java (pre-Java2D) Graphics

Painting: Down the Containment Hierarchy

- JFrame
  - ...
  - content pane
  - JPanel
    - JButton
    - JLabel
Custom Component

- The paint method delegates work to:
  - paintComponent
  - paintBorder
  - paintChildren
- Override paintComponent for specialized UI/look-and-feel
- Invoke repaint to schedule for painting

Simple Example

class ImagePanel extends JPanel {
  
  public void paintComponent(Graphics g) {
    // paint background
    super.paintComponent(g);
    // 85x62 image
    g.drawImage(image, 0, 0, this);
    // same image, scaled and moved right
    g.drawImage(image, 90, 0, 300, 62, this);
  }
}
Non-opaque Component

- In `paintComponent`:
  don't invoke `super.paintComponent`
- In constructor:
  `setOpaque(false);`

Graphics Object

- Context for
  - color, font, clipping, etc.
- Drawing methods
  - simple shapes
  - text
  - image
  - copying a region
Coordinate System

With 1-pixel Border
Simple Shapes

- Line
  \[ g.\text{drawLine}(x_1, y_1, x_2, y_2); \]
- Rectangle
  \[ g.\text{drawRect}(x, y, w - 1, h - 1); \]
  \[ g.\text{fillRect}(x, y, w, h); \]

Simple Shapes (cont’d)
Working with Text

- Loading a font
  ```java
  g.setFont(new Font("SansSerif", Font.PLAIN, 12));
  ```
- Drawing
  ```java
  g.drawString("Hj", x, y);
  ```

Font Metrics

```
getDescent

Hj

getAscent

getLeading

getHeight

Next line of text

Ascender Line

Baseline

Descender Line
```


Image

- Loading
  ```java
  myImage = getImage(URL);
  myImage = Toolkit.getDefaultToolkit().getImage(filenameOrURL);
  ```

- Drawing
  ```java
  g.drawImage(myImage, 0, 0, this);
  ```

Tracking image loading

```java
MediaTracker tracker = new MediaTracker(this);
tracker.addImage(myImage);

try {
    tracker.waitForAll(); // wait till image loading finishes
} catch (InterruptedException e) {};

if (tracker.checkAll()) { // loading completed
    // draw the image
} else { // loading interrupted
    // do something else
}
```
### ImageIcon

- Sequential image loading
- Loading
  ```java
  ImageIcon icon = new ImageIcon(filenameOrURL);
  ```
- Drawing
  ```java
  icon.paintIcon(this, g, x, y);
  ```

### Clipping

```java
Rectangle oldClipBounds = g.getClipBounds();

  g.setClip(new Rectangle(...));

  // Perform custom painting...

  g.setClip(oldClipBounds);
```

```
Color

- Pre-defined colors
  - Color.black, Color.cyan, . . .
- Custom colors
  - new Color(r, g, b)
- Setting color
  - g.setColor(myColor);

Forthcoming: Java2D!

- Complex shapes and attributes
- Fill gradients and textures
- Transformations
- Image processing
- Hit detection
- Composition of overlapping drawings
- Printing