

Java

Chapter 19 - Lecture Slides

(c) 2007 by E. S. Boese. All Rights Reserved.

Why Java?

(c) 2007 by E. S. Boese. All Rights Reserved.

Why Java?

ADVANTAGES	DISADVANTAGES
	Limited to abilities that all machines can produce since it has to run on all types of machines (e.g., lose the extra graphics abilities of SGIs)

Understanding O-O

- Object-oriented is defined as:
data and methods associated with the data are stored together as an 'object'
- Difference between OO languages and procedural languages:
- What are some *objects* you might create if you were asked to create:
 - Contact address application
 - Grade rolls for a course
 - Transcript details for students at a school
 - Search engine
- How are methods useful in objects?
- Once we create an object, how many times can we *instantiate* it?
For example, could we create 4 Smiley objects from the examples used previously?
How?
How much code did we need to make this happen?
Why was it easy?
How does this relate to objects?

(c) 2007 by E. S. Boese. All Rights Reserved.

Security

-
-
-
-
- Windows that are displayed from an applet are displayed differently than windows from an application, so that the user knows where it came from
- What is the danger in allowing applets to read files?
- What is the danger in allowing applets to write files?

(c) 2007 by E. S. Boese. All Rights Reserved.

Modifying and Using Free Applets

(c) 2007 by E. S. Boese. All Rights Reserved.

Modifying and Using Free Applets

- Free applets available
 - Some provide source code and allow you to modify it as you like.
 - Be sure to read the copyright, licensing and agreements before you use.
- <http://java.internet.com>
 1. Find an applet you want.
MAKE SURE it includes the java SOURCE code, NOT just the .class files!
 2. Download the files (if it's a .zip file, you'll need to extract the contents of the file)
 3. Copy the .java file(s) into your IDE (e.g. Eclipse) or editor.
 4. Compile and run the Java Applet.

(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D

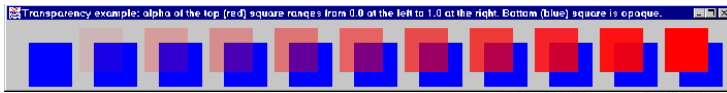
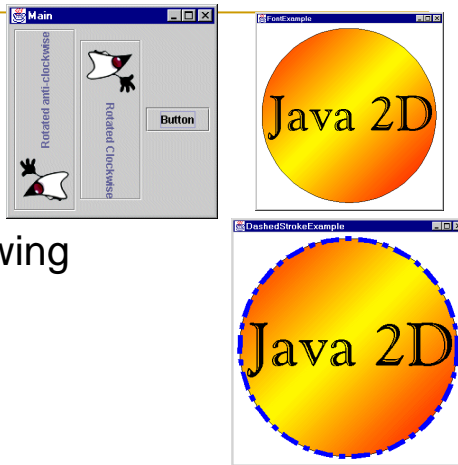
(c) 2007 by E. S. Boese. All Rights Reserved.

2-D Graphics

Graphics2D

- More control over drawing

-
-
-
-



Graphics2D

- Graphics2D instead of Graphics

-
-
-
-
- Composite
- Clip
- Rendering hints

(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D

- Cast Graphics to Graphics2D wherever you have a Graphics object

```
public void paint( Graphics g )
{
}
public void paintComponent( Graphics g )
{
    super.paintComponent( g );
}
```

- Use **setPaint** to specify
 - solid color (Color)
 - a gradient fill (GradientPaint),
 - tiled image (TexturePaint).

(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D Example

```
import java.awt.*;
import java.awt.geom.*;
import javax.swing.*;
public class Graphics2DEx extends JApplet
{
    public void paint (Graphics g)
    {
        Graphics2D g2 = ( Graphics2D ) g;
        int x = 15, y = 10, w = 50, h = 50;
        Ellipse2D ellipse = new Ellipse2D.Double(x, y, w, h);
        GradientPaint gp = new GradientPaint(50, 75, Color.white, 95, 95, Color.blue, true);
        g2.setPaint(gp); // fill with gradient
        g2.fill(ellipse);
    }
}
```



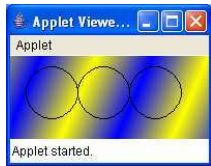
(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D

■ GradientPaint

```
GradientPaint gp = new GradientPaint(  
    x1, y1, color1,  
    x2, y2, color2, true );  
  
// gradient from one color to the other
```

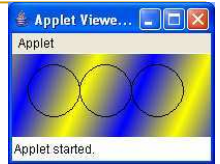
□



(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D - GradientPaint

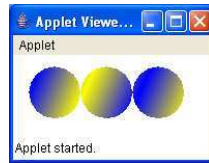
```
import java.awt.*;  
import java.awt.geom.*;  
import javax.swing.*;  
public class Graphics2DEx extends JApplet  
{  
    public void paint(Graphics g)  
    {  
        Graphics2D g2 = ( Graphics2D ) g;  
        int x = 15, y = 10, w = 50, h = 50;  
        Ellipse2D ellipse = new Ellipse2D.Double(x, y, w, h);  
        Ellipse2D ellipse2 = new Ellipse2D.Double(x+w, y, w, h );  
        GradientPaint gp = new GradientPaint(50, 75, Color.yellow, 95, 95, Color.blue, true);  
        g2.setPaint(gp); // fill with gradient  
        g2.fill( new Rectangle( 0,0,200,200 ) );  
        g2.setPaint( Color.BLACK ); // draw outlines of ovals  
        g2.draw(ellipse);  
        g2.draw(ellipse2);  
        g2.draw(new Ellipse2D.Double( x+w+w, y, w, h ) );  
    }  
}
```



(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D - GradientPaint

```
import java.awt.*;  
import java.awt.geom.*;  
import javax.swing.*;  
public class Graphics2DEx extends JApplet  
{  
    public void paint(Graphics g)  
    {  
        Graphics2D g2 = ( Graphics2D ) g;  
        int x = 15, y = 10, w = 50, h = 50;  
        Ellipse2D ellipse = new Ellipse2D.Double(x, y, w, h);  
        Ellipse2D ellipse2 = new Ellipse2D.Double(x+w, y, w, h );  
        GradientPaint gp = new GradientPaint(50, 75, Color.yellow, 95, 95, Color.blue, true);  
        g2.setPaint(gp); // fill with gradient  
        g2.fill(ellipse);  
        g2.fill(ellipse2);  
        g2.fill(new Ellipse2D.Double( x+w+w, y, w, h ) );  
    }  
}
```



(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D

Same drawing methods as before

```
clearRect( x, y, w, h )  
draw3DRect( x, y, w, h, true )  
drawArc( x, y, w, h, startAngle, arcAngle )  
drawImage( x, y, w, h, this )  
drawLine( x1, y1, x2, y2 )  
drawOval( x, y, w, h )  
drawPolygon( Polygon )  
drawRect( x, y, w, h )  
drawRoundRect( x, y, w, h, arcWidth, arcHeight )  
drawString( String, x, y )  
  
fill3DRect( x, y, w, h, true )  
fillArc( x, y, w, h, startAngle, arcAngle )  
fillOval( x, y, w, h )  
fillPolygon( Polygon )  
fillRect( x, y, w, h )  
fillRoundRect( x, y, w, h, arcWidth, arcHeight )  
  
setFont( Font )  
setColor( Color )
```

(c) 2007 by E. S. Boese. All Rights Reserved.

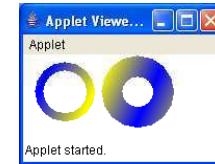
Graphics2D

- Use `g2d.setPaint()` for transparency.
See:
<http://www.cs.colostate.edu/~boese/JavaApplets/Code/Java2D/Composite.html>
- Improve drawing quality or enable **antialiasing** through use of rendering hints (see chapter 4)

(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D - Stroke

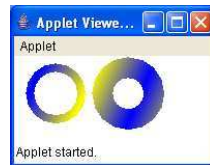
- **BasicStroke** (use `g2d.setStroke()`) lets you set
 -
 -



(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D - Stroke

```
import java.awt.*;
import java.awt.geom.*;
import javax.swing.*;
public class Graphics2DStroke extends JApplet
{
    public void paint(Graphics g)
    {
        Graphics2D g2 = ( Graphics2D ) g;
        int x = 15, y = 10, w = 50, h = 50;
        Ellipse2D ellipse = new Ellipse2D.Double(x, y, w, h);
        Ellipse2D ellipse2 = new Ellipse2D.Double(x+w+20, y, w, h );
        GradientPaint gp = new GradientPaint(50, 75, Color.yellow, 95, 95, Color.blue, true);
        g2.setPaint ( gp ); // fill with gradient
        g2.setStroke ( new BasicStroke( 8 ) ); // outline is 8 pixels
        g2.draw ( ellipse);
        // Stroke with a gradient.
        g2.setStroke ( new BasicStroke( 20 ) ); // outline is 20 pixels
        g2.draw ( ellipse2 );
    }
}
```



(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics2D - Stroke

```
import java.awt.*;
import java.awt.geom.*;
import javax.swing.*;
public class Graphics2DStroke extends JApplet
{
    public void paint(Graphics g)
    {
        Graphics2D g2 = ( Graphics2D ) g;
        int x = 15, y = 10, w = 50, h = 50;
        Ellipse2D ellipse = new Ellipse2D.Double(x, y, w, h);
        GradientPaint gp = new GradientPaint(50, 75, Color.yellow, 95, 95, Color.blue, true);
        g2.setPaint ( gp ); // fill with gradient
        // Stroke
        float strokeThickness = 5.0f;
        BasicStroke stroke = new BasicStroke(strokeThickness);
        Shape newShape = stroke.createStrokedShape(ellipse);
        g2.draw( newShape );
    }
}
```



(c) 2007 by E. S. Boese. All Rights Reserved.

Other Features through 3rd Party Libraries

(c) 2007 by E. S. Boese. All Rights Reserved.

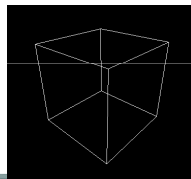
Other Features through 3rd-party libraries

- Some are free, some cost money
- Catch - users that view your website with your applet can usually only do so if they too have downloaded and installed the libraries or a run-time interpreter for these additional functionalities.
- Some of the available features include:
 -
 -
 -
 -
 -
 -
 -

(c) 2007 by E. S. Boese. All Rights Reserved.

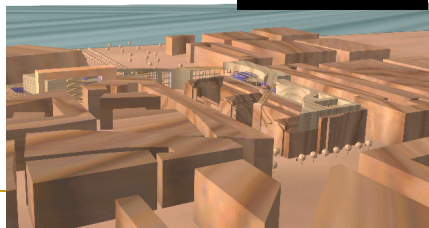
Graphics3D

- 3-D library
- JMF:
(video too)



<http://www.frontiernet.net/~imaging/java3dviewer.html>

http://www.frontiernet.net/~imaging/play_a_piano.html



(c) 2007 by E. S. Boese. All Rights Reserved.

Graphics3D

- For cell phones
 - (micro edition)
 - Mobile 3D Graphics API for J2ME (micro edition)



(c) 2007 by E. S. Boese. All Rights Reserved.

Jar files

- Compile all your files for your applet together
- Similar to .zip
- Ends with .jar
- Faster download for users
- Compresses files
- Reference in HTML page:

```
<HTML>
  <BODY>
    <APPLET CODE=XYZ.class ARCHIVE="files.jar"
      WIDTH=500 HEIGHT=500>
    </APPLET>
  </BODY></HTML>
```

(c) 2007 by E. S. Boese. All Rights Reserved.

Jar files

- Create
 - `jar cvf jar-file input-file(s)`
- View
 - `jar tvf jar-file`
- Extract
 - `jar xf jar-file`
- Example:
 - `jar cvf project.jar proj.class sound.mid img1.gif img2.gif`
 - `jar cvf project.jar *.class *.mid *.gif`

(c) 2007 by E. S. Boese. All Rights Reserved.

Summary

- Why Java
- Object-Oriented
- Security
- Modifying and using free applets
- Graphics2D
- Other features using 3rd party libraries
- Jar files

(c) 2007 by E. S. Boese. All Rights Reserved.