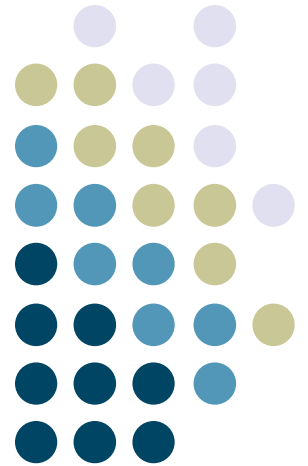


Java Swing GUI Programming I



**Georgia
Tech**

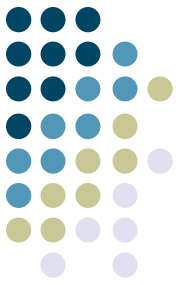


John Stasko

CS 6452

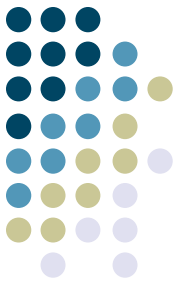
Prototyping Interactive Systems

Java Questions



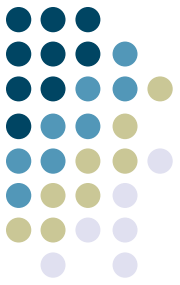
- What isn't clear or what is giving you difficulties?

Learning Objectives



- Java's toolkits for graphics and GUIs
- Fundamental drawing operations for different graphics objects
- Structure of a Swing application
 - Frame, Panel, components, painting, JLabel, ImageIcon concepts

Java GUI Programming



- For many years: AWT & Swing
- More recently: JavaFX

- We're going to do Swing
 - More straightforward
 - JavaFX uses some advanced concepts we haven't emphasized
 - Still communicates event-driven principles

Java Application

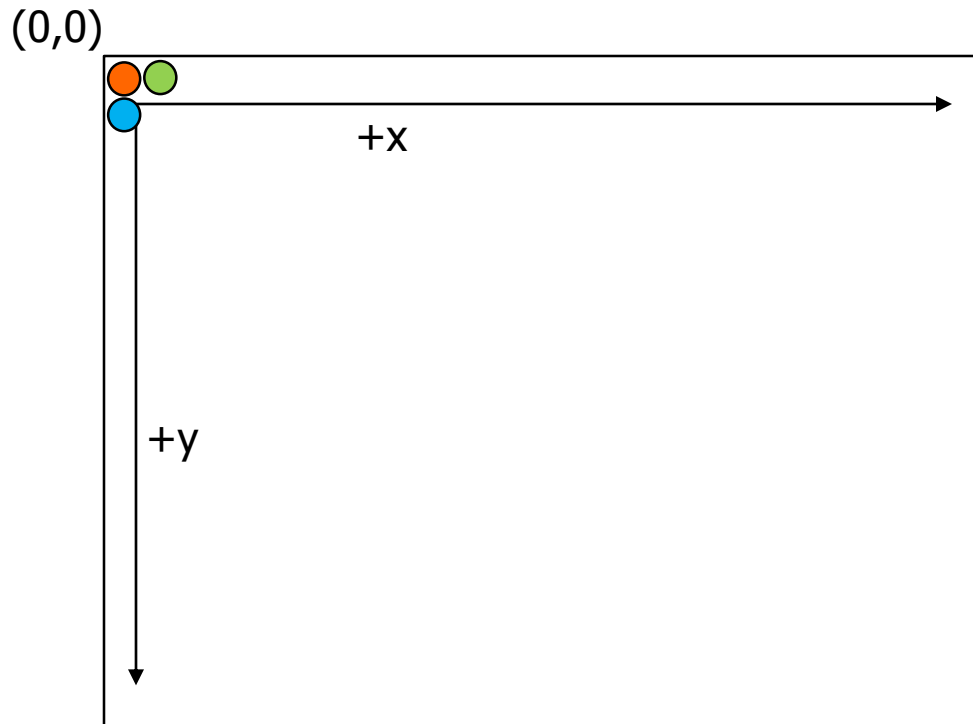


- Stand-alone graphics program with main()
- Two main components:
 - Graphics operations
 - Program structure (containers)

Graphics



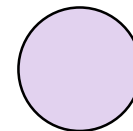
Pixel – picture element



1280 x 1024

Colors
red, green, blue

0 -> 255



226, 210, 239

Drawing Operations



Graphics class – provided by Java

```
void drawLine(int x1, int y1, int x2, int y2)
```

(x1,y1)

(x2,y2)

```
void drawRect(int x, int y, int width, int height)
```

```
void fillRect(int x, int y, int width, int height)
```

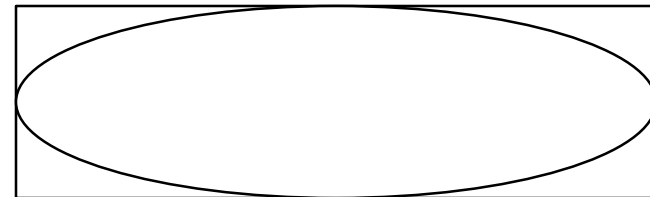
```
void drawOval(int x, int y, int width, int height)
```

```
void fillOval(int x, int y, int width, int height)
```

```
void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
```

(x,y)

width



height

```
void drawString(String str, int x, int y)
```

>Hello
(x,y)

Colors



Constructors

Constructor and Description

- `Color(ColorSpace cspace, float[] components, float alpha)`
Creates a color in the specified `ColorSpace` with the color components specified in the float array and the specified alpha.
- `Color(float r, float g, float b)`
Creates an opaque sRGB color with the specified red, green, and blue values in the range (0.0 - 1.0).
- `Color(float r, float g, float b, float a)`
Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0.0 - 1.0).
- `Color(int rgb)`
Creates an opaque sRGB color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7.
- `Color(int rgba, boolean hasAlpha)`
Creates an sRGB color with the specified combined RGBA value consisting of the alpha component in bits 24-31, the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7.
- `Color(int r, int g, int b)`
Creates an opaque sRGB color with the specified red, green, and blue values in the range (0 - 255).
- `Color(int r, int g, int b, int a)`
Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0 - 255).

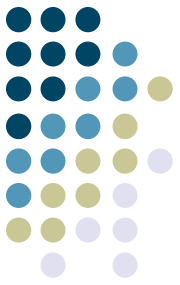
Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description		
<code>Color</code>	<code>brighter()</code>	Creates a new <code>Color</code> that is a brighter version of this <code>Color</code> .	
<code>PaintContext</code>	<code>createContext(ColorModel cm, Rectangle r, Rectangle2D r2d, AffineTransform xform, RenderingHints hints)</code>	Creates and returns a <code>PaintContext</code> used to generate a solid color field pattern.	
<code>Color</code>	<code>darker()</code>	Creates a new <code>Color</code> that is a darker version of this <code>Color</code> .	
<code>static Color</code>	<code>decode(String nm)</code>	Converts a String to an integer and returns the specified opaque <code>Color</code> .	

Also constants
Color.RED
Color.BLUE

...

Drawing Color



- Java uses concept of active foreground color
 - Anything drawn is shown in that color
- Change the active color via
 - `setColor(col);`

Example Program



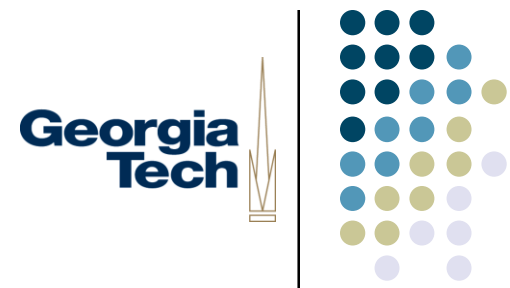
- Snowman (just focus on graphics)



How do it?

Examine Snowman program

GUI Components

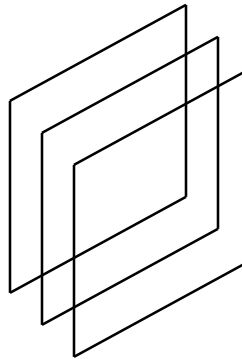
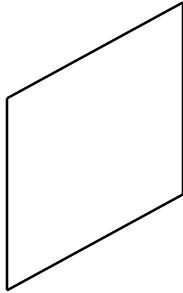


- Container – Special component for holding and organizing other components
 - Frame – Stand-alone window that is movable and resizable with title and corner buttons
`JFrame` class (heavyweight)
 - Panel – Container too, but must be added to another container
`JPanel` class (lightweight)

GUI Components



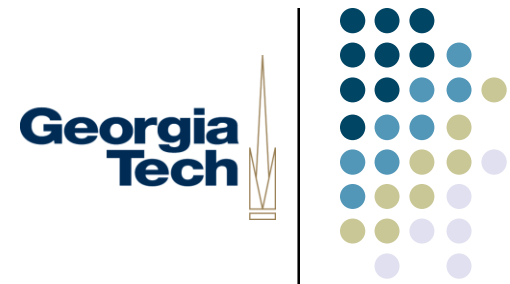
Frame



Multiple panes (not panels)

One is *content pane* that holds all visible components

Back to Snowman



```
import javax.swing.JFrame;

public class Snowman
{
    public static void main (String[] args)
    {
        JFrame frame = new JFrame ("Snowman");
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        SnowmanPanel panel = new SnowmanPanel();

        frame.getContentPane().add(panel);

        frame.pack();
        frame.setVisible(true);
    }
}
```

Your main should always look like this

Back to Snowman



```
import javax.swing.JPanel;
import java.awt.*;

public class SnowmanPanel extends JPanel
{

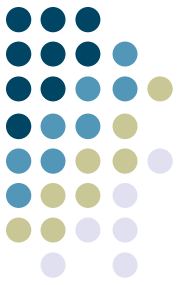
    public SnowmanPanel()
    {
        setPreferredSize (new Dimension(300, 200));
    }

    public void paintComponent (Graphics page)
    {
        super.paintComponent (page);

        // Graphics code here
    }
}
```

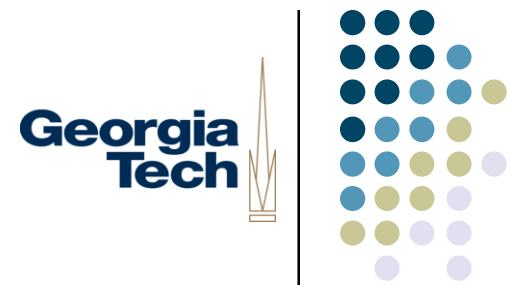
Top panel on the Frame

More Components



- Instead of doing graphics calls, we'll add component objects to be displayed
- `JLabel` – text label string

Panel Class



```
import java.awt.*;
import javax.swing.*;

public class LabelPanel extends JPanel
{
    public LabelPanel()
    {
        setPreferredSize(new Dimension(250,75));
        setBackground (Color.yellow);

        JLabel label1 = new JLabel ("Question authority,");
        JLabel label2 = new JLabel ("but raise your hand first.");

        add(label1);
        add(label2);
    }

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


Program Structure



JFrame



JPanel



JLabel

has-a relationships

Images



- Java can use jpg, gif, png image files from disk
- Graphics class has `drawImage` call or the image can be put in a `JLabel`
- Label can have text, image, or both

Images



- ImageIcon class used for images in labels

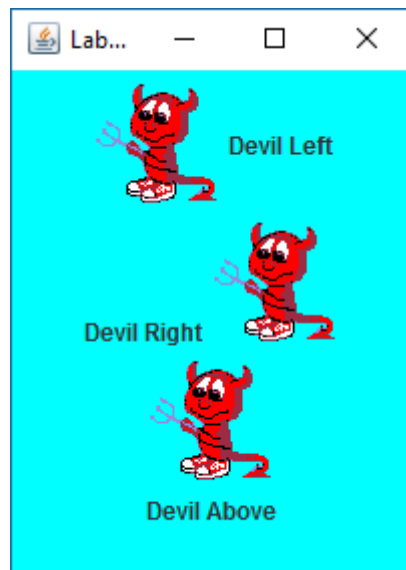
```
ImageIcon ii = new ImageIcon("face.gif");  
JLabel label = new JLabel("Text part", ii, SwingConstants.CENTER);  
label.setHorizontalPosition(SwingConstants.LEFT);
```

Where whole label goes
in panel

Orientation between image and text

Default is text right
and vertically centered

Image Demo



Examine ImageDemo program

Panel Class



```
public class ImageDemoPanel extends JPanel
{
    public ImageDemoPanel()
    {
        ImageIcon icon = new ImageIcon ("devil.gif");
        JLabel label1, label2, label3;

        label1 = new JLabel ("Devil Left", icon, SwingConstants.CENTER);

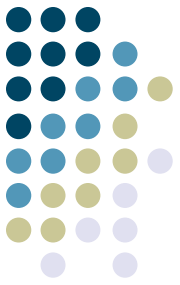
        label2 = new JLabel ("Devil Right", icon, SwingConstants.CENTER);
        label2.setHorizontalTextPosition (SwingConstants.LEFT);
        label2.setVerticalTextPosition (SwingConstants.BOTTOM);

        label3 = new JLabel ("Devil Above", icon, SwingConstants.CENTER);
        label3.setHorizontalTextPosition (SwingConstants.CENTER);
        label3.setVerticalTextPosition (SwingConstants.BOTTOM);

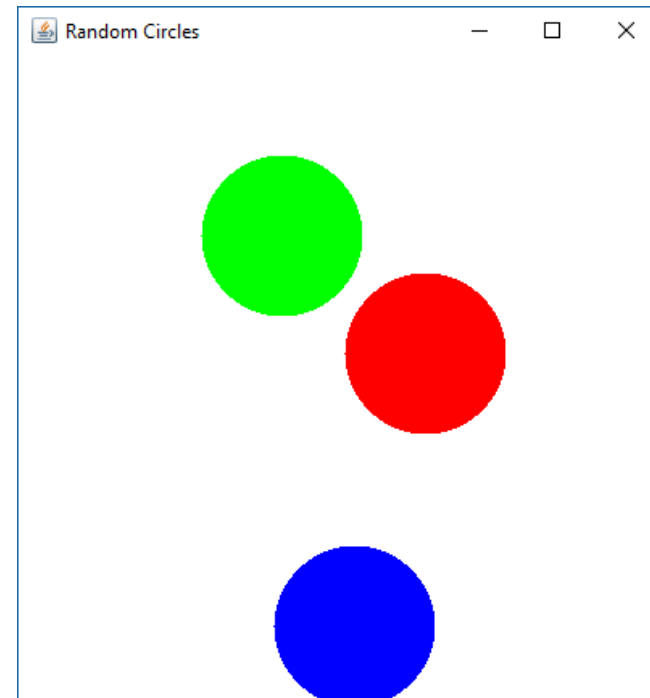
        setBackground (Color.cyan);
        setPreferredSize (new Dimension (200, 250));
        add (label1);
        add (label2);
        add (label3);
    }

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

Programming Challenge

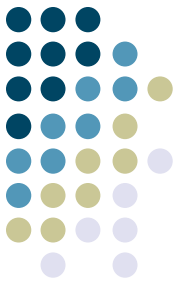


- Design a program that randomly positions 3 colored circles



Let's do it together

Questions



- What happens if you rerun it?
- What happens if you minimize it?
- How would you count the calls to the `paintComponent` method?
- Can you keep the circles in the same place all the time?

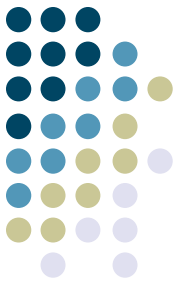
Going O-O

- Program that draws circles and remembers them
- Make Circle class



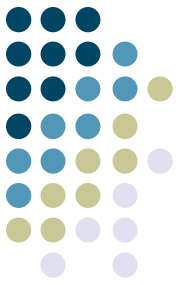
Examine Splat program

Learning Objectives



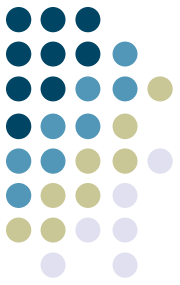
- Java's toolkits for graphics and GUIs
- Fundamental drawing operations for different graphics objects
- Structure of a Swing application
 - Frame, Panel, components, painting, JLabel, ImageIcon concepts

HW 6



- Design business card for yourself
 - Put text and graphics
 - Use `Card.java` and `CardPanel.java` from t2
- Due Thursday @ 1:30pm

Next Time



- Handling simple interaction events