

Java Graphics

Slides derived from the work of
Dr. Amy McGovern and Dr. Deborah Trytten

Graphics are fun!

- Demo...

How do I draw?

- Extend a JPanel
- Override:
 - protected void paintComponent(Graphics g)
 - Don't do this on a class already subclassing JPanel like JButton!
 - Draw your graphics inside paintComponent
 - Called autonomously using event handling

Drawing in the right spot

Remember to manage your coordinates

- Where is 0,0?
- How “tall” is the screen?
- How “wide” is the screen?
- Which way is positive x? Positive y?
- Are coordinates relative or absolute?

Drawing shapes

- Graphics class provides a LOT of useful drawing utilities
 - drawLine, drawArc, drawRect, drawString, ...
 - Can do fill versions of those too
 - Examine Graphics API
- Graphics2D provides more
 - draw(Shape s)
 - Define a general shape and draw it to the screen
 - Examine Shape API (and Graphics2D API)

Words/Strings

- `drawString` draws the string
- `Font` class: selecting fonts
- `FontMetrics` class: managing fonts
 - Can measure the extent of a `String` before it is displayed
 - Use to center text automatically

Example

Make a two panel GUI

- Top panel has text centered “Have a nice day!”
- Bottom panel shows the smiley face from previous example

Images

- `ImageIcon` is fixed size
 - Rendering: `ImageIcon.paintIcon()`
- `Image` is flexible
 - Rendering: `Graphics.drawImage`

Adding Animation

Need:

- An object to keep track of where the item you are drawing is
 - `paintComponent()` will use this information to properly place the object
- Control loop to update its coordinates
- Sleep between animation frames:
 - `Thread.sleep(ms)`
- Don't forget to call `repaint()`!

Adding Animation: Alternative

Need:

- Use a Timer to trigger the control loop (coming soon)

Example

- Make a four panel screen of bouncing smileys (one per panel)

Example

- What do you want to draw/animate?

