

# Graphics

3-21-2011

# Opening Discussion

- Do you have questions about the assignment?
- Piazza
- Tech and AI in society.
  - Face time, jobs, laziness of people
  - So much more ... dark factories, etc.
- Minute essay comments:
  - Scala applets
  - Tables in GUIs, `Array[Any]`
  - Does Java go beyond Scala?

# Motivation

- You can do lots of things with the standard GUI elements in Swing.
- We could set up quite a bit of a GUI using that. However, no GUI library can predict everything that you will want to do and we want to be able to add custom drawing to our applications.
- Uses:
  - Drawing a shot-chart in basketball.
  - Drawing Dinosaurs.
  - Drawing anything there isn't a GUI component for.

# Graphics in Scala/Java

- For this we will rely on the Java2D library. Java2D was added about the same time Swing was and it is fundamentally based on the `java.awt.Graphics2D` class.
- Let's go find the Java API and find this class in it.

# Making Custom Drawn Components

- We are going to follow a simple approach to doing this making a new Panel that draws what we want.
  - new Panel {
    - override def paint(g:Graphics2D) {
    - ...
    - }
  - }
- Whatever you draw to g appears in the panel.

# Capabilities of Java2D

- Let's look a bit at the Graphics2D class to see what some of the possibilities might be for what we can draw.
- The `java.awt.geom` package and the `java.awt.image` package also have some useful things in them.

# Settings

- There are several things that we can set on the Graphics2D object that are used when we draw things. Here are some:
  - Paint – could be a color, but there are also gradients and textures
  - Stroke – determines how lines are drawn
  - Font – how you want text to appear
  - Transform – AffineTransform allows translate, rotate, scale, or shear

# Less Used Settings

- Composite – how colors combine when you draw over old stuff
- Clip – where your drawings will appear
- Render hints – other things like antialiasing



# More General Drawing

- Of course, Graphics2D objects aren't limited to just drawing on components.
- The Image class (and its subtype BufferedImage) will let you get Graphics objects that you can draw to and what you draw will be on the image.
- We'll typically do this even if we are drawing to a component to implement buffering which reduces flicker.

# Loading Images

- The easiest way to load images from disk is using `javax.imageio.ImageIO`.
- This class has read methods that take `File` or `URL` objects.

# Minute Essay

- How do you think we would go about creating animation?
- There is an IcP next class.