3-26-2012
Opening Discussion

- Minute essay comments:
  - Graphics will always involve Java libraries.
  - Graphics measured in pixels by default. We will see that can change.
  - Teaching back at Trinity.
  - Dealing with the uphill battle.
  - A CS minor does not a professional programmer make.
  - How to go about trying to do something completely new.
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
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.
The easiest way to load images from disk is using javax.imageio.ImageIO.

This class has read methods that take File or URL objects.
The “proper” way to write the paint method is to have a BufferedImage that you draw to, then draw the image to the provided Graphics2D object.

This is called double buffering and it prevents flicker and can make things run faster.
Mouse Events

- There are several types of events that relate to the mouse.
  - MouseClicked
  - MouseDragged
  - MousePressed
  - ...
- Listen to one of the publishers on a component.
  - mouse.clicks
  - mouse.moves
  - mouse.wheel
Key Events

- There are three event types for keys.
  - KeyPressed
  - KeyReleased
  - KeyTyped

- Listen to the keys object in a component to get these.

- Compare the key value in the event to values in the Key object.
  - if(e.key==Key.A) …
  - if(e.key==Key.Left) …
Animations

- For animations and many other things we want code to happen at regular intervals.

- We can set this up with a `javax.swing.Timer`.
  - `new Timer(delay:Int, ae: ActionListener)`
  - Delay is in milliseconds.
  - `Swing.ActionListener(handler: (ActionEvent) => Unit)`

- The function body will be executed at the desired intervals.
Last time we mentioned AffineTransforms, but didn't see what they can do.

Let's take some time now to write code that uses an AffineTransform in our drawing.
Questions?

IcP #6 is next class.