



Notes – Graphics

- I. User Interfaces
 - a. How does the machine get from objects to an image on the screen?
 - b. It stems from the `.setVisible()` call, but what happens behind the scenes?
 - c. Windows are repainted automatically under several conditions
 - i. Window, or part of it, is exposed from behind some other window.
 - ii. May become visible (from being minimized, etc)
 - iii. May get changed.
 - d. Every component knows how to display itself
 - i. `paintComponent()` draws the insides
 - ii. `paintBorder()` draws a border if needed
 - iii. `paintChildren()` draws any children of the component
 - iv. The three methods are called in that order.
 1. It's easy to have data spill into the border otherwise.
 2. Children are always on top.
 3. Drawn from back to front, just like using a pen.
 - v. Never call them directly!
 1. Call `repaint()`
 2. You'd only need to do this if you know you've changed something.
 3. `repaint()`: It happens fast enough for most purposes (50 ms)
 4. `repaint(int t)`: Happens within `t` milliseconds. Good for animations.
 5. `repaint(Rectangle r)`
 - a. Repaint only the subset of the window inside the rectangle.
 - b. Use this if only one piece of data changed on a component that's expensive to draw.
- II. Graphics Terms
 - a. Pixel = one dot on the screen. ~1,000,000 pixels on the whole screen.
 - b. Screens use standard Cartesian coordinates.
 - c. Origin is normally the upper left, with positive `y` DOWN, and positive `x` to the right.
 - d. A coordinate really defines the infinitely thin line *between* pixels.
 - i. See slide 100-15-8
 - ii. Drawing occurs below and to the right of the point you give.
- III. Graphics Class
 - a. Controls drawing.
 - b. font attribute
 - i. `Font(string face, int style, int size)`
 - ii. Faces
 1. Logical faces
 - a. Serif, SansSerif, Dialog, DialogInput, MonoSpace.
 - b. Always safe and portable. The system will select an appropriate font.
 2. Can use any font's name, but that's less portable.
 3. Usually best to stick with the five logical faces.
 - iii. Style
 1. BOLD, ITALIC, BOLD|ITALIC, PLAIN
 2. Members of Font
 - iv. Size
 1. Given in points
 2. One point is $\frac{1}{72}$ of an inch.
 - c. color attribute
 - i. Uses RGBA: Red, Green, Blue, Alpha
 - ii. Alpha sets how transparent the object is.

- 1. 0 means the object is completely opaque; 255 completely transparent.
 - iii. Useful colors are available as static fields: Color.RED etc.
 - iv. color.brighter() and color.darker()
 - v. Can explicitly create a color if really needed, but the above should be enough in most situations.
 - d. Many methods are available
 - i. drawString() draws a string at the coordinate given (lower left of text goes at that point)
 - ii. drawLine() draws an ordinary line.
 - iii. See list on CS100-15-14
- IV. Graphics2D Class
- a. Subclass of Graphics as of Java1.2
 - b. Advanced drawing support.
 - c. Don't *need* to know anything in Graphics2D for CS100
 - d. See Javadoc from Sun for in-depth tutorials.
 - e. Better shape support. (Shape class)
 - f. Transforms. (Eg: Skew a rectangle into a parallelogram)
 - g. Line Styles (Dotted, etc)
 - h. Strokes
 - i. Shape of pen, width, texture.
 - ii. Can change the appearance of corners.
 - i. Paint
 - i. Replaces Color.
 - ii. Includes not only color, but also textures and patterns.
 - j. More image processing. Graphics can do some simple image stuff, but Graphics2D can do some real manipulation.
- V. Clipping
- a. Not everything in the window should always be drawn.
 - b. Some elements may be behind another window.
 - c. Hidden elements may be expensive to draw
 - d. All drawing accounts for the Clip area – an arbitrary shape that's usually a rectangle.
 - e. Nothing outside the clip area will ever be drawn.
 - f. Three clip regions are defined
 - i. Device Clip. Restricts drawing to the physical limits of the screen.
 - ii. Window Clip. Won't draw over other windows.
 - iii. User Clip
 - 1. User-defined area that can cut out expensive pieces that haven't changed since the last paint.
 - 2. This is the only clip that isn't automated.