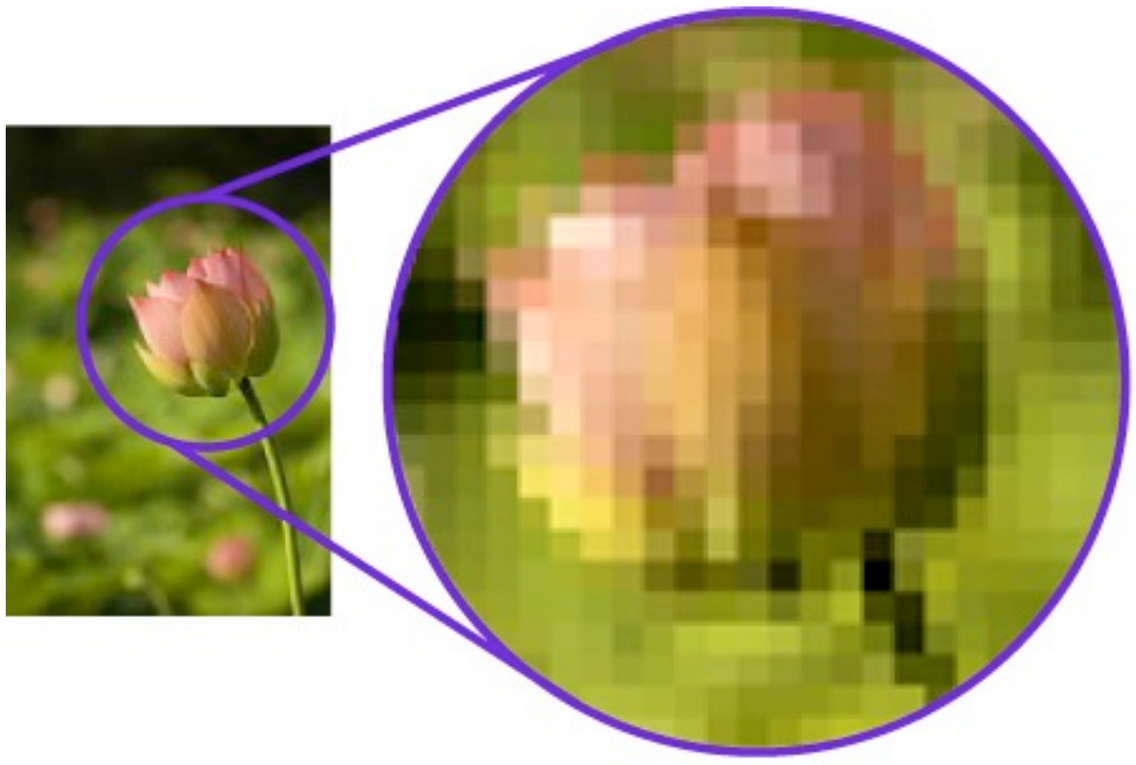
Raster Graphics and Displays

CS 355: Interactive Graphics and Image Processing

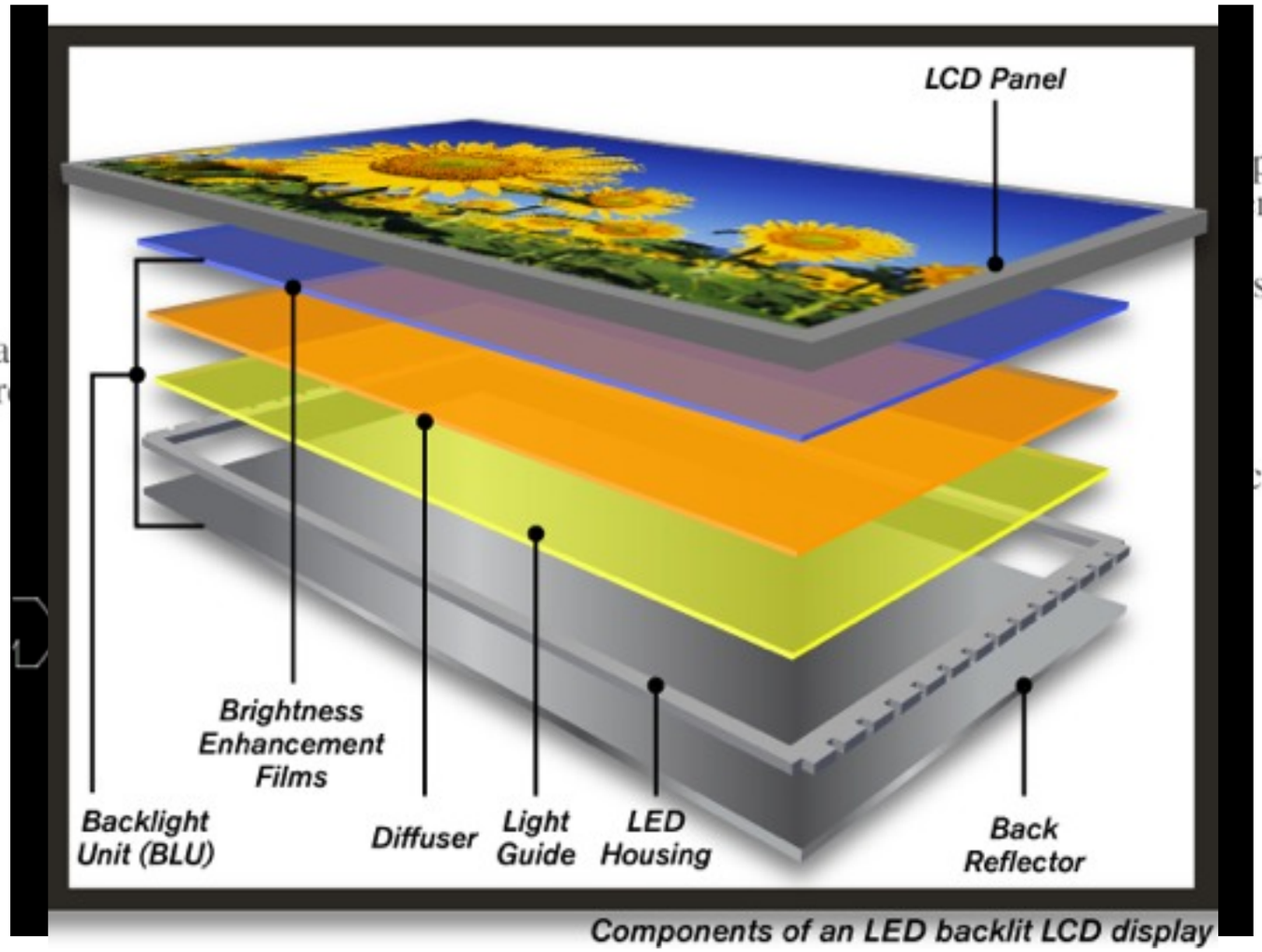
Raster Images



Most digital displays are made up of discrete dots called “pixels” (short for *picture elements*)

Quick History

- Vector graphics
- Raster (CRT)
- Raster (Digital)



Components of an LED backlit LCD display

Raster Images

- Size: number of pixels (height x width)
- Bit depth: bits of precision for each pixel
  - Binary (true black and white)
  - 2-bit gray (4 shades)
  - 8-bit gray (256 shades)
  - 12-bit gray (X-rays / CT)
  - 24-bit color (8 bits each of R, G, B)

Common Displays

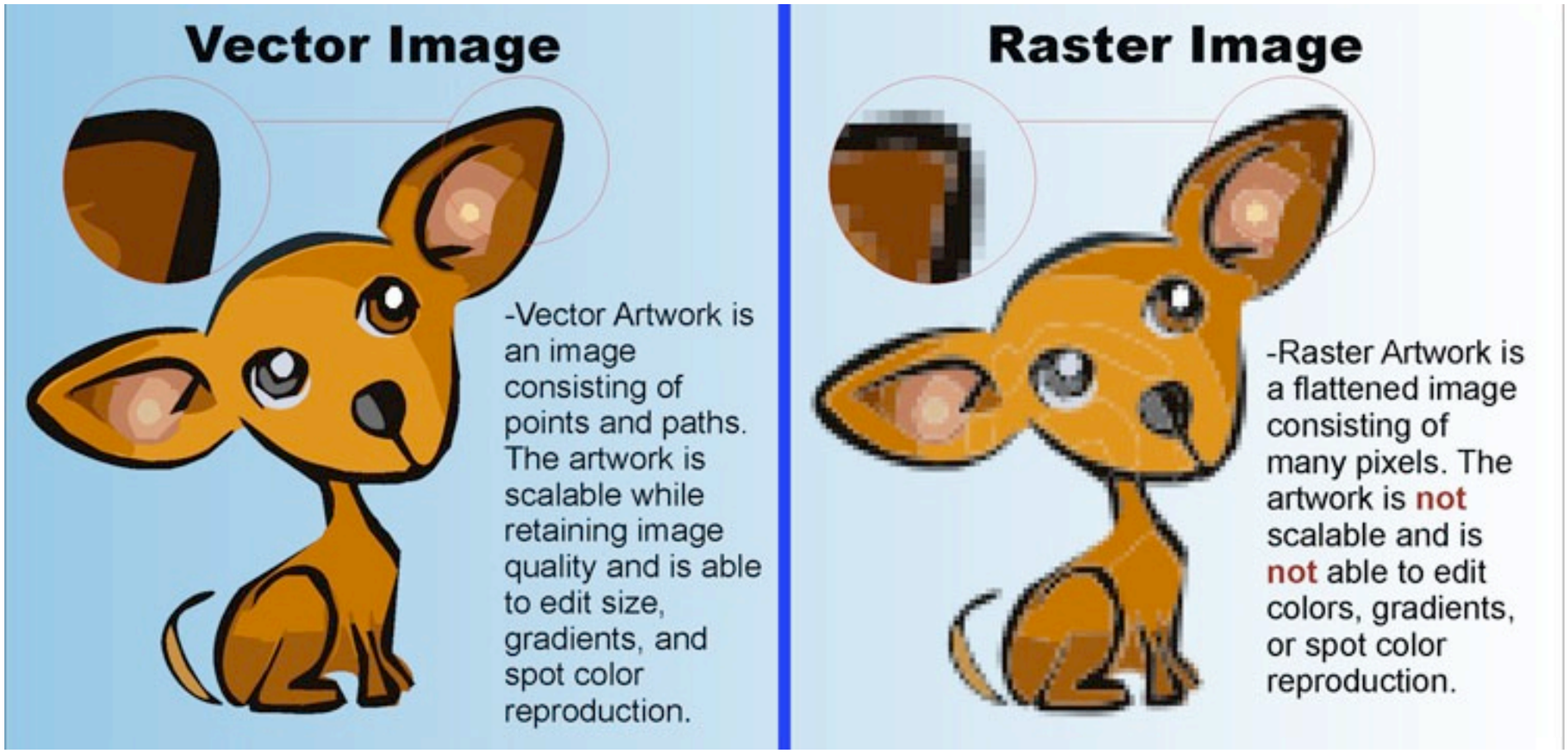
- "Standard definition"
  - 4:3 aspect ratio
  - 480 horizontal lines
- "High definition"
  - 16:9 aspect ratio
  - Either 720 or 1080 horizontal lines

- Interlacing
  - Send full screen 30 times / second ("progressive")
  - Send half screen 30 times / second ("interlaced")

➡ Odd lines, then even lines...

How you display pictures isn't how you have to store them!

Raster vs. Vector Graphics



Vector:
Continuous Curves

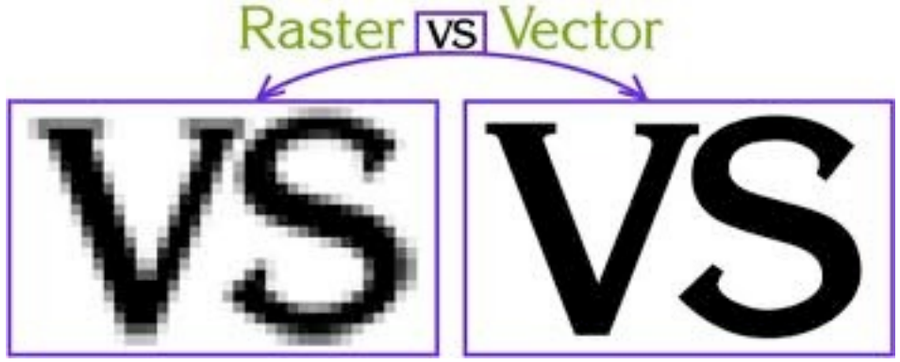
Raster:
Lots of dots

Rasterization

- Vector graphics are higher quality
- But everybody uses raster displays
- Have to convert to a raster image first (But get to target resolution to device!)
- This process is called *rasterization* (sometimes *scan conversion*)

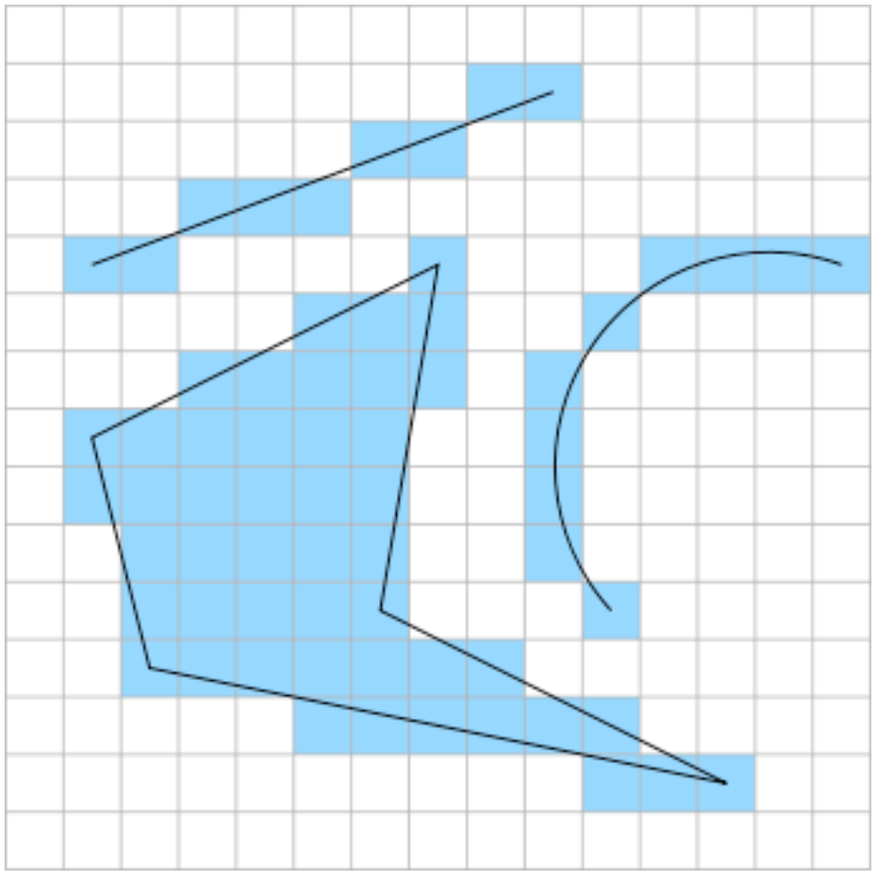
Example: Text

- Most fonts aren't stored as bitmaps (images)
- Stored instead as curves representing the *outline* of the characters
- When displayed or printed, *can convert to whatever resolution your device supports*



What Could Go Wrong?

Which are in?



Which are out?

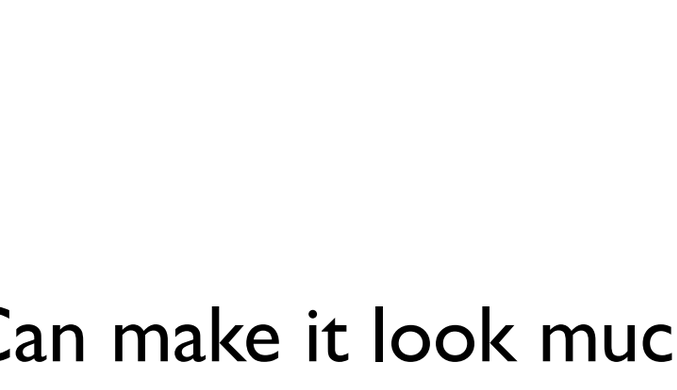
Aliasing

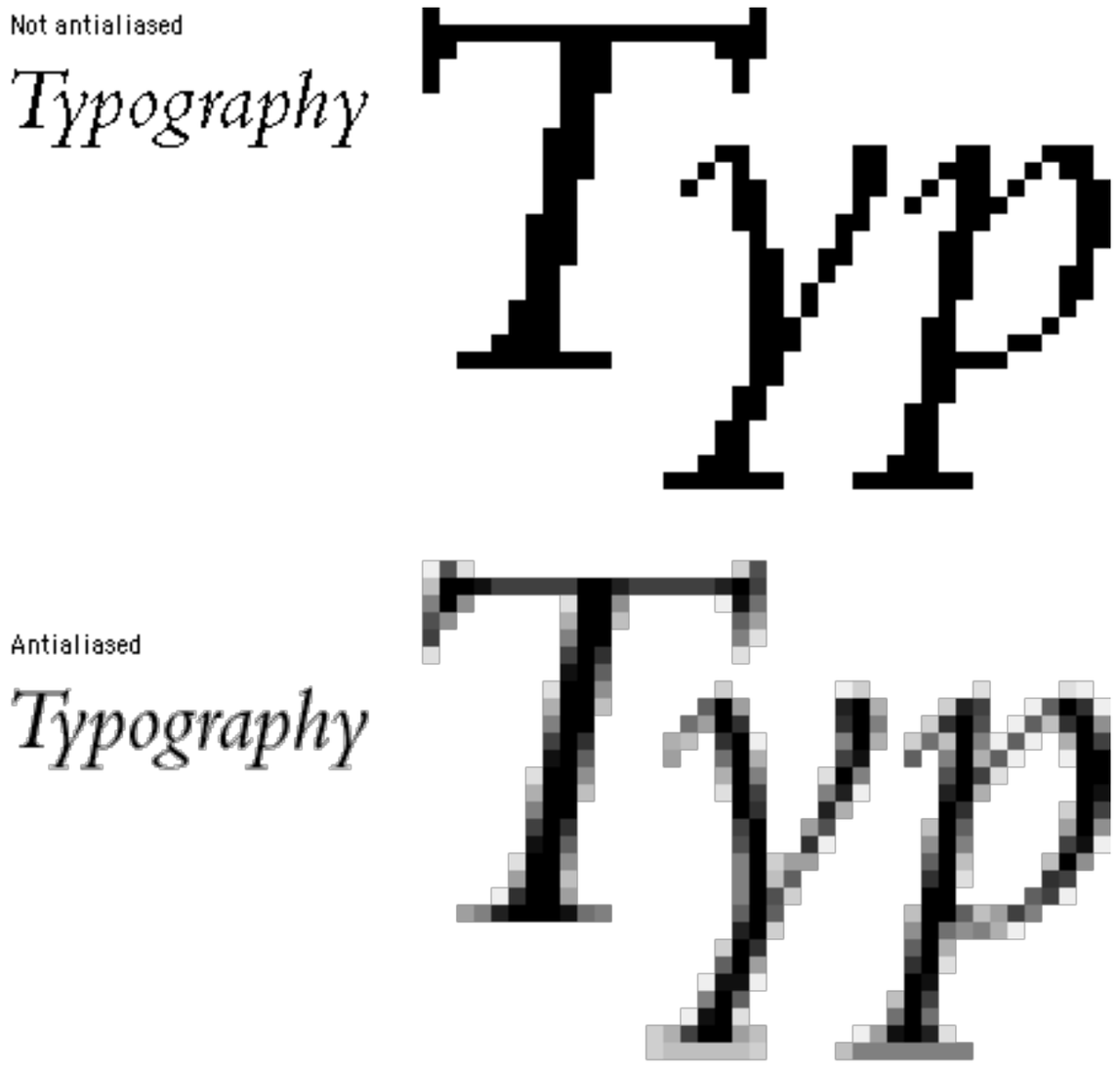
“Jaggies”

Antialiasing

- Can make it look much sharper by blurring a bit (no, really!)

- Key: *partially* fill in pixels during rasterization





Screen Buffers

- To drive a raster display the computer must store a raster image of what goes on it (usually in graphics card)
- Called a *screen buffer*
- Display hardware regularly sends the contents of the screen buffer to the screen
- You draw by writing into the buffer

A Little Note about Color Buffers

- Color images are usually 24 bits
- Memory words are usually 32 bits (or some multiple of this)
- For speed, word-align your pixels (works for other data as well!)
- But, but...

Alpha Channels

- But what about the wasted byte per pixel?
- Usually used for the *alpha channel*
  - Controls transparency
  - Useful for compositing
  - 0 = transparent, 255 = opaque
- More on this later...

Handling Buffers

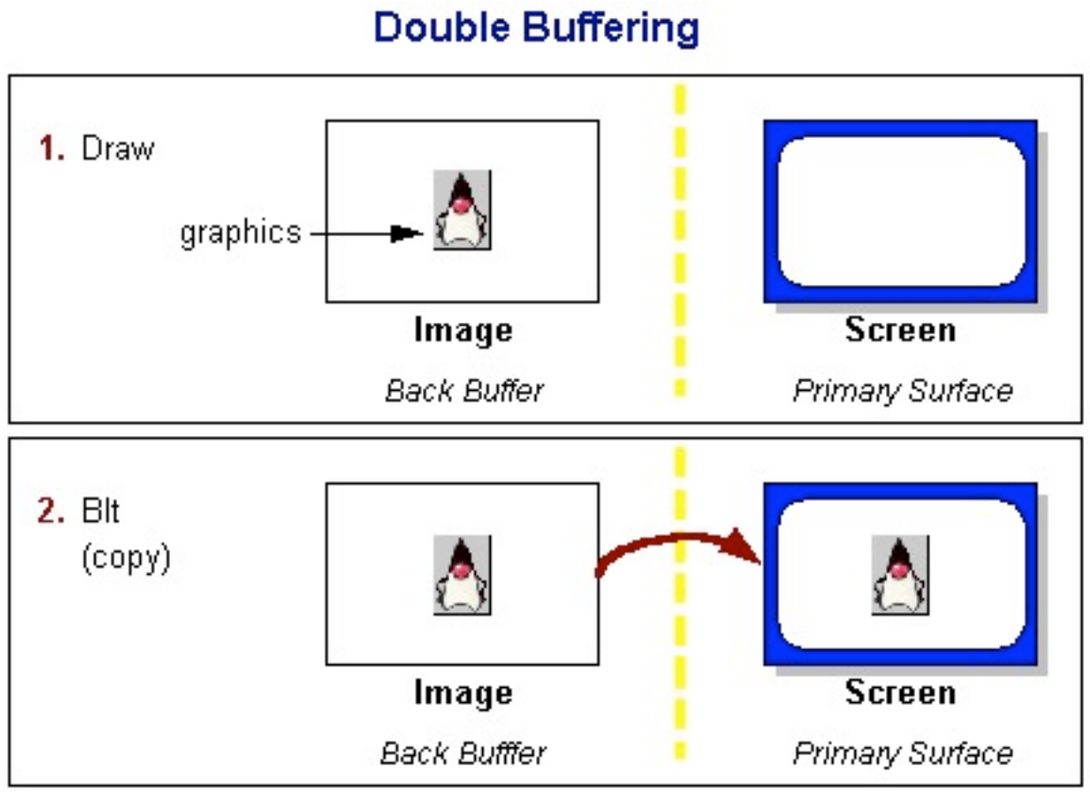
- What if the screen refreshes from the buffer while you're making changes?
- Two strategies:
  - Single buffer:
    erase / redraw only what's changed
  - Double buffer

Single Buffer

- Erase / redraw only what's changed
  - Use clipping
  - Determining what's in that area is usually not worth it
- Might lead to flicker in these areas, *but only these areas*
- Java:
  - "repaint" tells system to redraw *part* of display
  - Sets clip area and then calls regular "paint" routine to redraw everything
- Used mainly in things that don't update a lot

Double Buffering

- Don't really draw to the real screen buffer
- Draw to *offscreen buffer*
- Copy buffers (fast)
  - Some systems support switching with just a pointer change
- Most common for games, animation, etc.



Layers upon layers...

- In most systems there are lots of layers:
  - Drawing API
  - GUI
  - OS
  - Graphics drivers
  - Graphics cards
  - Display

Next time...

- Graphical user interface software (especially Java)
- Model-view-controller architecture