Output Model

A picture is worth a thousand words

(and let’s not forget about sound)

Coordinate Systems

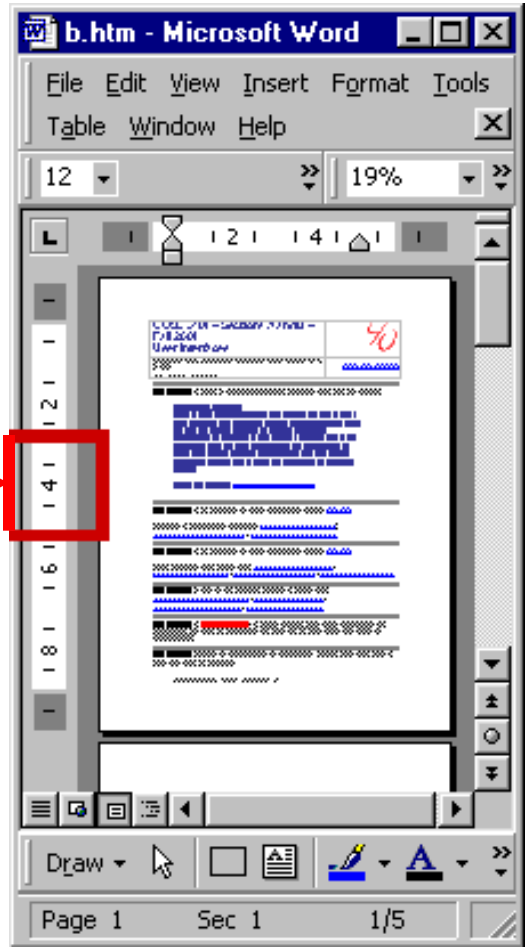
- Device coordinates
- Physical coordinates

Device Coordinates

- Most natural units for the output device
- Typically dots or pixels
- Origin possibilities
  - Centre
  - Bottom left
  - Upper left

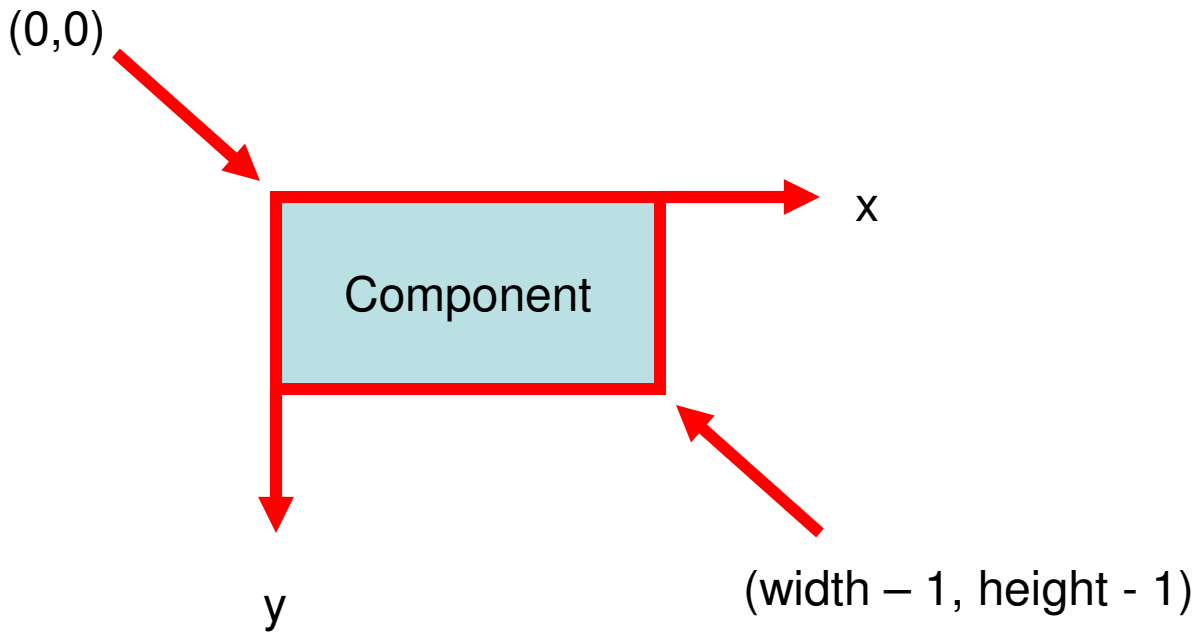
Physical Coordinates

- Printed page → inches, cm



- Architectural drawings → feet, meters

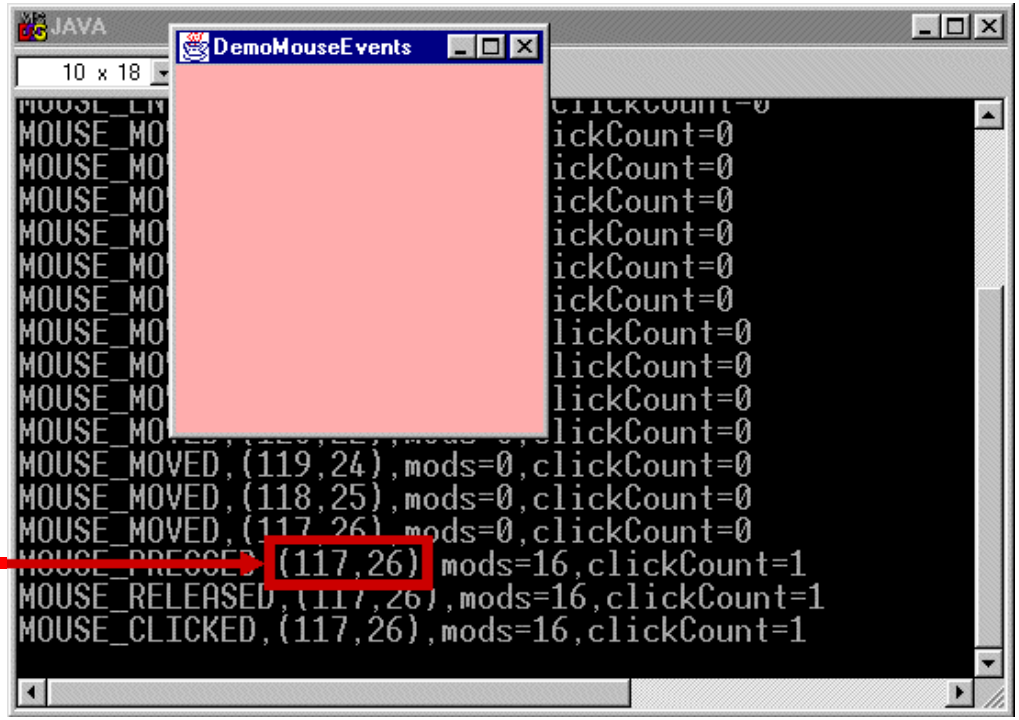
Java's Coordinate System



Example Program

DemoMouseEvents.java

Shown earlier



(x, y) coordinate of pointer in component

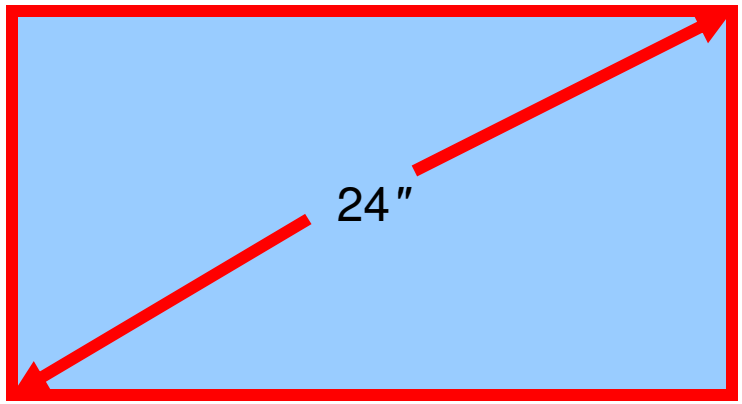
Pixels

- A Pixel is a “picture element”
  - a single point in a graphic image
  - A graphics display is divided into thousands (or millions) of pixels arranged in rows and columns
  - The pixels are so close together they appear connected
  - The number of bits used to represent each pixel determines how many colours or shades of grey can be represented
  - For a B&W (black and white) display, each pixel is represented by 1 bit
  - With 8 bits per pixel, a monitor can display 256 shades of grey or 256 colours (Note: $2^8 = 256$)

An image presented on a display is composed of pixels

Display Size

- Usually specified in “inches”
- Value cited is the diagonal dimension of the raster -- the viewable area of the display
- E.g., a 24″ monitor



Resolution

- Resolution is the number of pixels on a display
- Usually cited as *n by m*
  - *n* is the number of pixels across the display
  - *m* is the number of pixels down the display
- Typical resolutions range from…
  - 640 by 480 (low end), to
  - 1,920 by 1,200 (high end)

Video RAM Requirements

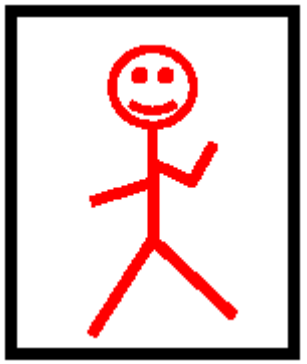
- Total number of pixels is $n \times m$
- Examples
  - 640 × 480 = 307,200 pixels
  - 1,920 × 1,080 = 2,073,600 pixels
- Video RAM required equals total number of pixels times the number of bits/pixel
- Examples
  - 640 × 480 × 8 = 2,457,600 bits = 307,200 bytes = 300 KB
  - 1,920 × 1,080 × 24 = 49,766,400 bits = 6,220,800 bytes = 6,075 KB = 5.93 MB
  - Note: 1 KB = 2^{10} = 1024 bytes, 1 MB = 2^{20} = 1,048,576 bytes

Video RAM (KB) By Resolution

Resolution	Bits per pixel		
	8 bit	16 bit	24 bit
640 x 480	300	600	900
800 x 600	468.75	937.5	1406.25
1024 x 768	768	1536	2304
1152 x 1024	1152	2304	3456
1280 x 1024	1280	2560	3840
1600 x 1200	1875	3750	5625

Aspect Ratio

- Aspect ratio is the ratio of the width to height of a display screen
- For a 640 by 480 display, the aspect ratio is 640:480, or 4:3
- Related terms
  - Landscape
    - The width is greater than the height
  - Portrait
    - The height is greater than the width



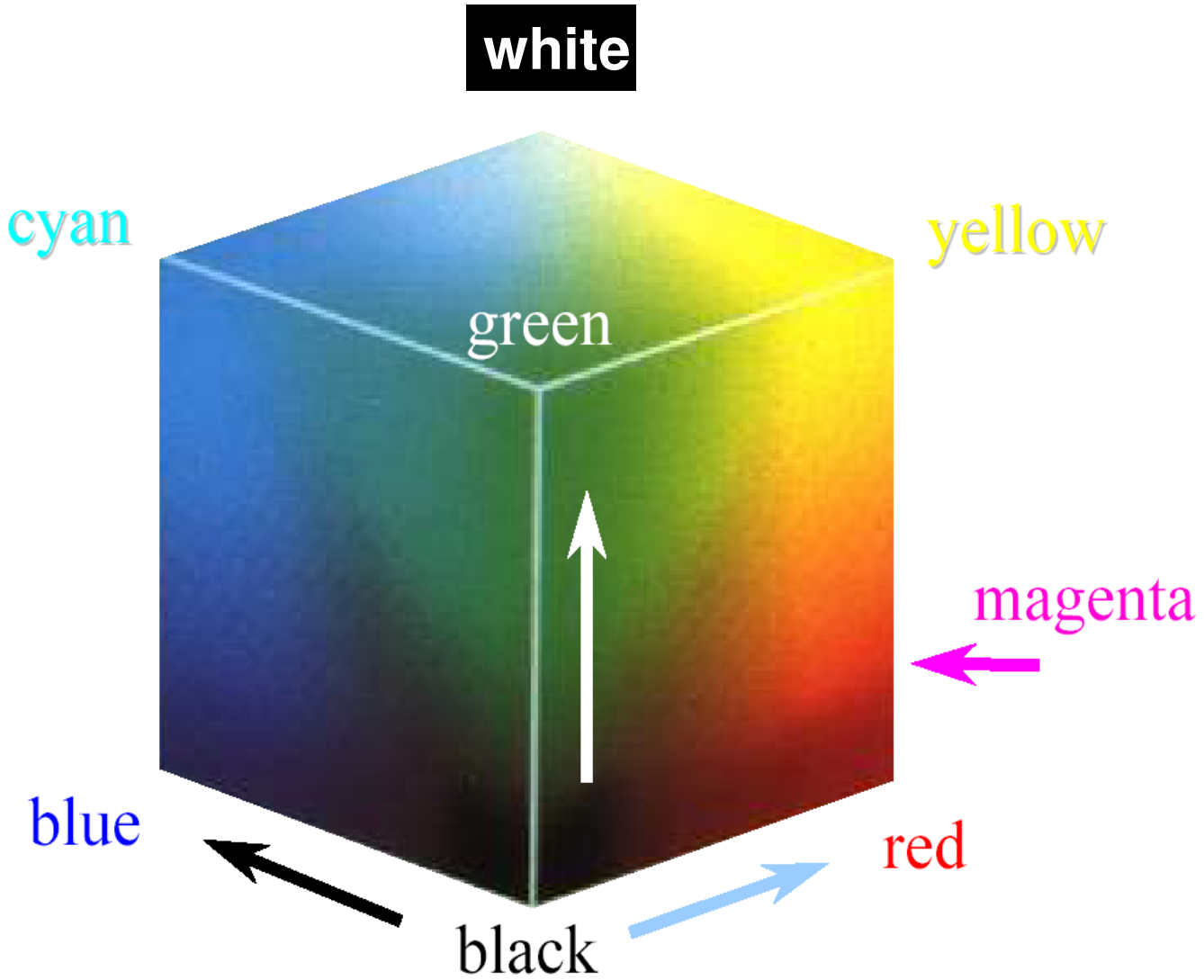
Dot Pitch & Pixel Density

- Dot pitch is a measure of the diagonal distance between pixels on a display
  - One of the principal characteristics that determines the quality of a display
  - The lower the number, the crisper the image
  - Cited in mm (millimeters)
- Pixel Density
  - Number of pixels in H or V dimension
  - Higher number – crisper image
  - Cited in ppi or dpi
  - Typical values: from 96 (desktop) to 400+ (mobile phone)

Colour

- Two models for colour
  - RGB
    - Individual specifications for **RED**, **GREEN**, and **BLUE**
  - HSB
    - Individual specifications for hue, saturation, and brightness
    - Together, hue and saturation are called chrominance; they represent the colour
    - Hue is the distinction between colours (e.g., red, orange, yellow, green, etc.)
    - Saturation is the purity of a colour, or the amount of grey in proportion to the hue
      - High saturation very intense
      - Low saturation washed out
      - Zero saturation white or neutral grey
    - brightness is also called luminance or intensity

RGB model

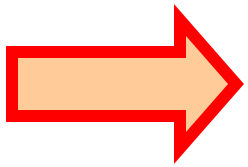


RGB Model (2)

Color	Red	Green	Blue
Red	255	0	0
Green	0	255	0
Blue	0	0	255
Yellow	255	255	0
Cyan	0	255	255
Magenta	255	0	255
White	255	255	255
Black	0	0	0

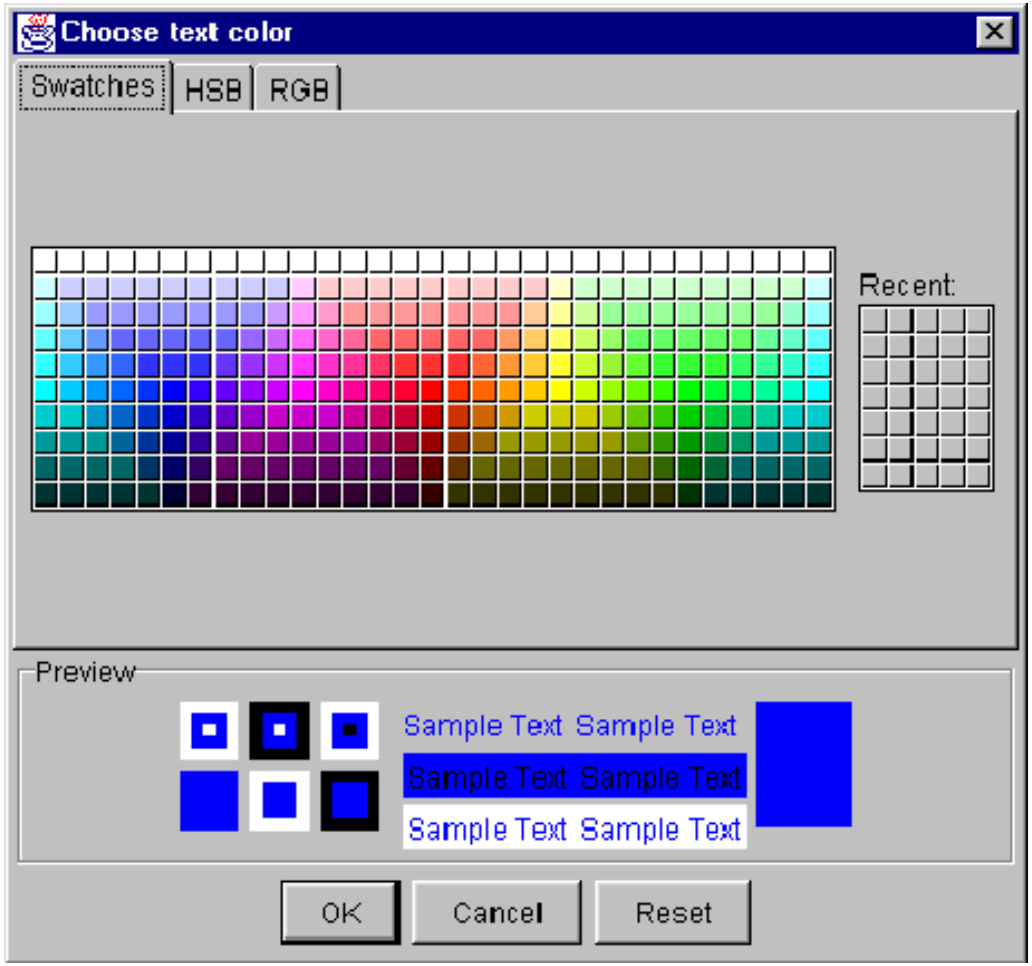
Colour Choosers

- Control for colour usually employs a colour chooser (aka colour picker)
- Colour selected three ways:
  - A pre-defined palette
  - HSB values
  - RGB values



Java's JColorChooser (1)

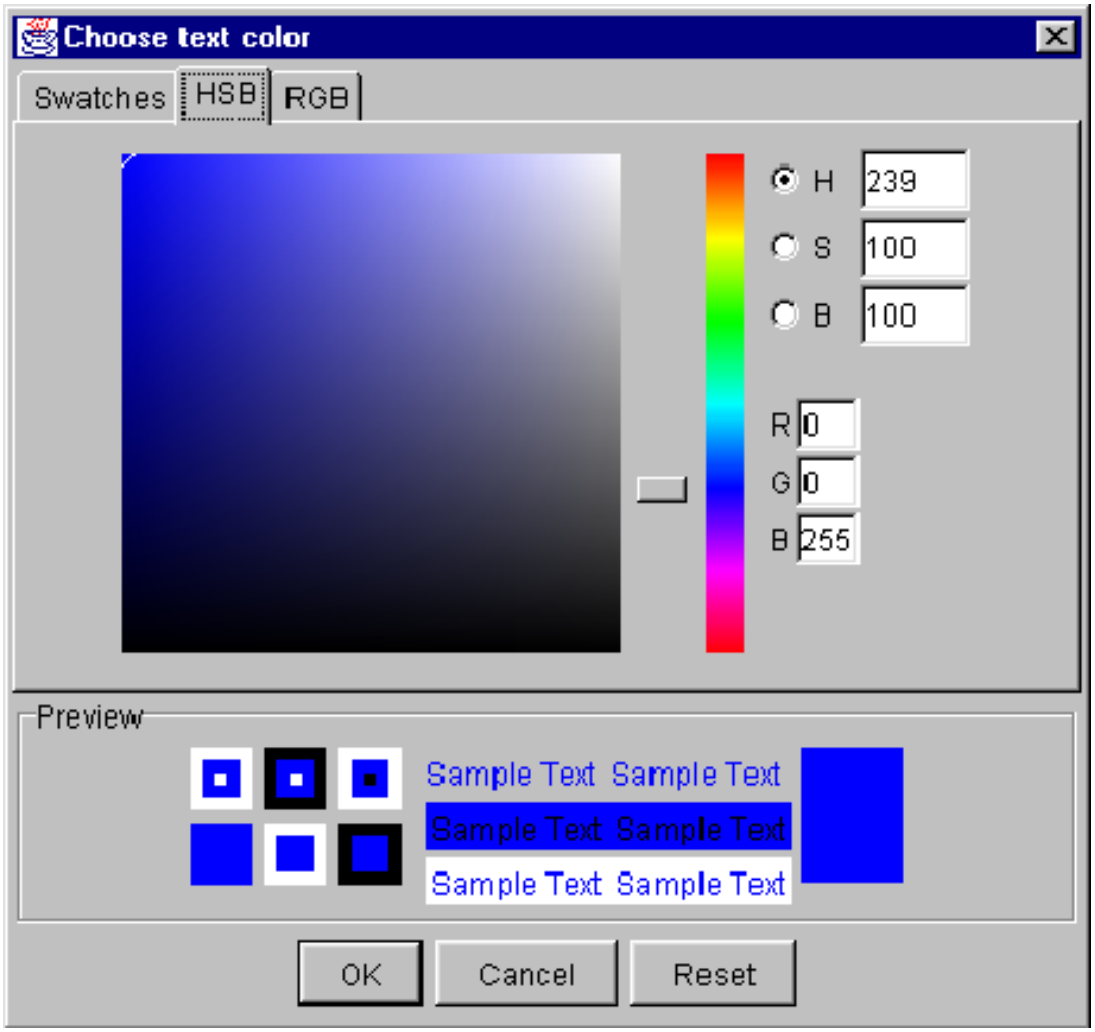
Pre-defined pallete



For a demo, see DemoMenu2.java

Java's JColorChooser (2)

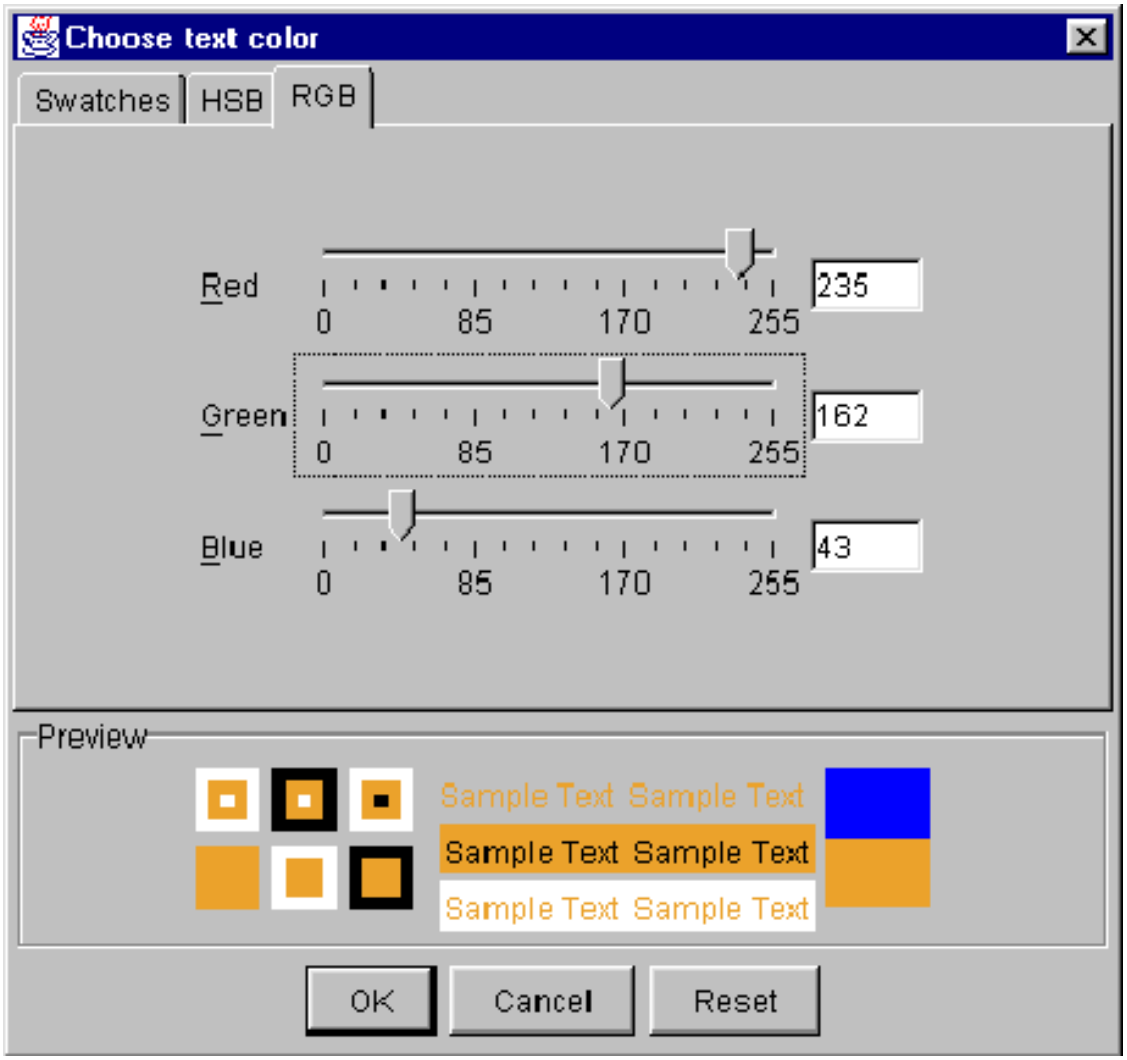
HSB



For a demo, see DemoMenu2.java

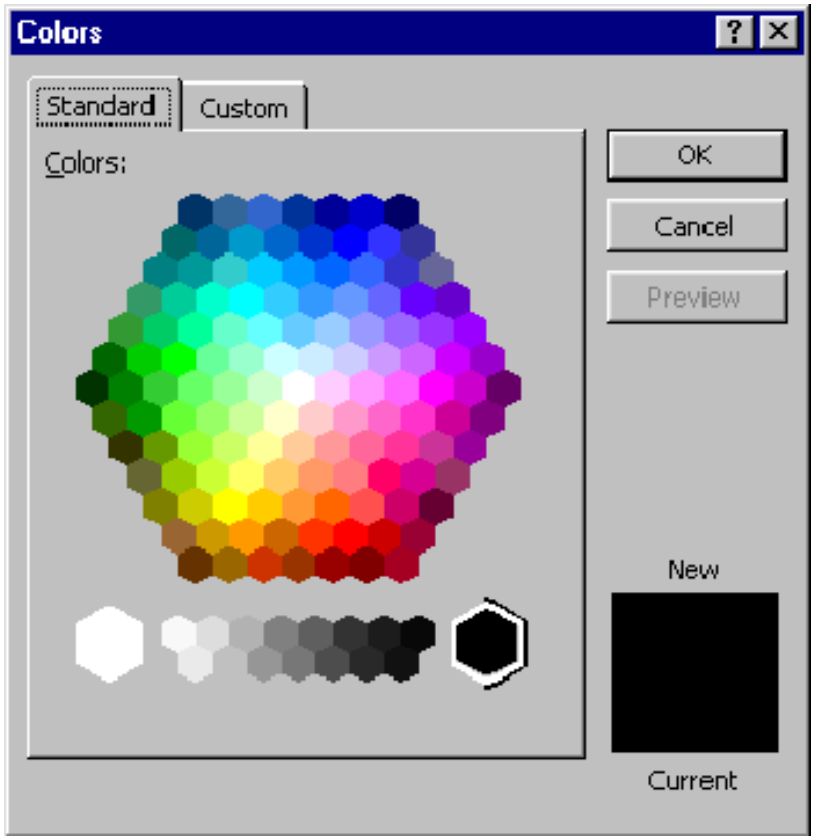
Java's JColorChooser (3)

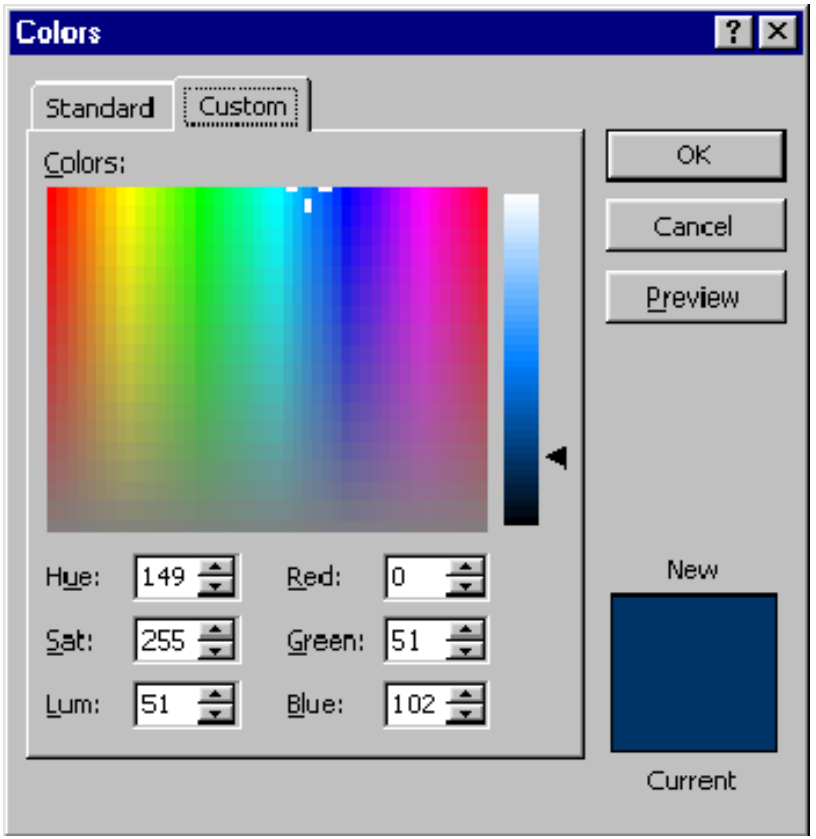
RGB



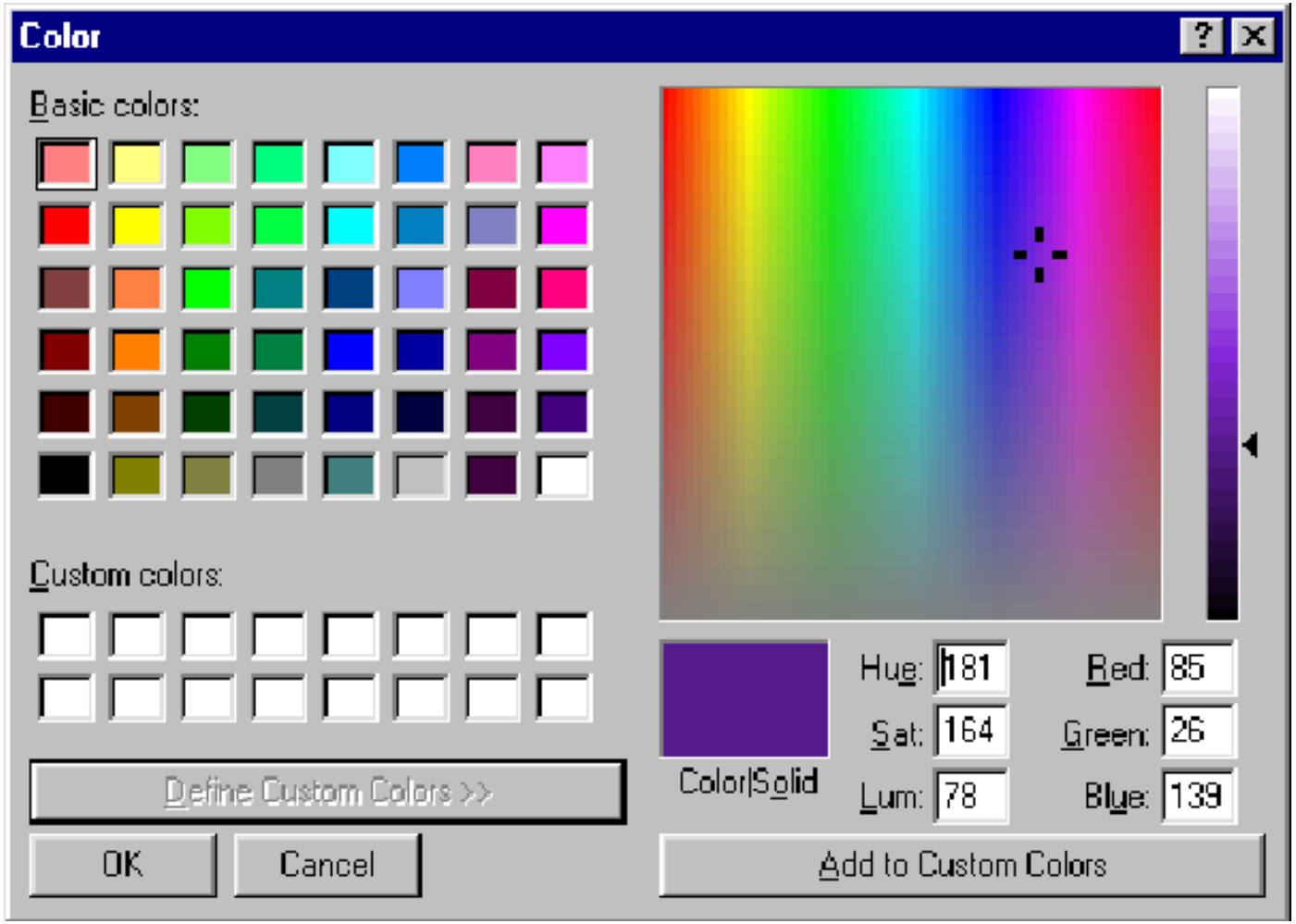
For a demo, see DemoMenu2.java

Microsoft Office

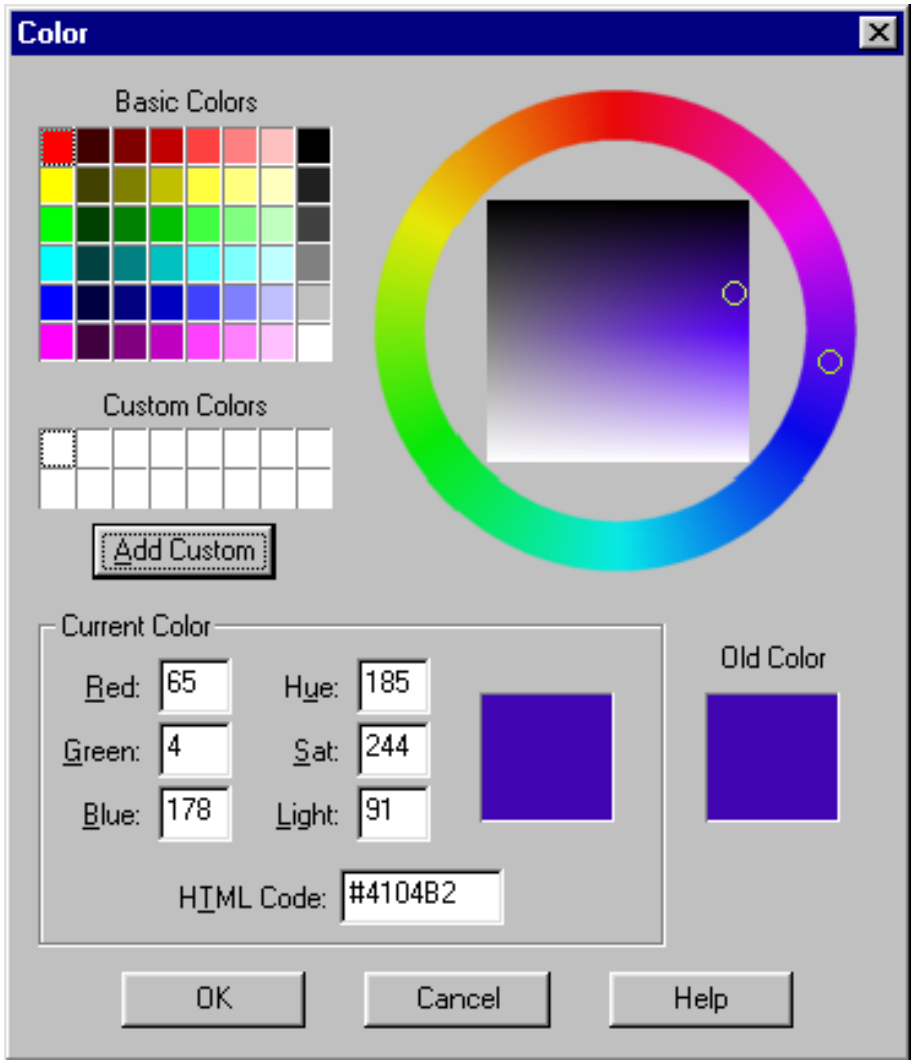




Netscape Navigator and Microsoft IE



Paint Shop Pro

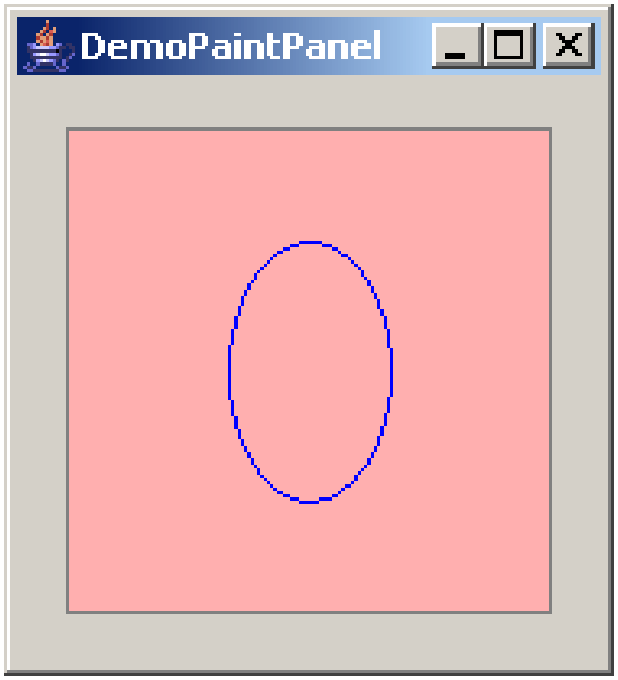


Drawing

- Java's "J" components are the building blocks
- of graphical user interfaces
- At a lower level, Java provides a set of drawing primitives for
  - Shapes
  - Lines
  - Curves
  - Images
  - Text

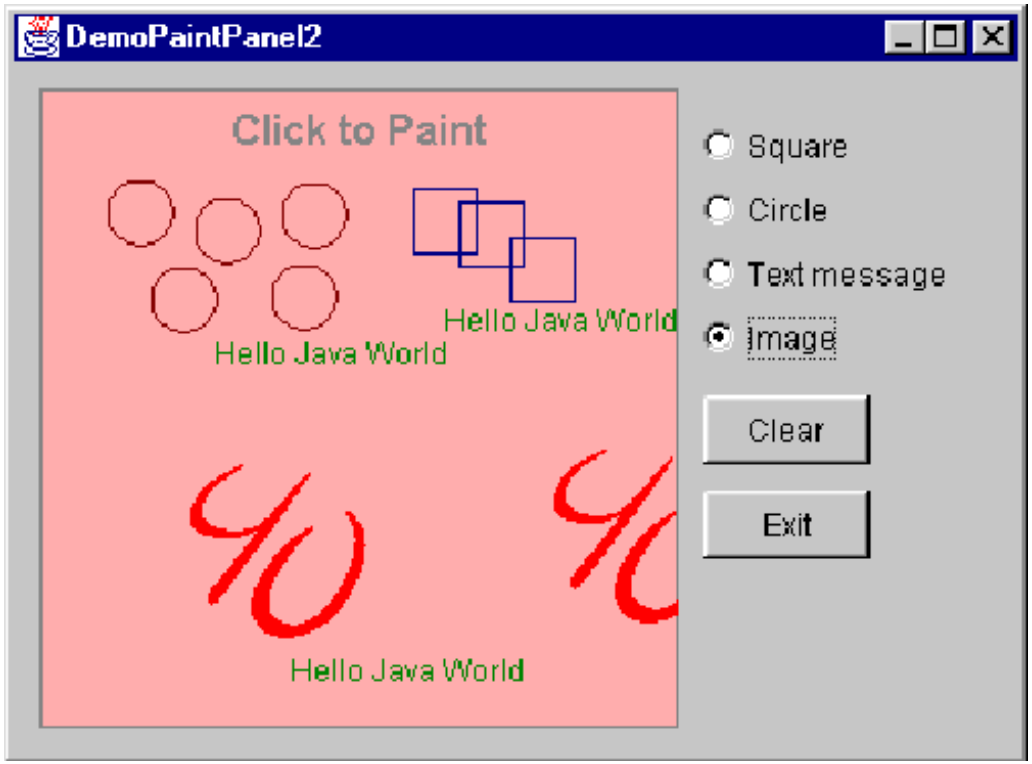
Example Program

DemoPaintPanel.java



Example Program

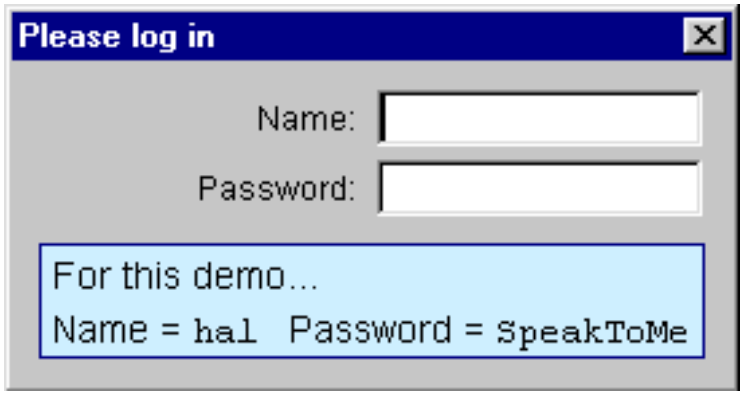
DemoPaintPanel2.java

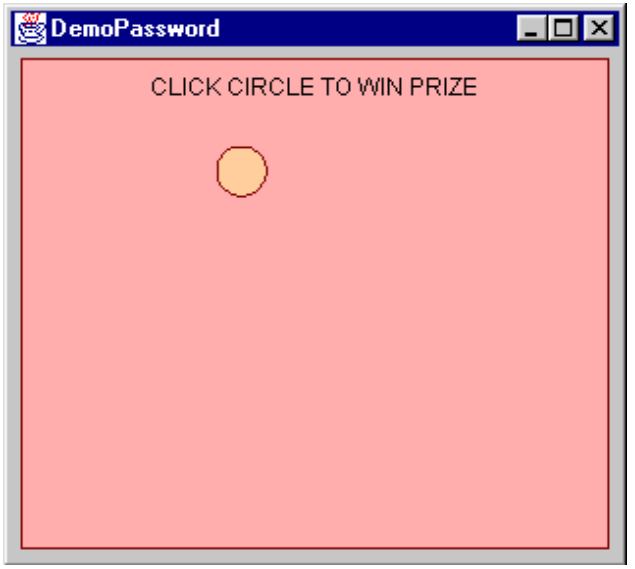


Example Program

DemoPassword.java

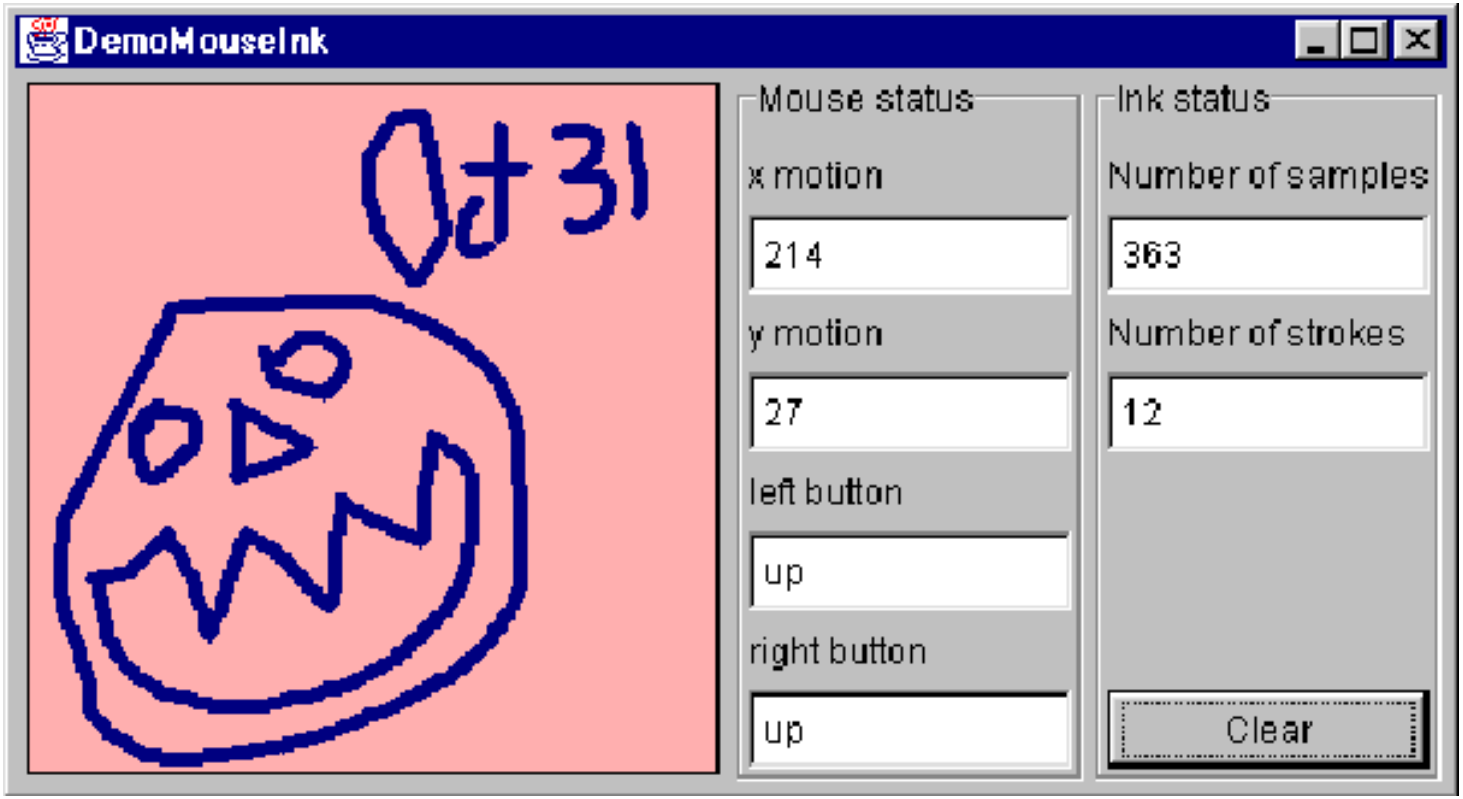
(Shown earlier)





Example Program

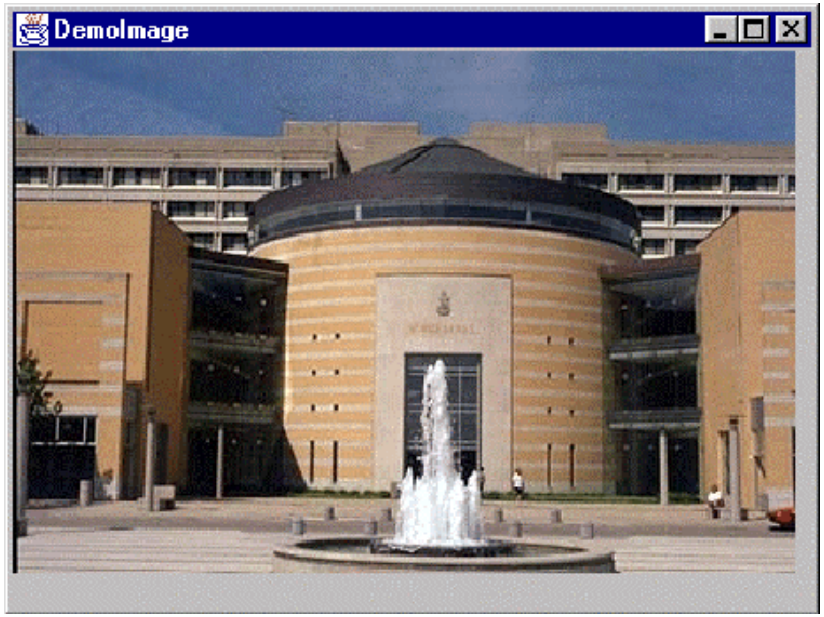
DemoMouseInk.java

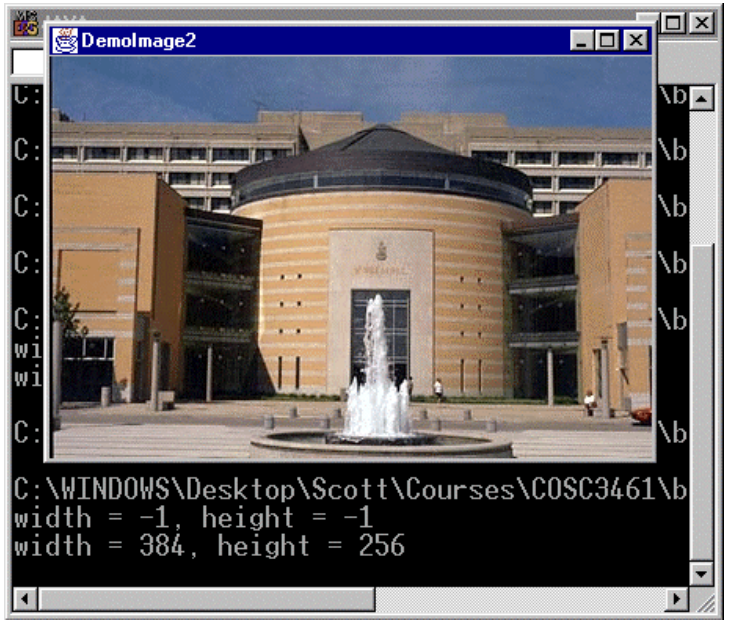


Images

DemoImage.java

DemoImage2.java





Example Program

DemoImageSizePosition.java

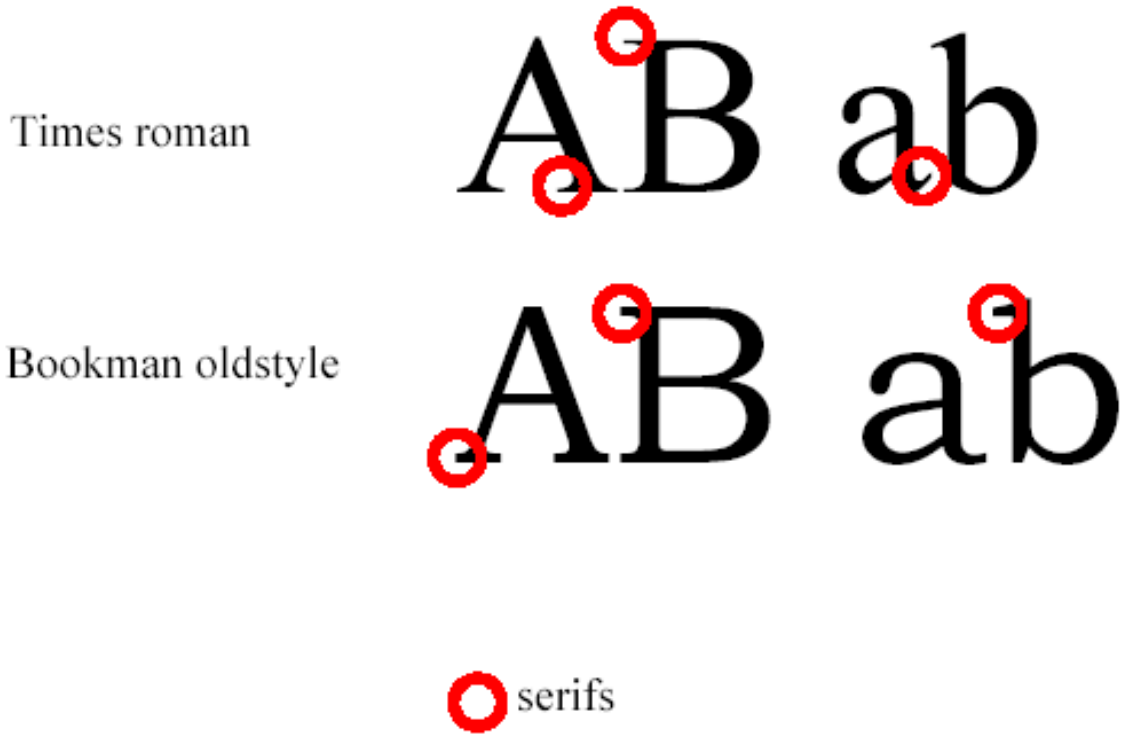
Text

- Characterized by
  - Font family
  - Style
  - Size and Spacing

Font Families

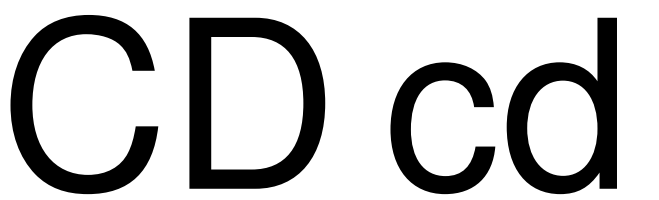
- Three types
  - Serif
    - A serif is a short line extending from and at an angle to the upper and lower strokes of a letter
    - Serif fonts facilitate human perception in distinguishing among letters
  - Sans serif
    - Without serifs
  - Monospaced or fixed-pitch
    - Each character occupies the same amount of horizontal space (cf. variable pitch)

Serifs Illustrated

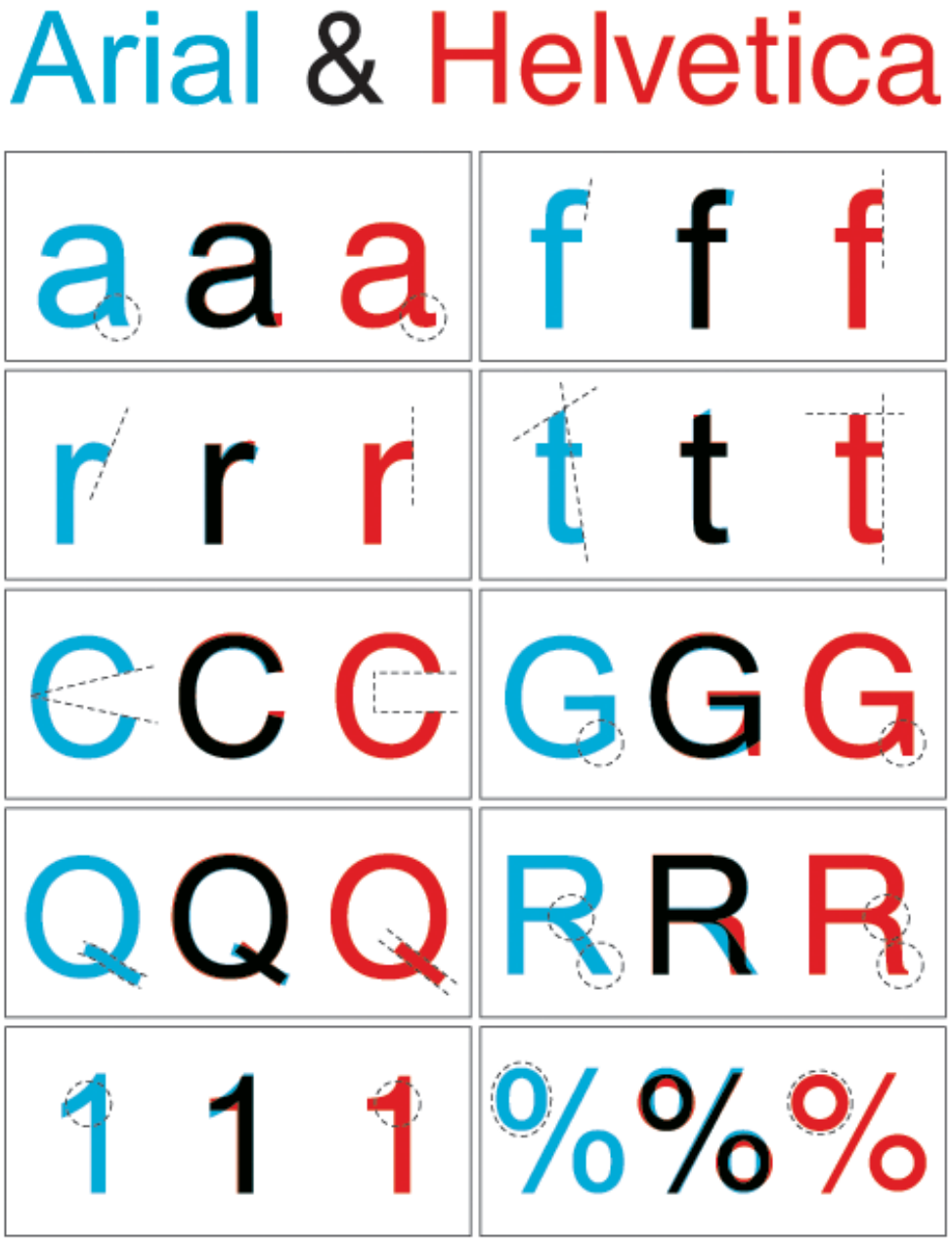


Sans Serifs Illustrated

Arial



Lucida Sans


Arial & Helvetica
a a a
f f f
r r r
t t t
C C C
G G G
Q Q Q
R R R
1 1 1
% % %

Monospaced Illustrated

Courier New IM im

Lucida Console IM im

Font Style

- Plain — Hello Java World

- Italic — *Hello Java World*

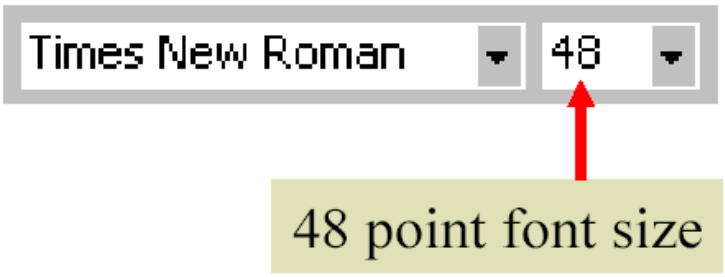
- Bold — **Hello Java World**

- Italic + bold — ***Hello Java World***

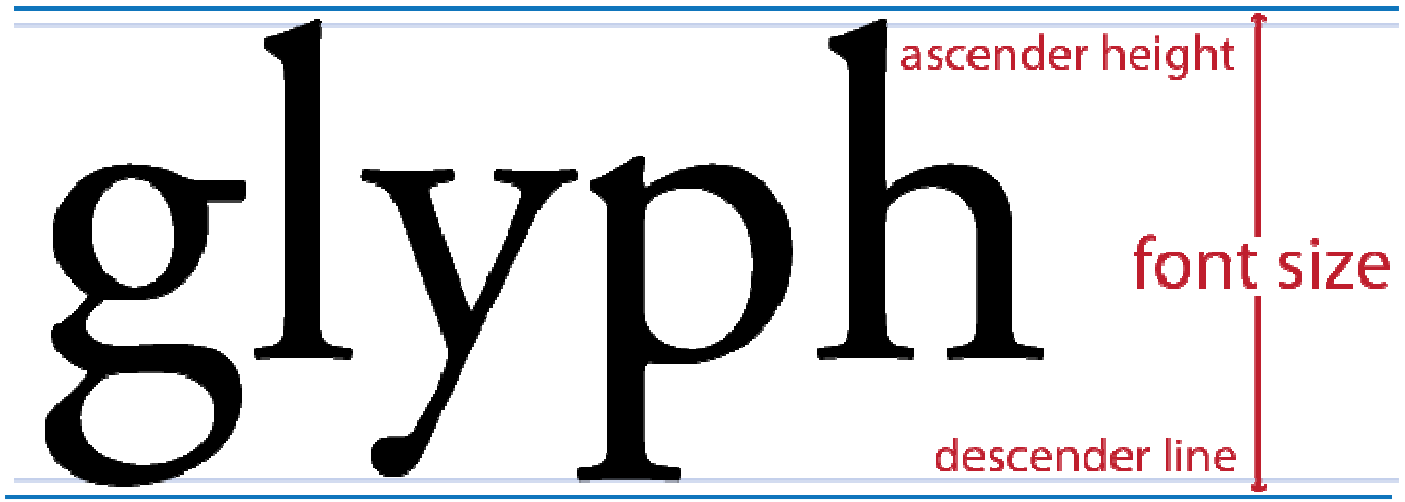
Font Size

- Font size is measured in points
- A point is the smallest typographical unit of measurement
- 1 point = 1 / 72 inch (~0.3528 mm)
  - 12 points = 1 pica, 6 picas = 1 inch

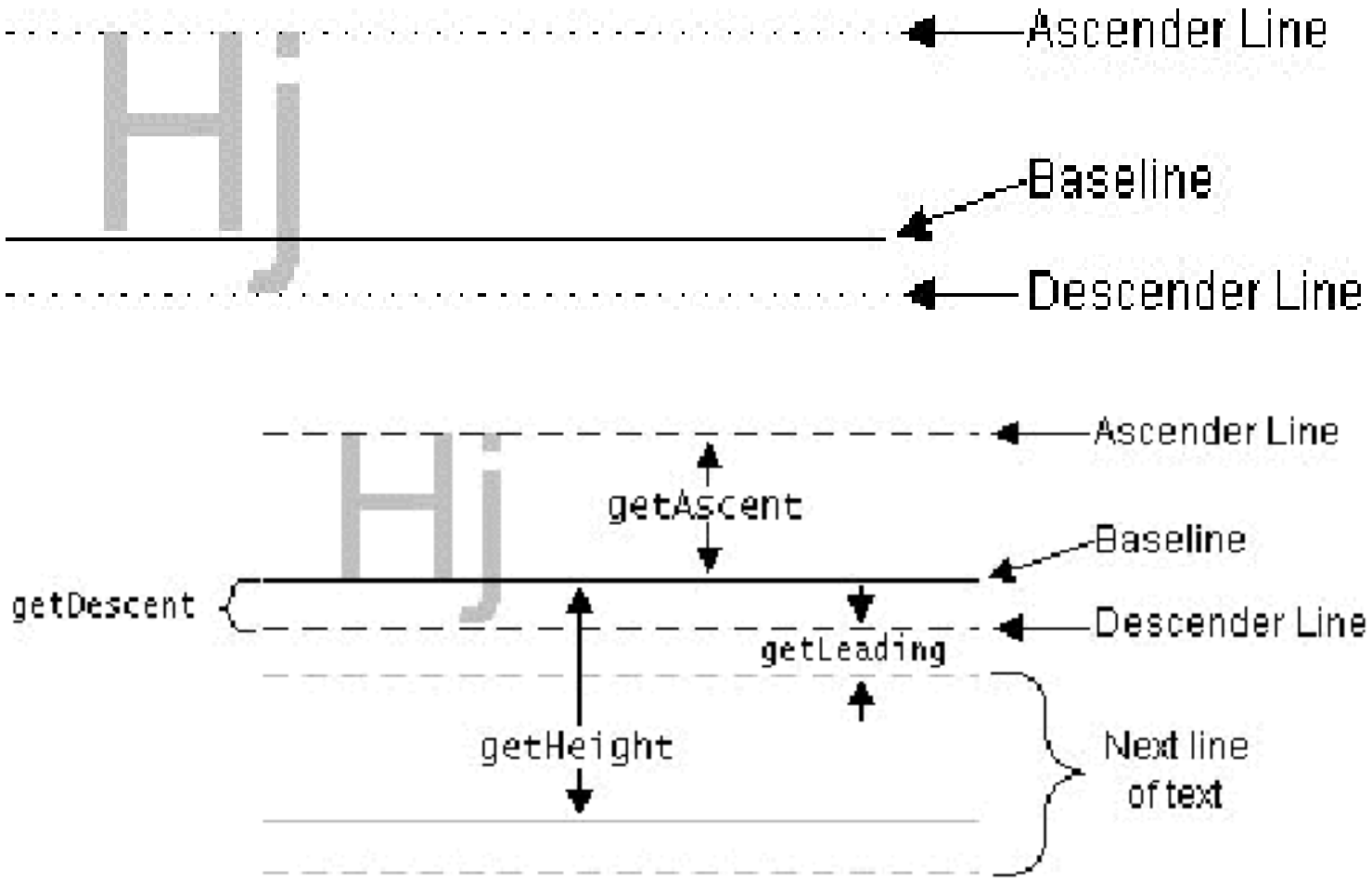
Hello Java World



Font Size (2)



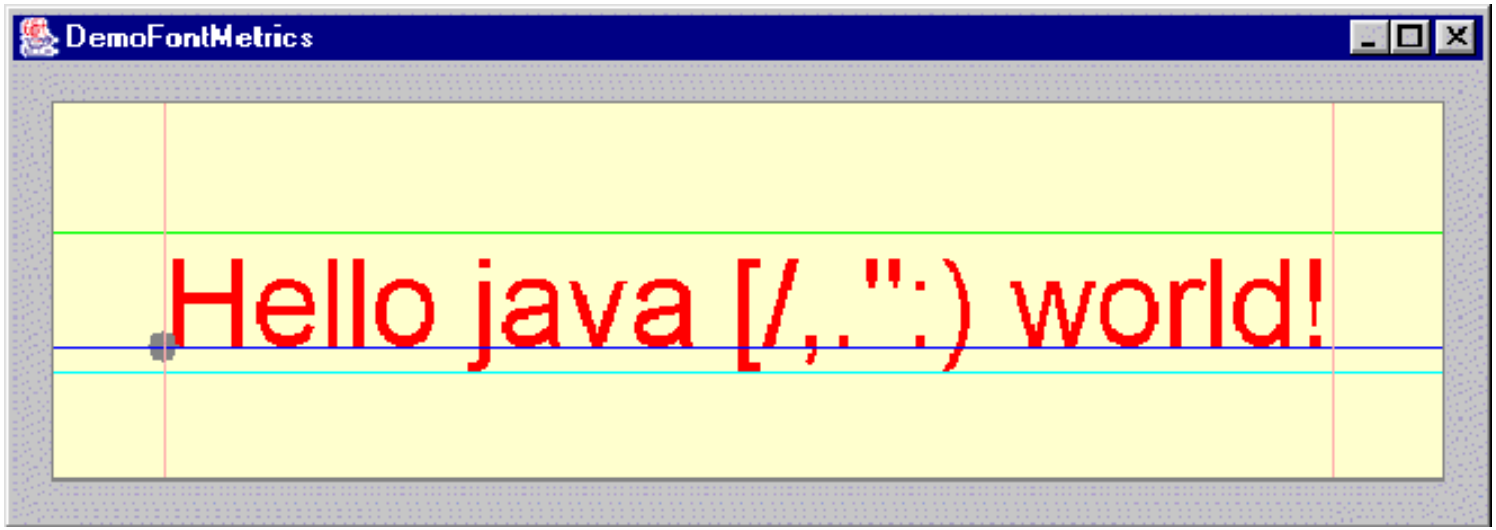
Font Spacing



See `FontMetrics` API

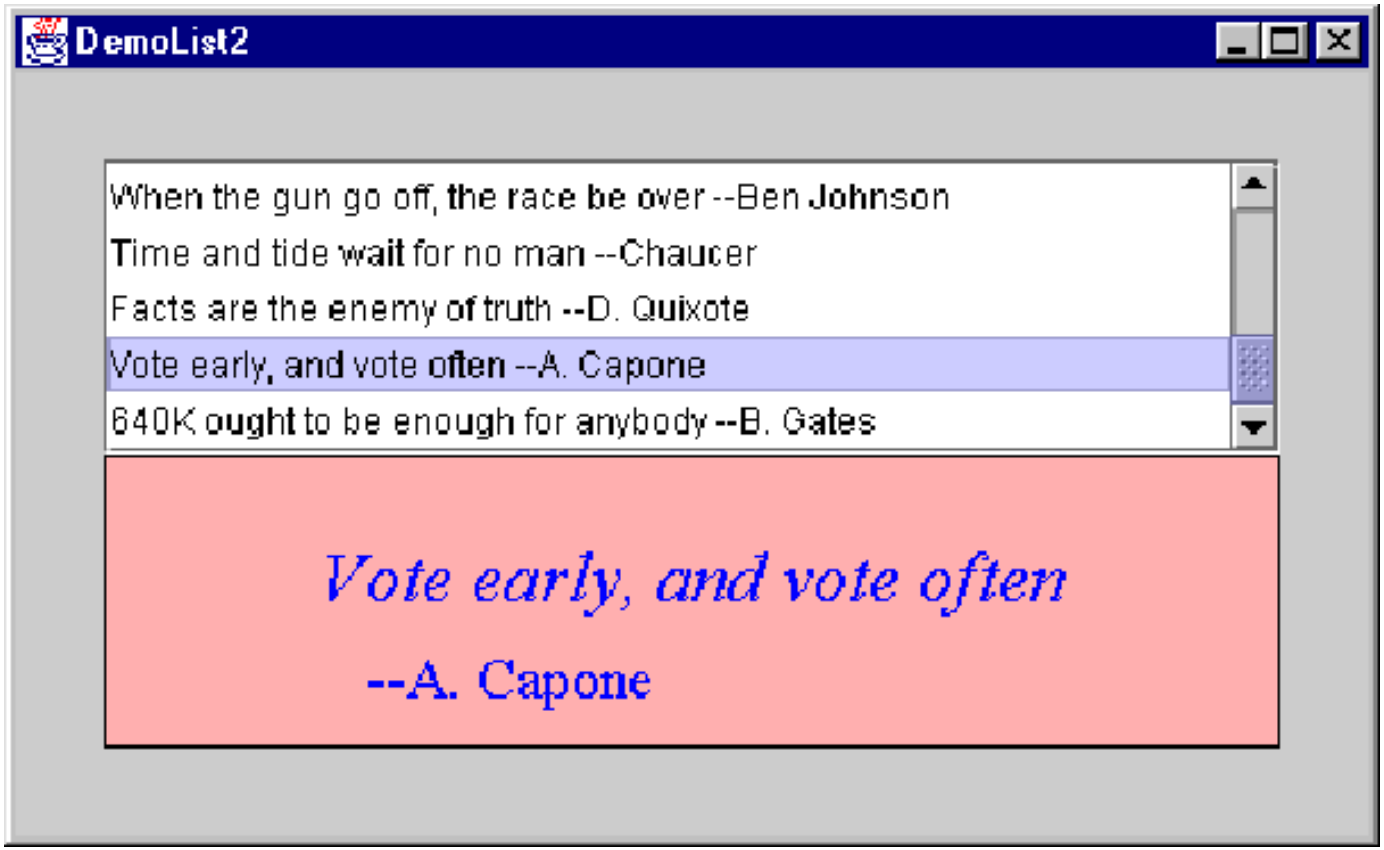
Example Program

DemoFontMetrics.java



Example Program

DemoList3.java



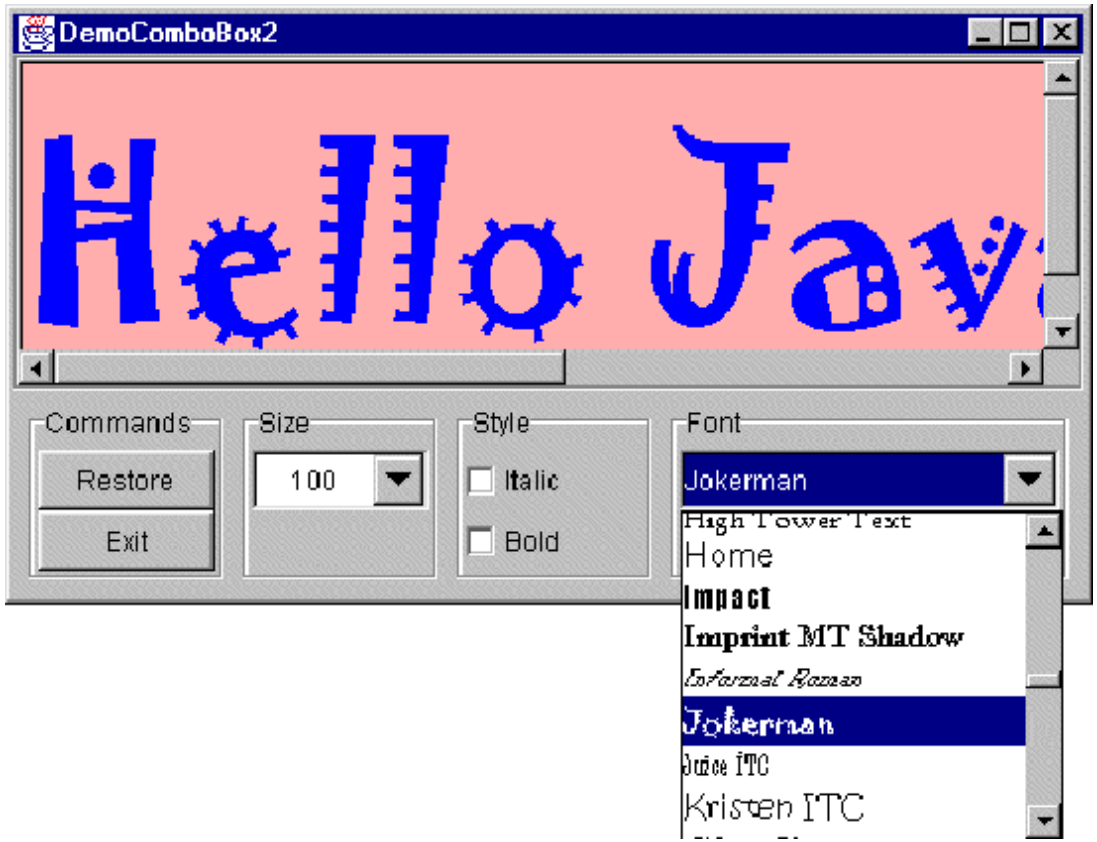
Readability of Text

- Guidelines:
  - Uppercase vs. lowercase
    - WORDS WRITTEN IN BLOCK CAPITALS take longer to read than those written in lowercase
    - However, an important word written in CAPITALS is quickly perceived provided it is surrounded by words in lowercase
  - Serif vs sans serif
    - For printed text, serif fonts are easier to run one's eyes along and take in the written content
    - For on-screen text, serif fonts generally produce less well than sans serif fonts due to poorer resolution of the display

Example Program

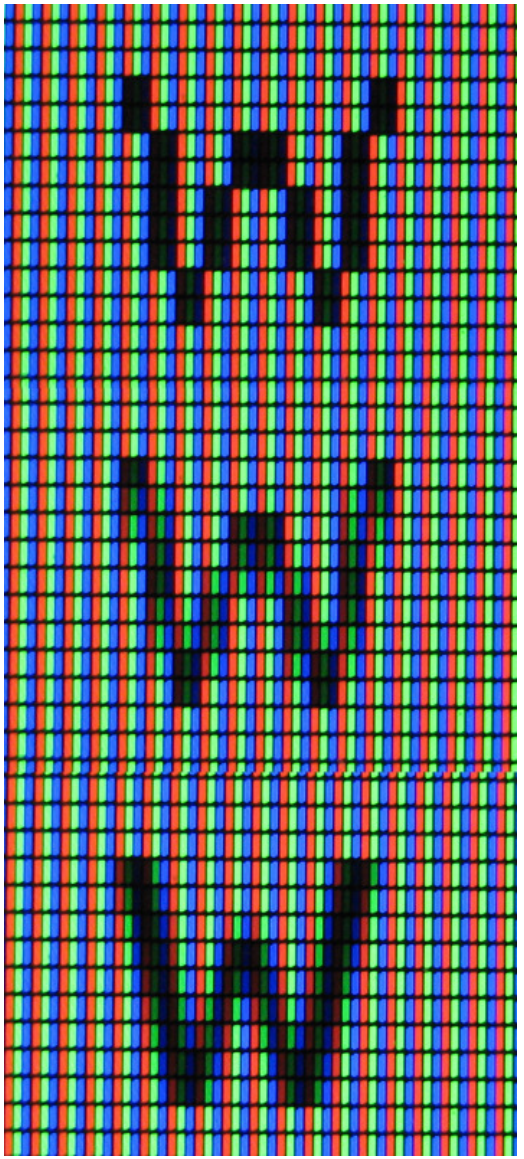
Shown earlier

DemoComboBox2.java



Font Rendering

- Antialiasing (middle)
  - Use smoothing to avoid staircase effect (top)
- Subpixel Rendering (bottom)
  - See next slide



Subpixel Rendering

- Colour displays contain 3 R|G|B sub-pixels for every white pixel
- Idea: when rendering text, treat colour subpixels as if they were 3x smaller grey pixels
  - At small (<0.5 mm) scale the difference in colour is almost unnoticeable
  - Perceived sharpness is improved!

Font Hinting

- Fonts are often outline objects
  - High (infinite in theory) resolution
- Displays have finite resolution
  - Glyph boundaries and pixels do not align
- Idea
  - “snap” the outline to pixel coordinates
  - Lose fidelity, unpredictable hor. dimensions
  - BUT Text is sharper

Hinting Example

Top: no hinting; bottom: with hinting. Source: Wikipedia

OS differences

- OS X
  - Focus on screen appearance matching print appearance as closely as possible
  - Sharpness can be sacrificed somewhat
- Windows
  - On-screen sharpness is more important
  - Looks better on (low-resolution) screens
- Linux
  - Customization is possible
  - Varying success (due to patents on TrueType partly)

Ubuntu example

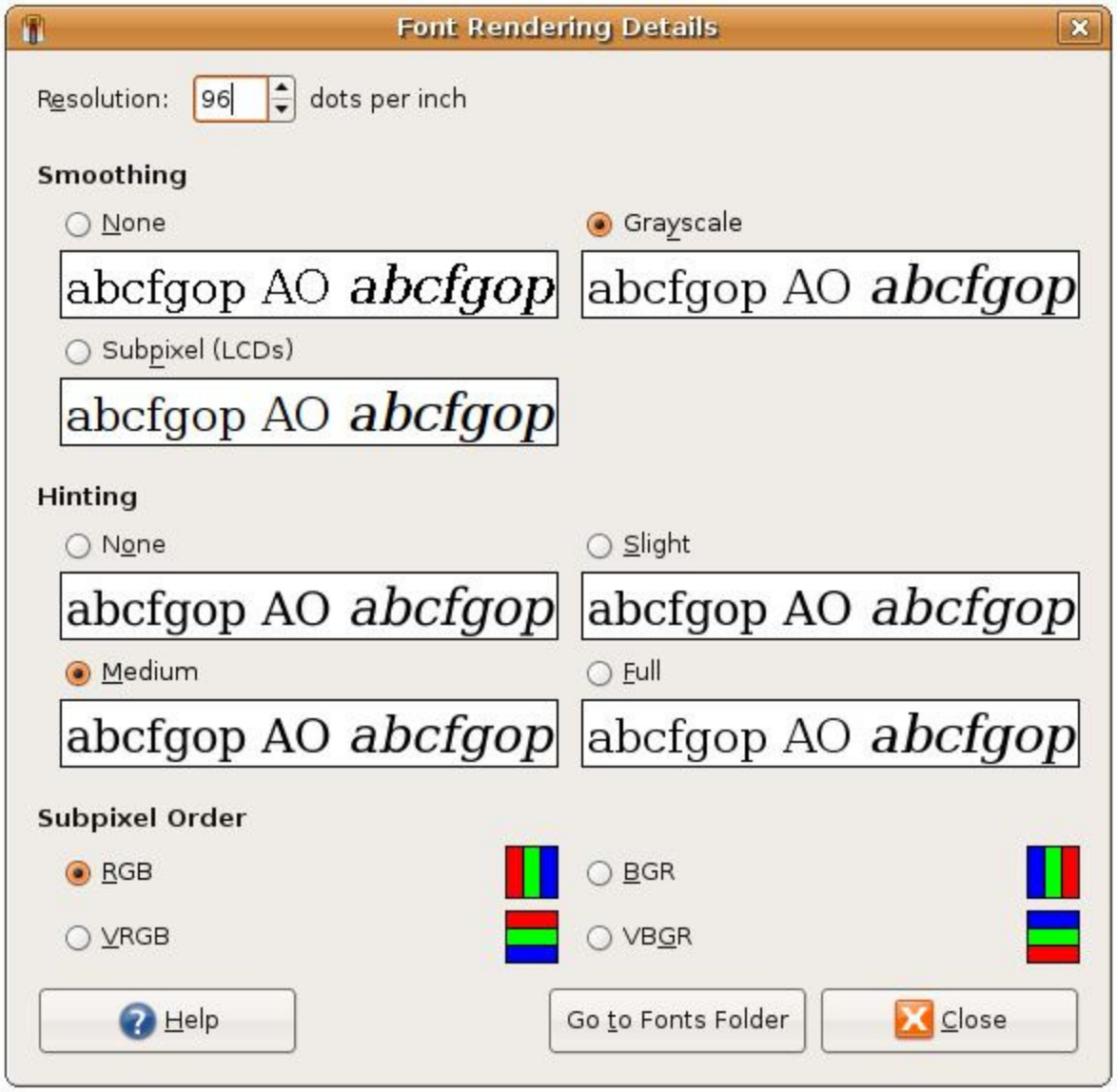


Image File Formats

- There are many, many file formats for storing images
- These include…
  - gif, jpg, png, tiff, bmp

gif

- gif = graphics interchange format
- Pronounced *giff* (with a hard ‘g’)
- Introduced in 1987 by CompuServe Inc.
- Very popular format for web pages
- Features
  - Limited to 256 colors (8-bit)
  - Lossless data compression[a]
  - Compresses best for images with lots of repetition; e.g., flat colors)
  - Support for transparency
    - one colour in the image’s pallete (usually the ‘background’) is treated as transparent
  - Support for animation

[a] decompressed data exactly the same as original data

jpg

- jpg = jpeg = joint photographic experts group
- Pronounced *jay-peg*
- Features
  - Always uses 24-bit color
  - Lossy data compression (up to 95% reduction, but with loss of image quality proportional to amount of compression)

png

- png - Portable Network Graphics
- Pronounced “ping”
- Supports pallete-based colour, greyscale, RGB[A] (24 or 32 bit with alpha)
- Lossless compression
- Non-patented alternative to GIF

tiff

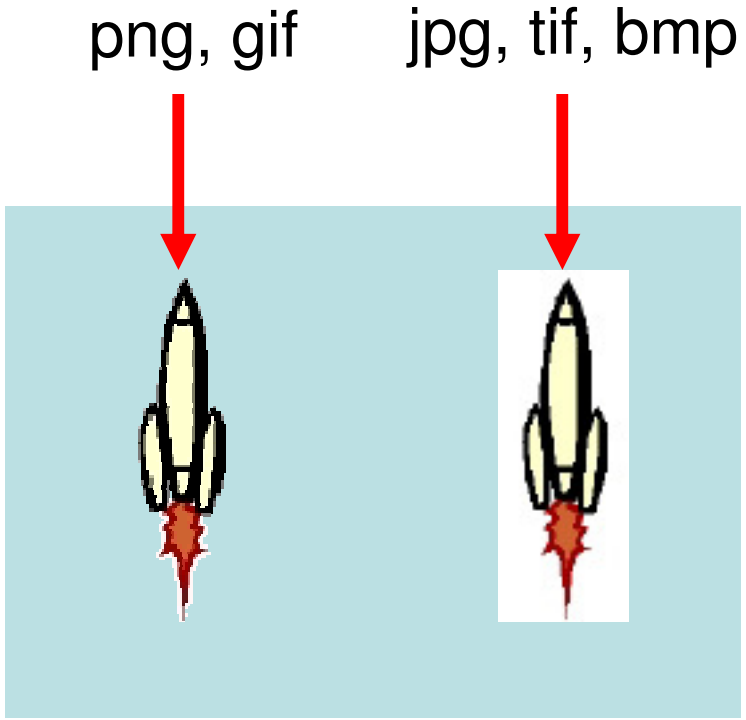
- tif = tiff = tagged image file format
- Features
  - Supports different compression schemes (lossy and lossless)
  - Supports any resolution
  - Black and white, color, or grey shades
  - Widely used in Desktop Publishing

bmp

- bmp = bit-map
- Standard for *Windows* environment
- Uses a pixel map to hold line-by-line raster information
- Features
  - No compression
  - Files are large
  - Bottom to top order of bytes

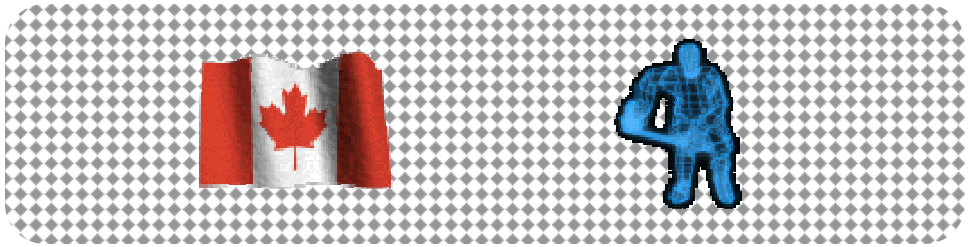
Example (1)

Format	File Size
png	1317
gif	1639
jpg	5522
tif	3328
bmp	7942



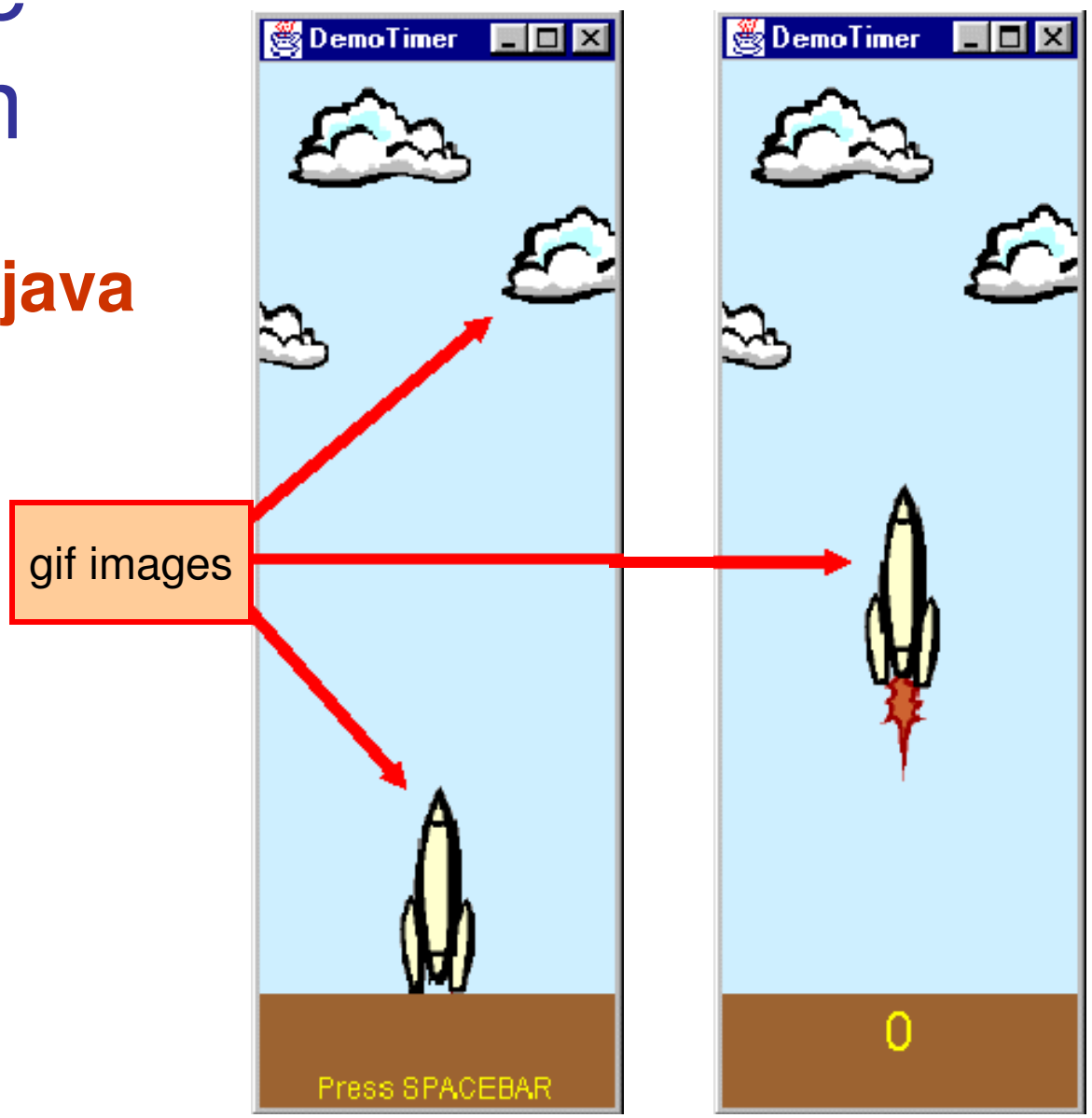
Example (2)

Animated gif
with transparent
background



Example Program

DemoTimer.java



Sound

- Auditory “displays” important too

DemoSound.java