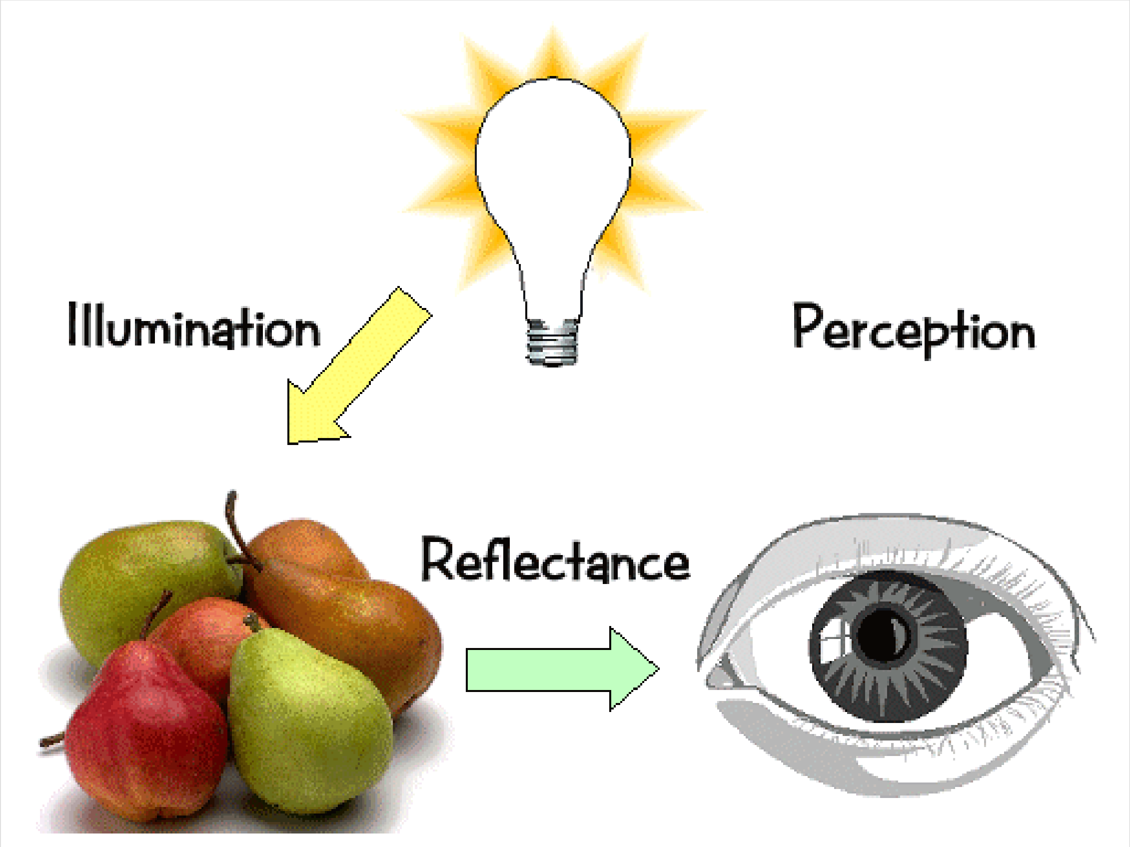
# Colors, Paint, Etc

# Color

- To understand how to make realistic images, we need a basic understanding of the physics and physiology of vision.

# Basics Of Color

- Elements of color:

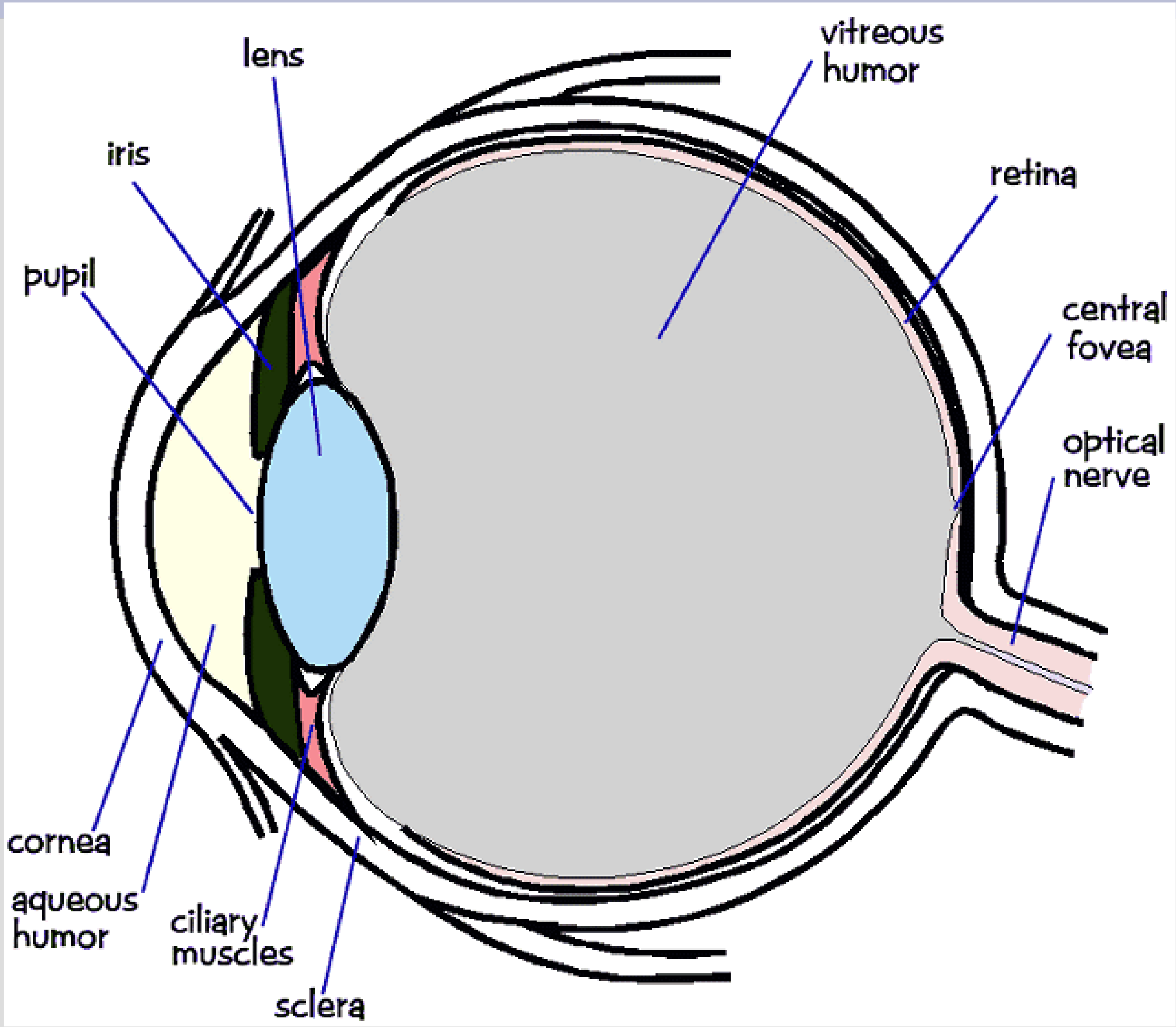


# Basics of Color

- Physics:
  - Illumination
    - Electromagnetic spectra
  - Reflection
    - Material properties
    - Surface geometry and microgeometry (i.e., polished versus matte versus brushed)
- Perception
  - Physiology and neurophysiology
  - Perceptual psychology

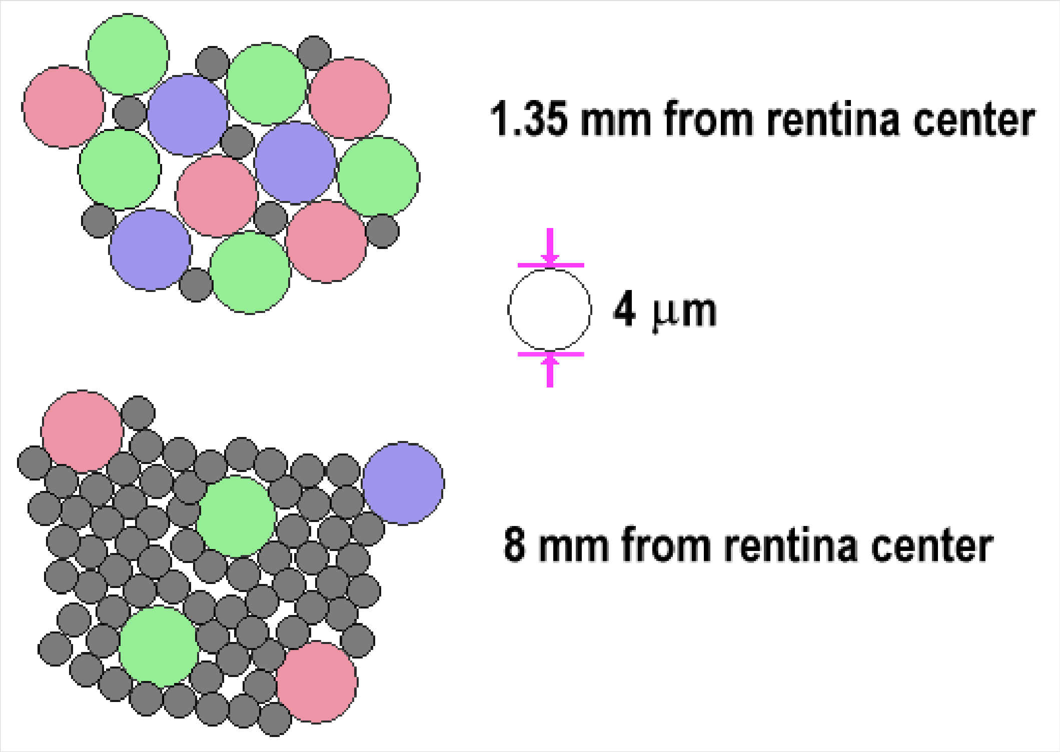
# Physiology of Vision

- The eye:
- The retina
  - Rods
  - Cones
    - Color!

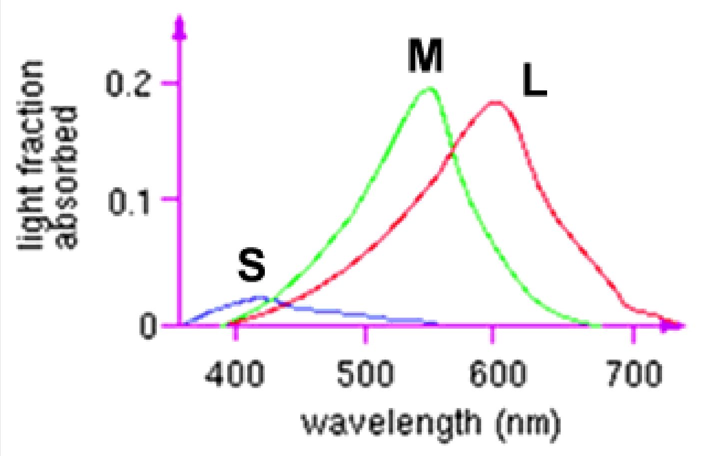


# Physiology of Vision

- The center of the retina is a densely packed region called the *fovea*.
  - Cones much denser here than the *periphery*



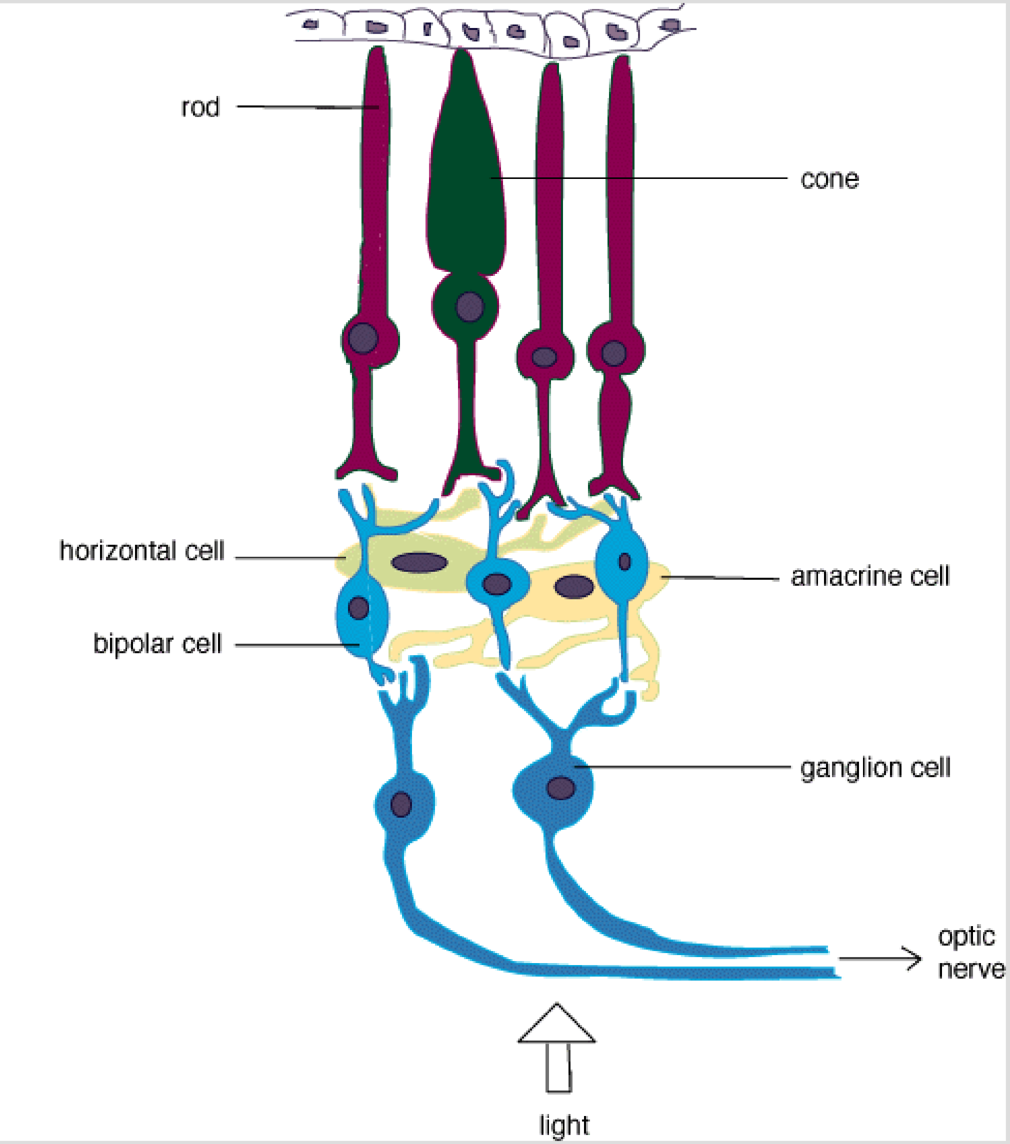
# Physiology of Vision: Cones

- Three types of cones:
  - **L** or **R**, most sensitive to red light (610 nm)
  - **M** or **G**, most sensitive to green light (560 nm)
  - **S** or **B**, most sensitive to blue light (430 nm)
  - 

  - Color blindness results from missing cone type(s)

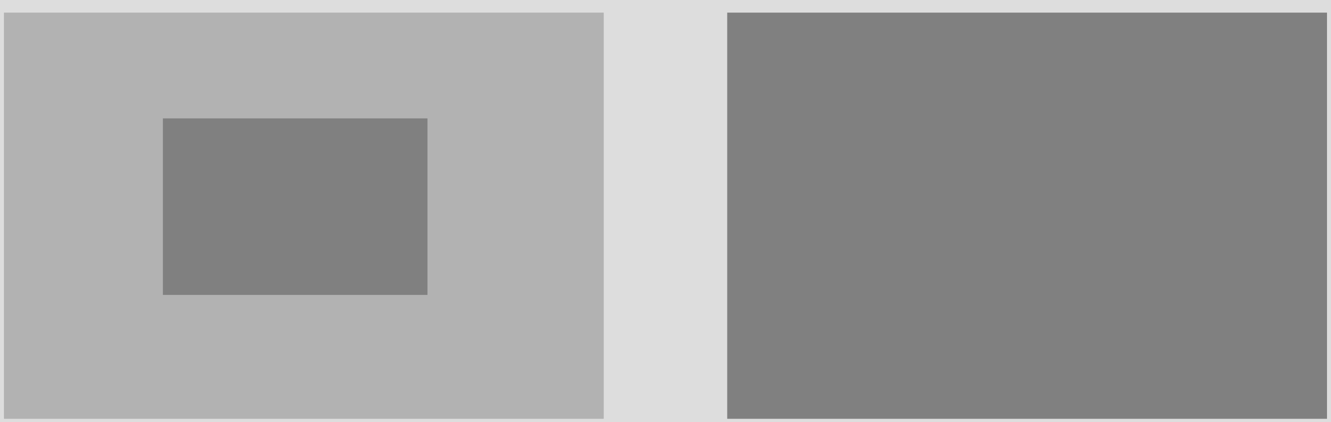
# Physiology of Vision: The Retina

- Strangely, rods and cones are at the *back* of the retina, behind a mostly-transparent neural structure that collects their response.
- http://www.trueorigin.org/retina.asp



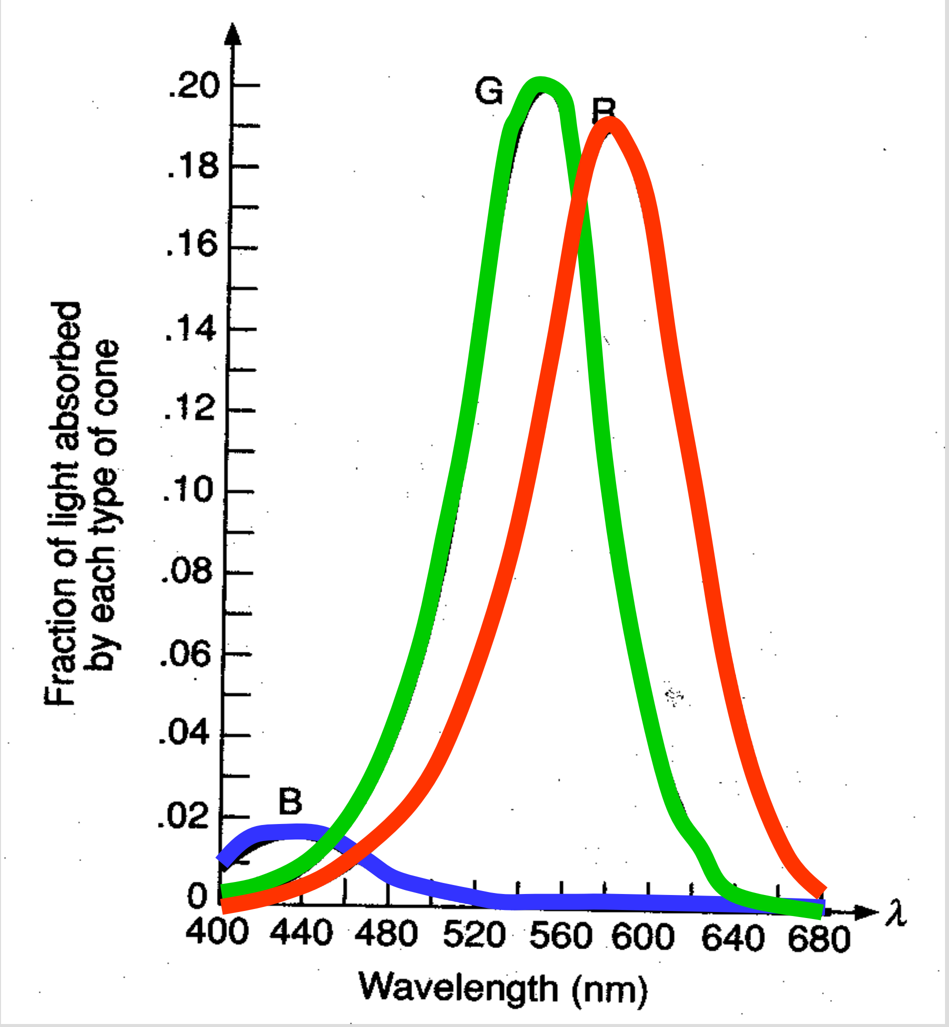
# Perception: Other Gotchas

- Color perception is also difficult because:
  - It varies from person to person
  - It is affected by adaptation (stare at a light bulb… don't)
  - It is affected by surrounding color:



# Human Color Vision

- Humans have 3 light sensitive pigments in their cones, called L, M, and S
- Each has a different spectral response curve:




$$L = \int L(\lambda) E(\lambda) d\lambda$$

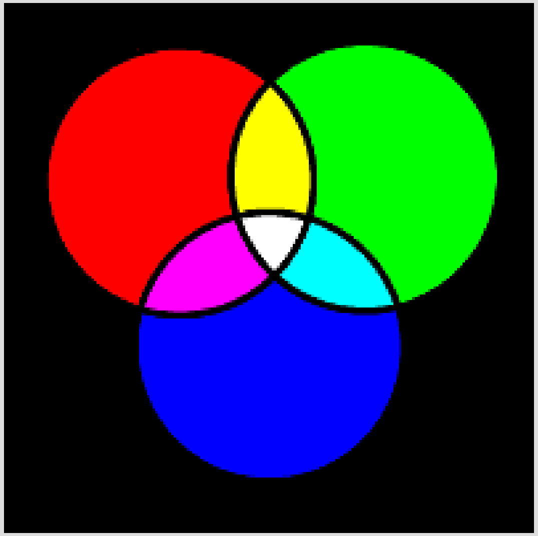
$$M = \int M(\lambda) E(\lambda) d\lambda$$

$$S = \int S(\lambda) E(\lambda) d\lambda$$

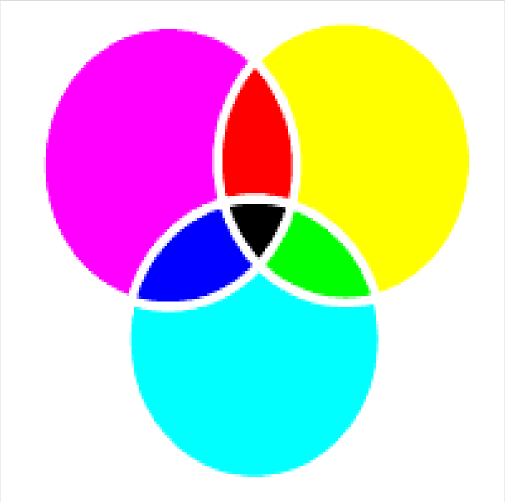
- This leads to metamerism
- “Tristimulus” color theory

# Combining Colors

Additive (RGB)
Shining colored lights
on a white ball



Subtractive (CMYK)
Mixing paint colors and
illuminating with white light



# Color Space

- CIEXYZ
- RGB
- CYMK
- sRGB

$$c = r \cdot p_r + g \cdot p_g + b \cdot p_b$$

Source

Run

# Color Space

- RGB
  - Red, Green, Blue
  - Color space of most monitors
- CMYK
  - Cyan, Magenta, Yellow, Black
  - Color space for printers
- Both are device dependent
- Cannot represent all visible colors with **positive** coefficients

# Color Space

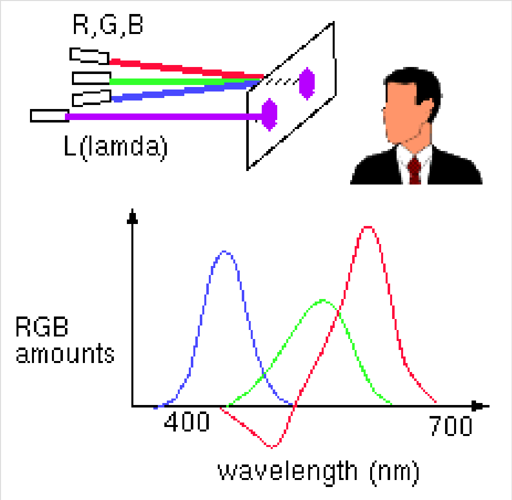
- CIEXYZ
  - A color standard
  - Three primary colors X, Y, and Z
  - Any visible color can be represented as a positive combination of X, Y, and Z
  - Really difficult to implement on physical devices
- Device-dependent

# Color Space

- **sRGB:** standard RGB
  - Device independent color space
  - Proposed standard color space
  - Red, green, and blue components
  - An alpha component that specifies transparency

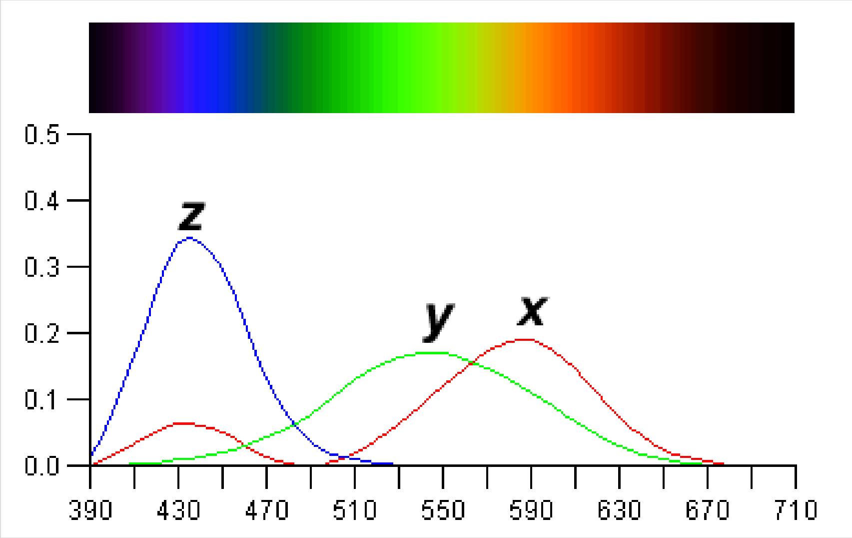
# Color Spaces

- Three types of cones suggests color is a 3D quantity.  How to define 3D color space?
- Idea: shine given wavelength ($\lambda$) on a screen, and mix three other wavelengths (R,G,B) on same screen.  Have user adjust intensity of RGB until colors are identical:



# CIE Color Space

- The *CIE* (Commission Internationale d'Eclairage) came up with three hypothetical lights X, Y, and Z with these spectra:



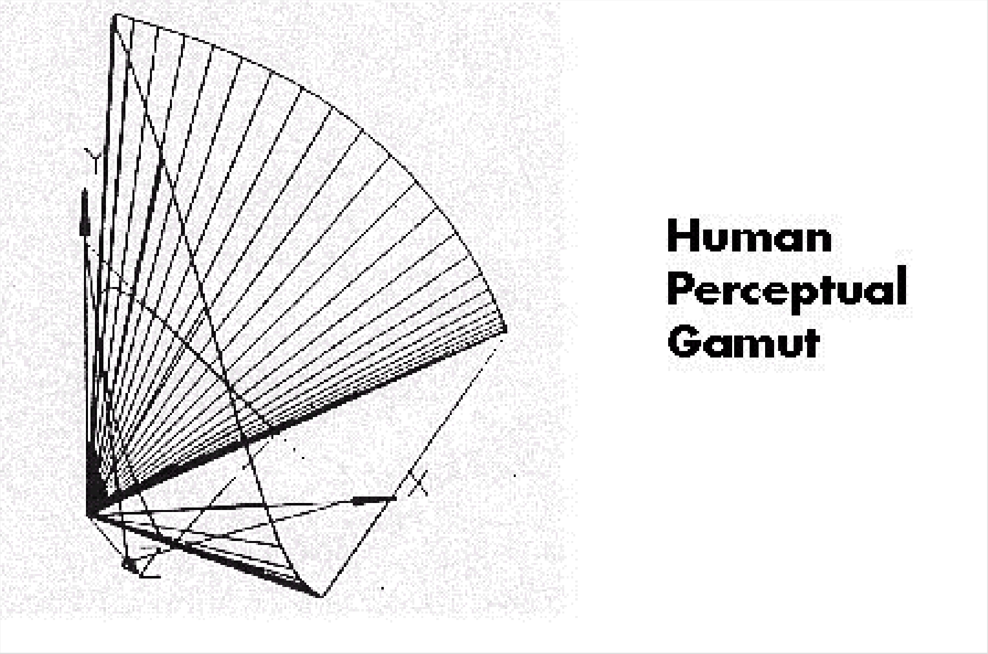

Note that:

X ~ R
Y ~ G
Z ~ B

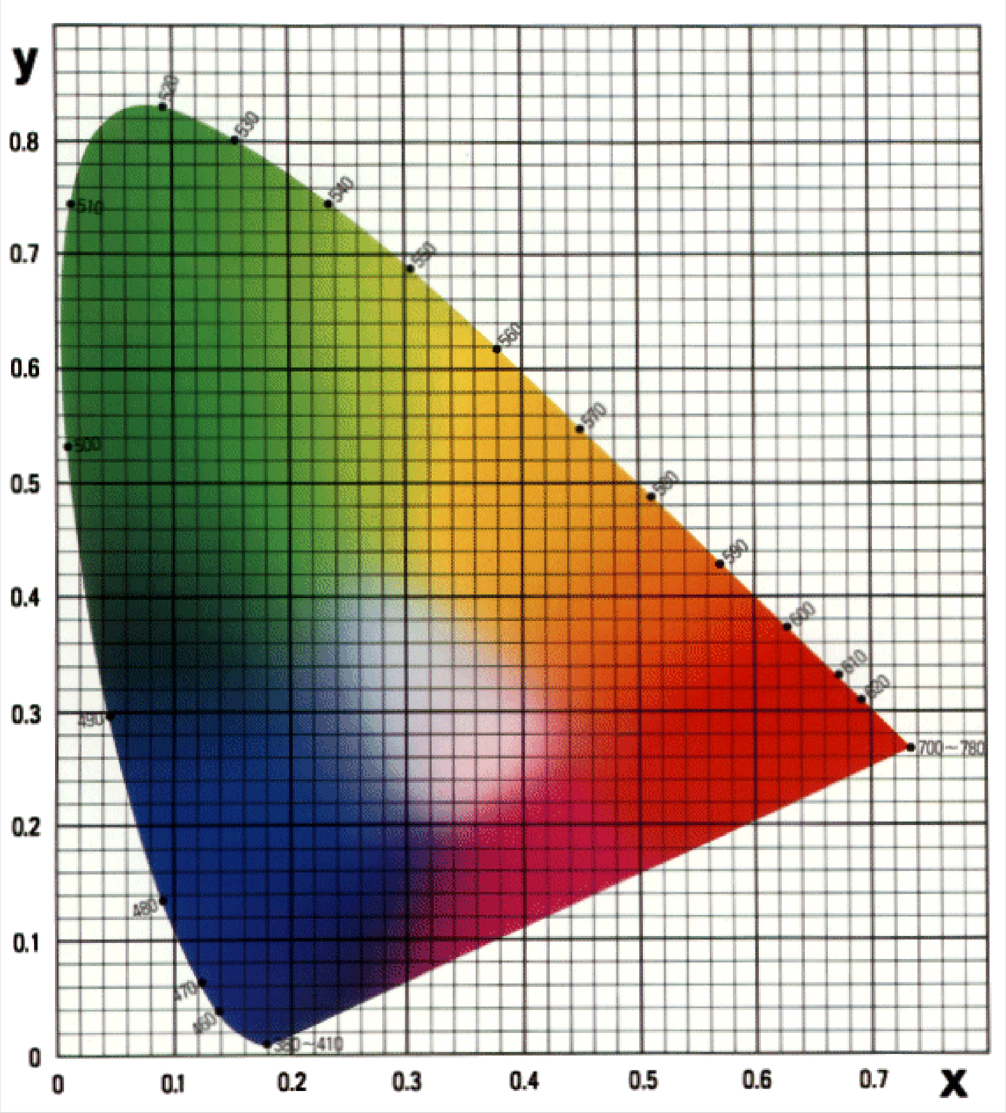
- Idea: any wavelength λ can be matched perceptually by *positive* combinations of X,Y,Z

# CIE Color Space

- The *gamut* of all colors perceivable is thus a three-dimensional shape in X,Y,Z
- Color = X'*X* + Y'*Y* + Z'*Z*



# CIE Chromaticity Diagram (1931)




For simplicity, we often project to the 2D plane x+y+z=1

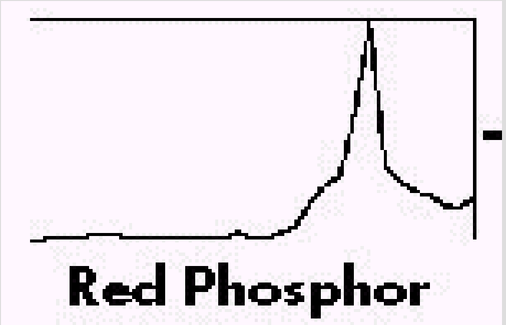
$x = X / (X+Y+Z)$

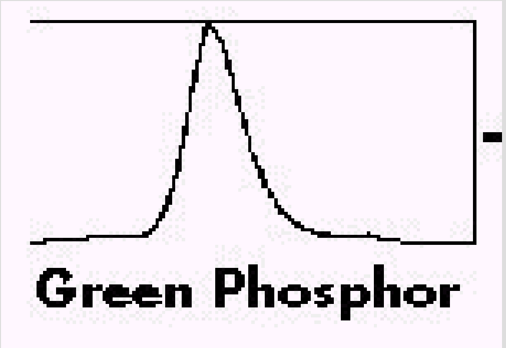
$y = Y / (X+Y+Z)$

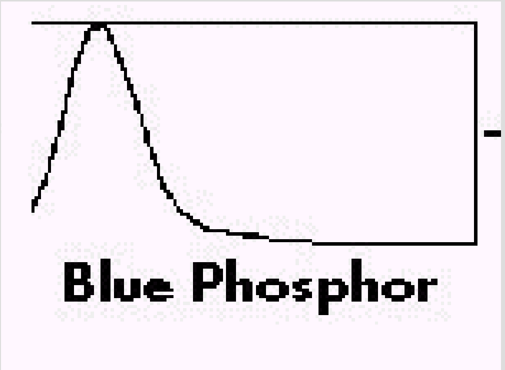
$z = 1 – x - y$

# Device Color Gamuts

- Since X, Y, and Z are hypothetical light sources, no real device can produce the entire gamut of perceivable color
- Example: CRT monitor

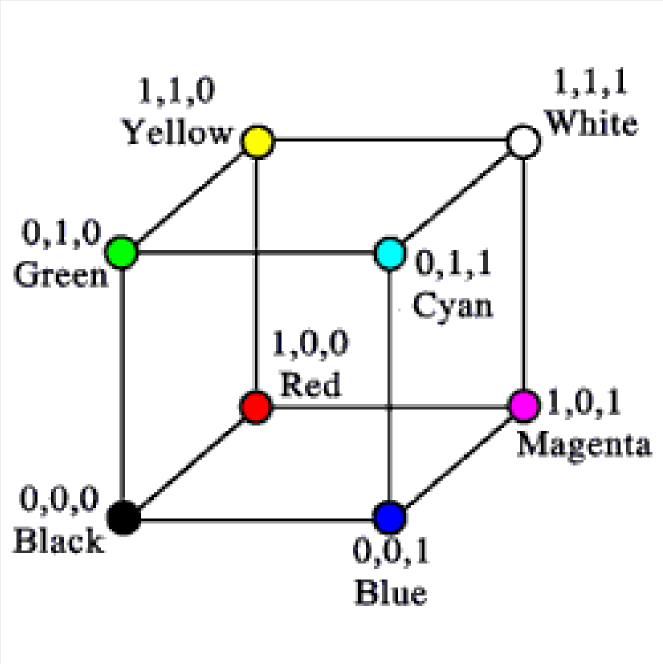


# RGB Color Space

- Define colors with (r, g, b) amounts of red, green, and blue



# Converting Color Spaces

Simple matrix operation:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The transformation $C_2 = M^{-1}_2 \, M_1 \, C_1$ yields RGB on monitor 2 that is equivalent to a given RGB on monitor 1

# YIQ Color Space

YIQ is the color model used for color TV in America.  Y is brightness, I (orange-cyan) & Q (green-magenta) are color

- Note: *Y* is the same as CIE's *Y*
- Result: Use the Y alone and backwards compatibility with B/W TV!
- These days when you convert RGB image to B/W image, the green and blue components are thrown away and red is used to control shades of grey (usually)

# Converting Color Spaces

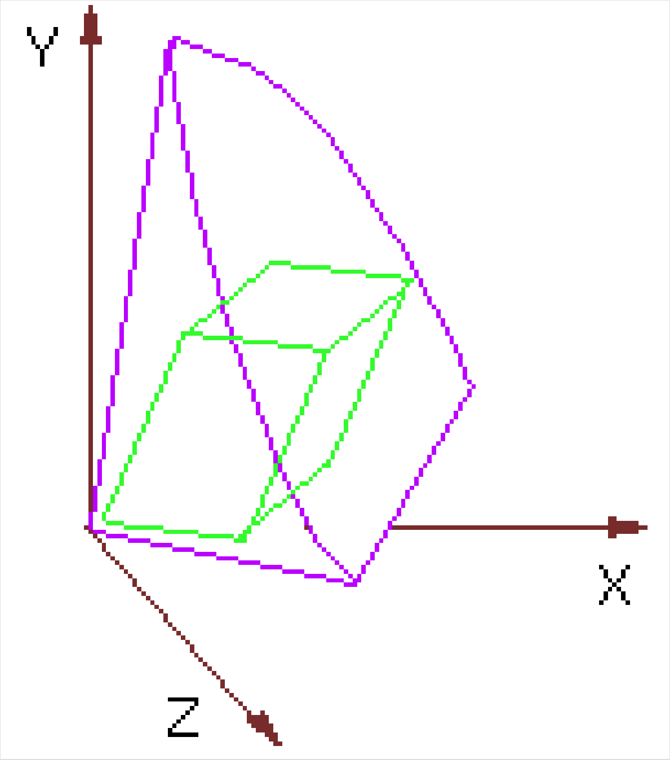
Converting between color models can also be expressed as such a matrix transform:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Note the relative unimportance of blue in computing the Y

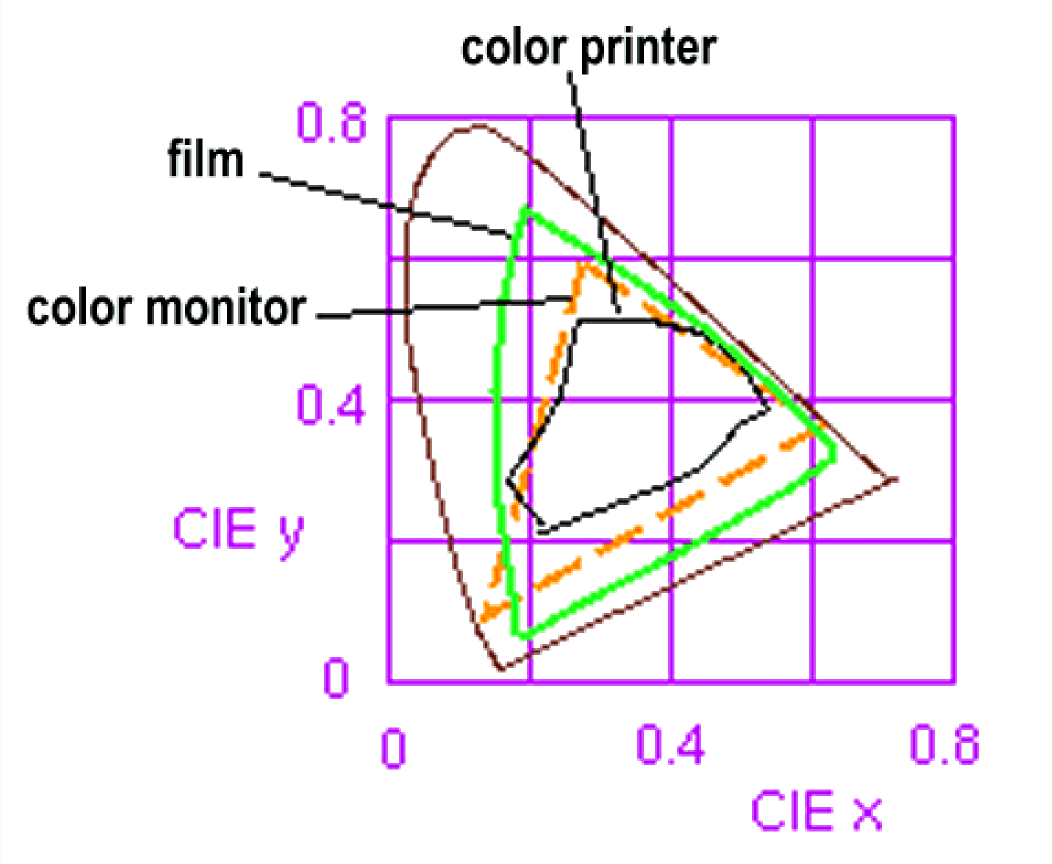
# Device Color Gamuts

- The RGB color cube sits within CIE color space something like this:



# Device Color Gamuts

- We can use the CIE chromaticity diagram to compare the gamuts of various devices:
- Note, for example, that a color printer cannot reproduce all shades available on a color monitor

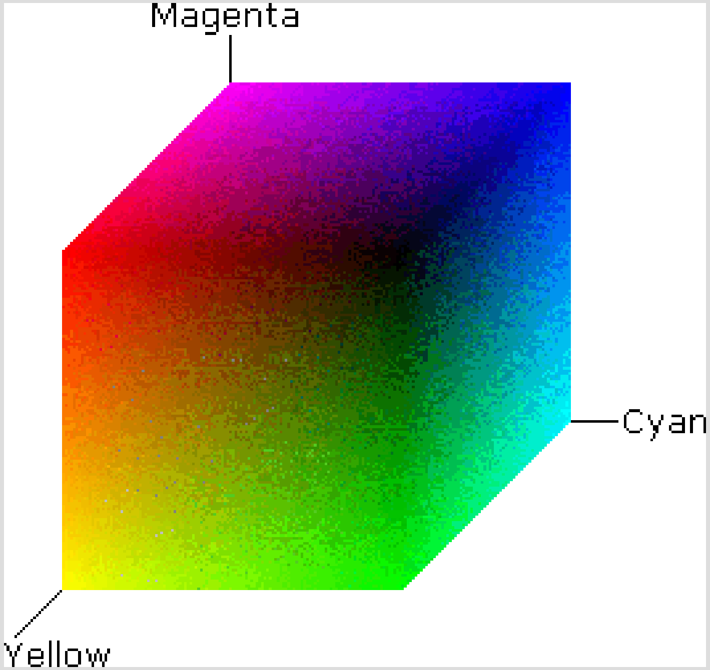


# CMY(K): printing

- Cyan, Magenta, Yellow (Black) – CMY(K)
- A subtractive color model

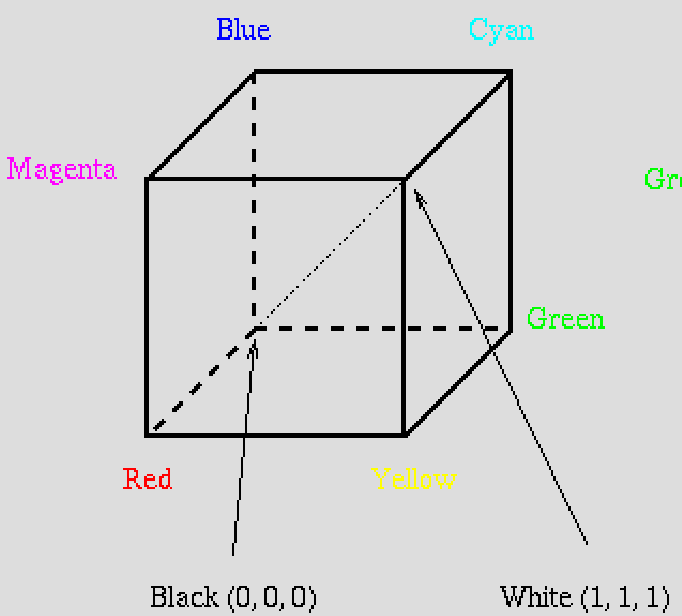
| *dye color* | *absorbs* | *reflects* |
|---|---|---|
| cyan | red | blue and green |
| magenta | green | blue and red |
| yellow | blue | red and green |
| black | all | none |

# CMYK Space

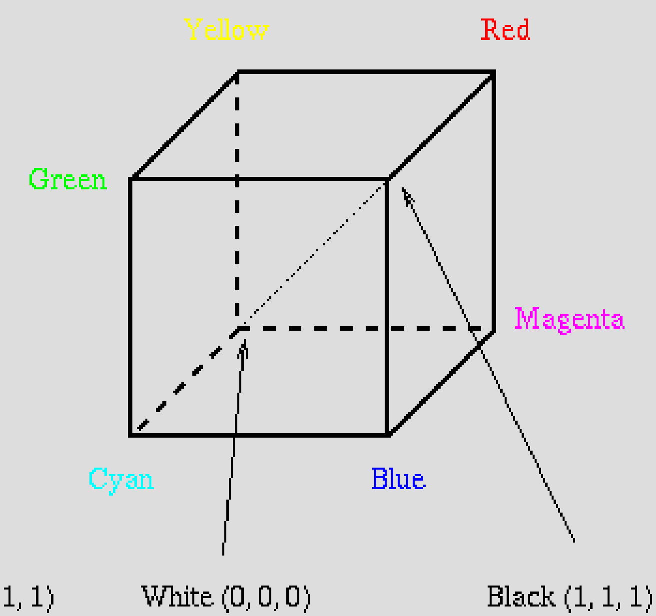


# RGB and CMY

- Converting between RGB and CMY




The RGB Cube

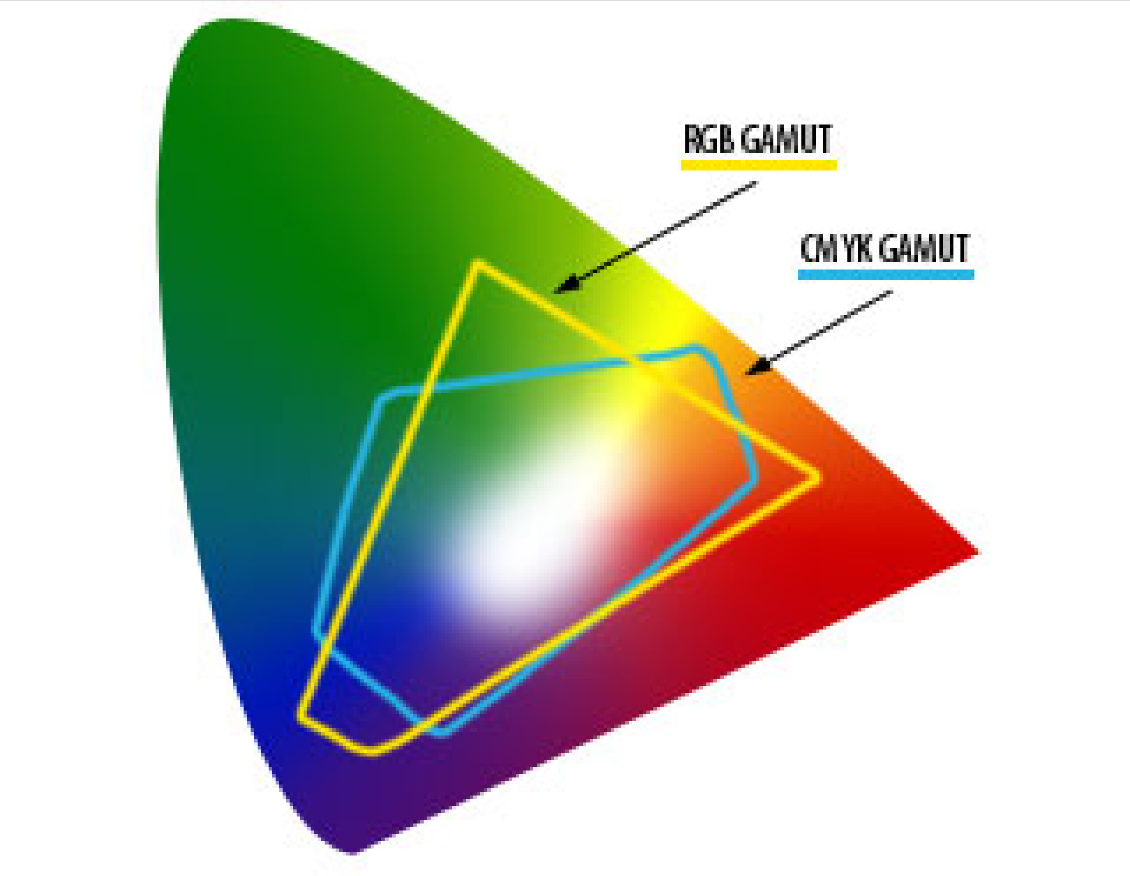



The CMY Cube

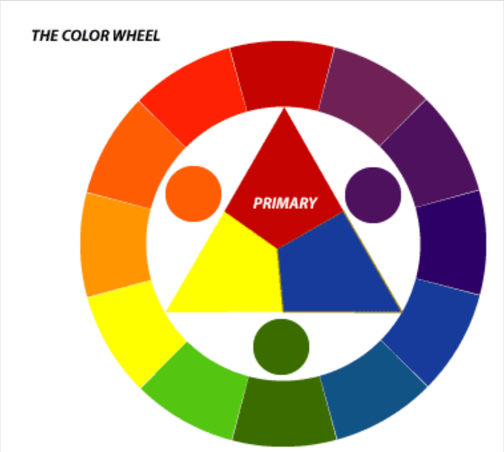
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

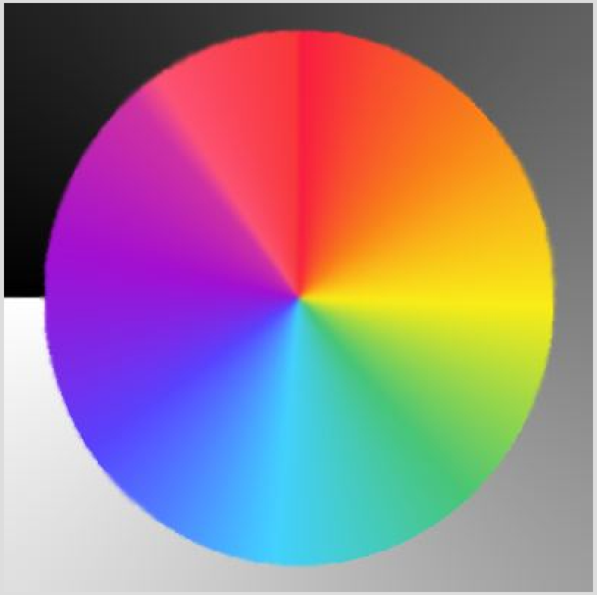
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

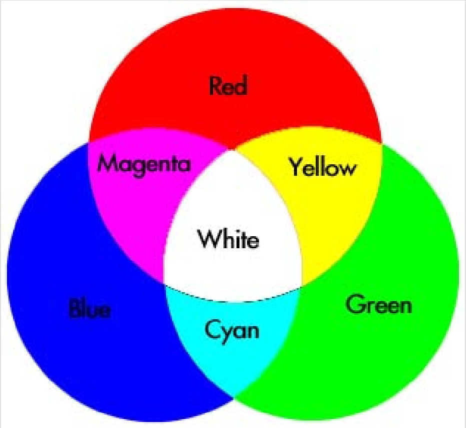
# RGB and CMY



# Color Wheels

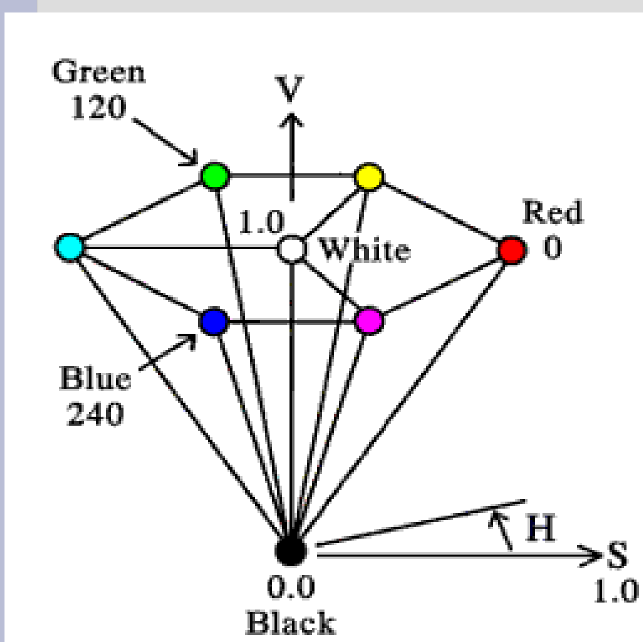


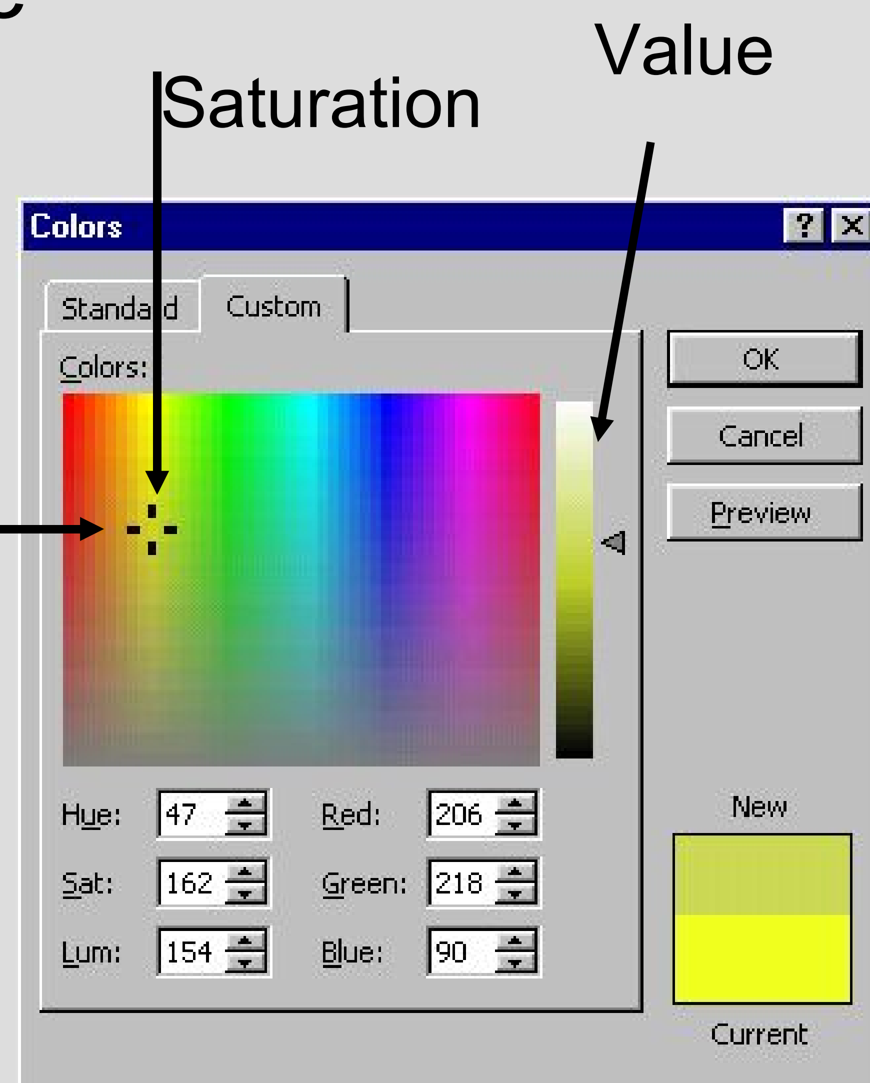





# HSV Color Space

- A more intuitive color space
  - H = Hue
  - S = Saturation
  - V = Value (or brightness)



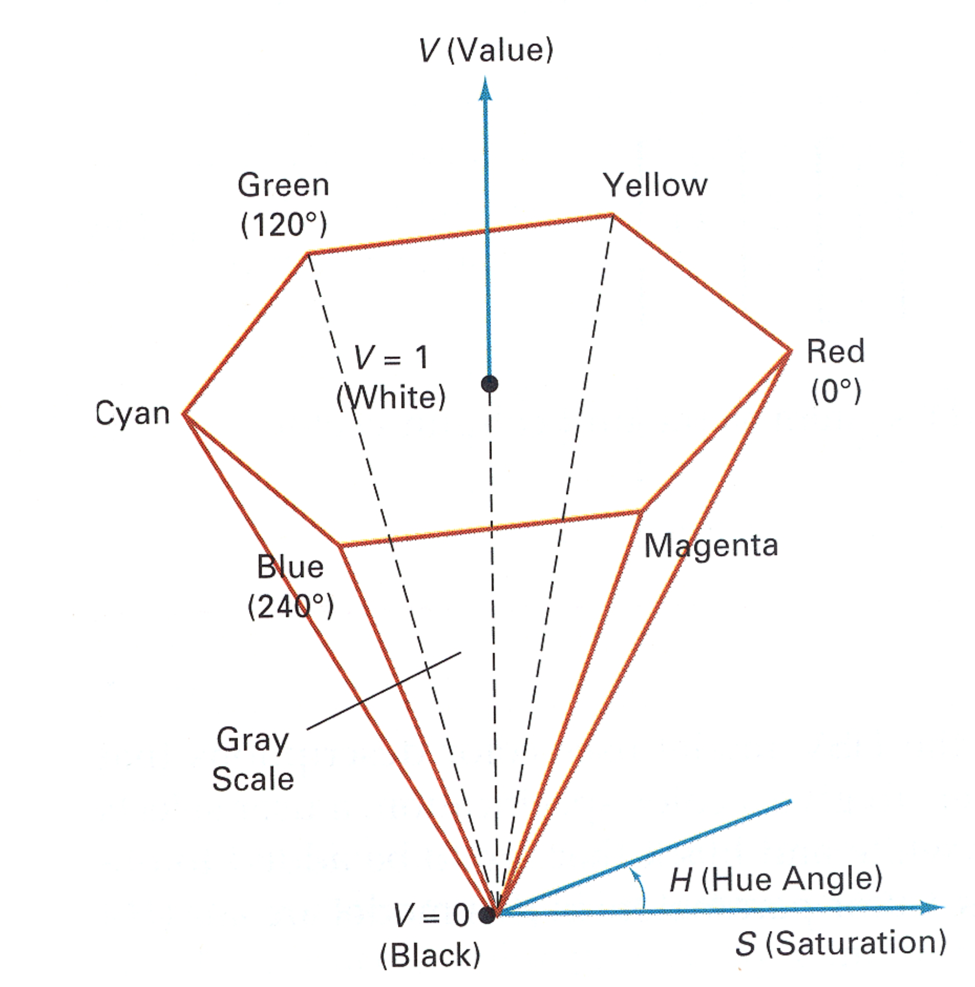



# HSV Color Space

Computer scientists frequently use an intuitive color space that corresponds to tint, shade, and tone:

- Hue - The color we see (red, green, purple)
- Saturation - How far is the color from gray (pink is less saturated than red, sky blue is less saturated than royal blue)
- Brightness (Luminance) - How bright is the color (how bright are the lights illuminating the object?)

# HSV Color Model

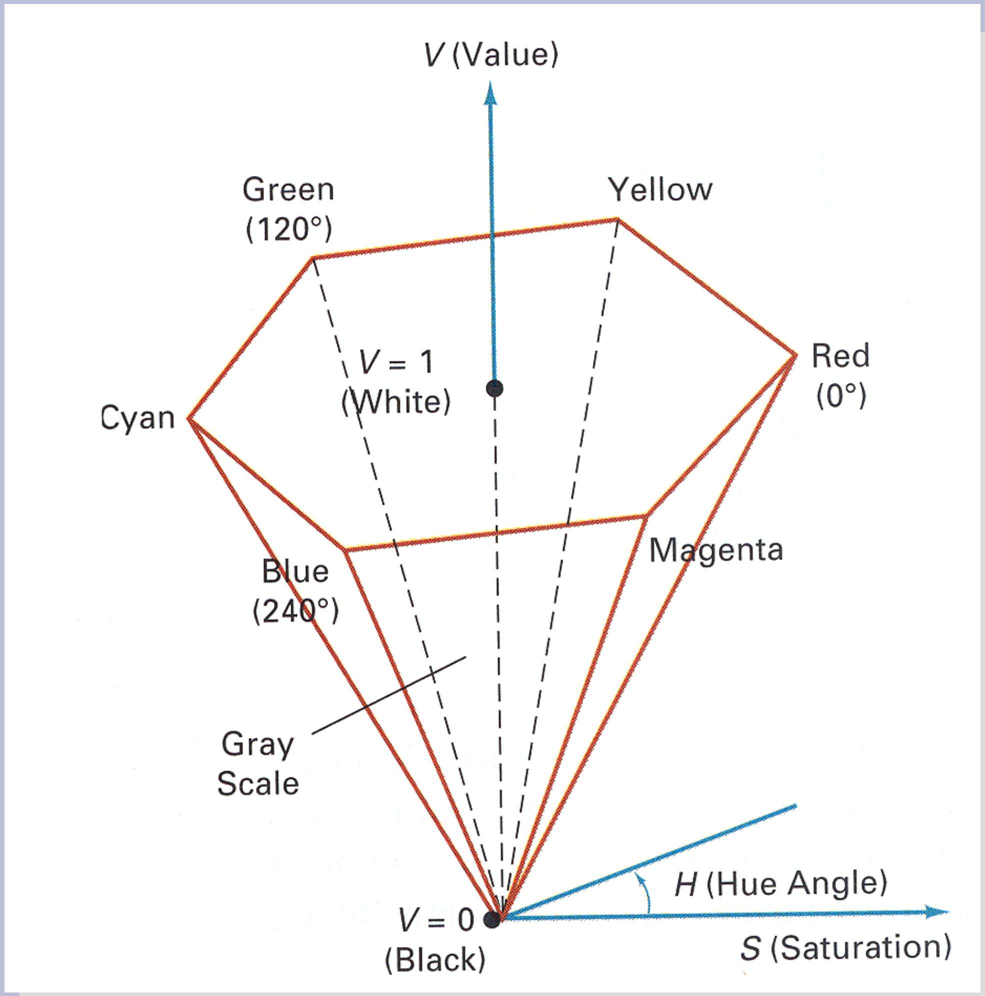



Hue (H) is the angle around the vertical axis

Saturation (S) is a value from 0 to 1 indicating how far from the vertical axis the color lies

Value (V) is the height of the hexcone”

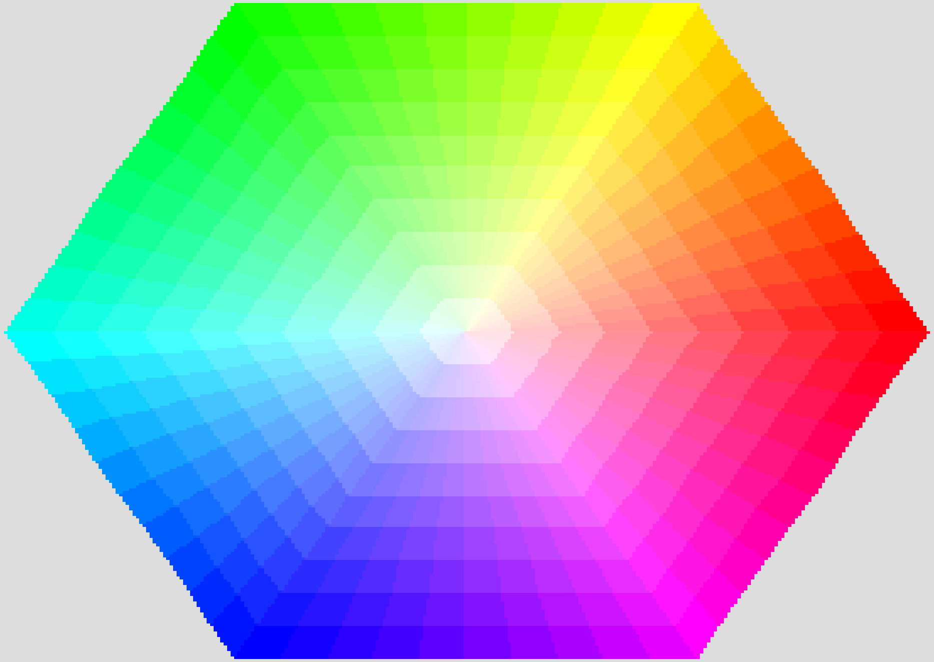
# HSV Color Model




| H | S | V | Color |
|---|---|---|---|
| 0 | 1.0 | 1.0 | Red |
| 120 | 1.0 | 1.0 | Green |
| 240 | 1.0 | 1.0 | Blue |
| * | 0.0 | 1.0 | White |
| * | 0.0 | 0.5 | Gray |
| * | * | 0.0 | Black |
| 60 | 1.0 | 1.0 | ? |
| 270 | 0.5 | 1.0 | ? |
| 270 | 0.0 | 0.7 | ? |

# Intuitive Color Spaces

A top-down view of hexcone

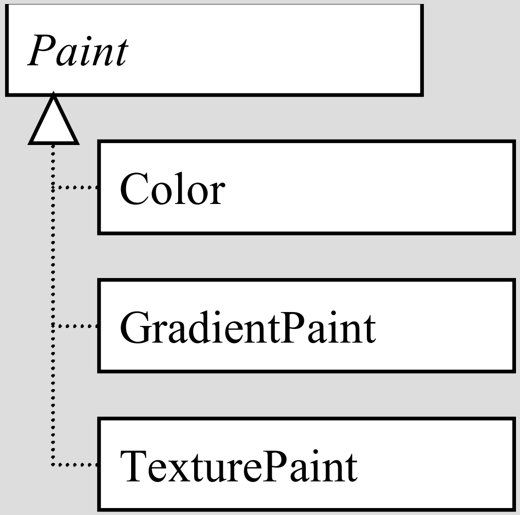


# Paint

Generalizing the concept of color, Java 2D drawing applies an attribute called paint

Source

Run



# Strokes

- Width
- End style
- Join style
- Miter limit
- Dash pattern

Source

Run