Color

Elements of Graphics
CS324e
Spring 2019
Color Models

- Final color derived from combination of light sources
- Additive color models add light sources
- Subtractive color models subtract light sources

© Genevieve B. Orr (Willamette University)
Additive vs Subtractive

- Painting is subtractive (white surface)
- Computer monitors are additive (black surface)
Digital Color

- Each pixel has three light elements:
  - Red
  - Green
  - Blue

- Light element intensity range from 0 to 255
  - 0 means color is transparent (least intensity)
  - 255 means color is fully opaque (highest intensity)
RGB

- Red: (255, 0, 0)
- Green: (0, 255, 0)
- Blue: (0, 0, 255)

- Colors at full opacity tend to be a little garish!
- Processing includes a color selector for more intuitive color selection if you don’t have access to digital paint program
Hexadecimal

- Color notation useful for HTML and CSS
- RGB color (0 - 255) encoded as a two-digit base 16 value

Examples:
- #000000 ➝ (0, 0, 0)
- #FFFFFF ➝ (255, 255, 255)
- #6699CC ➝ (102, 153, 204)
Color Depth

- 1 bit can represent 2 values \((2^1)\)
- 2 bits can represent 4 values \((2^2)\)
- 4 bits can represent 16 values \((2^4)\)
How many color values can 8-bits represent?

Answer: 256 (2^8)
True Color

- Supports three 8-bit channels (RGB)
  - RGBA adds a fourth channel for alpha (transparency)
- RGB supports 24-bits total or 16,777,216 values ($2^{24}$)
- The human eye can discriminate around 10M colors
Note that any RGB model is limited to colors within the RGB gamut — such models cannot represent all human-visible colors!
Image Formats

❖ GIF
  ❖ Color depth: 1-bit to 8-bit
  ❖ Transparency: 1-bit

❖ JPEG
  ❖ Color depth: 24-bit
  ❖ Transparency: None

❖ PNG
  ❖ Color depth: 1-bit to 24-bit
  ❖ Transparency: 8-bit
Setting Color in Processing

- `background(int red, int green, int blue)` sets the color of the window in terms of RGB.
- `fill(int red, int green, int blue)` sets the color for any subsequent shape primitives.
- `fill(int red, int green, int blue, int alpha)` includes a transparency channel to modify opacity.
Using the color Primitive

- Processing has a special primitive for color:
  - `color(float red, float green, float blue);`
- Can be used in `fill`, `stroke`, `background` functions
Consider...

color c = color(255.0, 255.0, 0.0);
fill(c);
rect(0, 0, 200, 200);
Transparency and Blending

- Transparency (alpha channel) also ranges from 0 to 255
- Transparency allows for on-screen color mixing based on the blend mode
- Default blend mode is BLEND
  - `blendMode(BLEND)`
ADD
Additive blending with maximum value of white:
C = min(A*factor + B, 255)

SUBTRACT
Subtractive blending with minimum value of black:
C = max(B - A*factor, 0)

LIGHTEST
The lightest color is used:
C = max(A*factor, B)

DARKEST
The darkest color is used:
C = min(A*factor, B)

MULTIPLY
Multiply the colors, result will always be darker:
C = A * B

A is source image
B is destination image
Factor is source alpha
Color Theory

❖ The study of color interactions
❖ Color classification
❖ Color mixing
❖ Color design
❖ Cultural context

RYB color model: primary, secondary, tertiary
Warm vs Cool

- Warm colors (yellow, orange, red, tan)
  - Active
  - Advancing
  - Stimulating
- Cool colors (green, blue, violet, gray)
  - Passive
  - Receding
  - Relaxing
Achromatic and Monochromatic

- Achromatic colors schemes are neutral (white, black, gray)
- Unsaturated colors are near neutral (tans, pastels)
- Monochromatic schemes focus on value using a single hue
Complementary schemes use colors on opposite ends of the color wheel
- High contrast
- Dramatic
- Forces eye movement

Van Gogh
Split Complementary

- Split complementary schemes use a color and the color adjacent to its complement
  - Subtle contrast
  - Balanced tension

Vermeer
Analogous

- Analogous schemes use adjacent primary, secondary or tertiary colors
- Harmonious
- Moody

Monet
Triadic schemes use three equidistant colors along the wheel

- Balanced
- Vibrant

Rubens
Tetradic and Square

Tetradic color scheme

Square color scheme

(from Tiger Color)
Hands-on: Using Color

❖ Today’s activities:

1. Use Processing’s color picker to incorporate multiple colors via \texttt{fill} and \texttt{stroke}
2. Store \texttt{color} primitives in an array for reuse
3. Use \texttt{blendMode} to affect color interactions
4. Create an image using one of the color theory schemas listed after this slide: achromatic, monochromatic, complementary, split complementary, analogous, color triad, tetrad, or square