Fundamentals of Computer Science I: Media Computing (CS151.02 2007F)

Laboratory: RGB Colors

Summary: In this laboratory, you will begin to explore various aspects of the RGB representation of colors.

Contents:

- Preparation
- Exercises
    - Exercise 1: Primaries
    - Exercise 2: Secondaries
    - Exercise 3: Reduced Components
    - Exercise 4: Common Components
    - Exercise 5: Your Favorite Colors
    - Exercise 6: Nearby Colors
    - Exercise 7: Psuedo-Complementary Colors
    - Exercise 8: Psuedo-Complementary Colors, Revisited
    - Exercise 9: Generating Colors
- Explorations
    - Exploration 1: Color Context
    - Exploration 2: Related Colors
- Notes
    - Preparation
    - Notes on Exercise 4: Common Components
    - Notes on Exercise 8: Psuedo-Complementary Colors, Revisited

Reference:

- `(rgb.new r g b)` - make an RGB color with the given components

- `(rgb.blue color)` - extract the blue component of an RGB color
- `(rgb.green color)` - extract the green component of an RGB color
- `(rgb.red color)` - extract the red component of an RGB color

- `(rgb->cname color)` - given a color obtained from an image, find a common name for that color
- `(rgb->string color)` - convert an RGB color to a more human readable representation
- `(rgb->rgb-list color)` - convert an RGB color to a more human readable representation

Preparation

Plan to do your work for this laboratory in the interactions window, rather than in the definitions window.

In this laboratory, it will help to work with a blank canvas, so create a new 11x9 image, call it `canvas`, show it, and zoom in to 16:1 resolution. If you've forgotten how to do all of that, see the notes at the end of this document.

Exercises

Exercise 1: Primaries

The primary colors are the colors in which one of the three components has the full value (255) and the other two components are 0.

a. What colors do you expect to get from `(rgb.new 255 0 0)`, `(rgb.new 0 255 0)`, and `(rgb.new 0 0 255)`?

b. Explore your answers by setting various pixels in `canvas` to the specified colors. For example, you might set the top-left pixel to the first color with

```
(image.set-pixel! canvas 0 0 (rgb.new 255 0 0))
```

c. Explore your answers by using `rgb->cname` to find the names associated with these colors.

Exercise 2: Secondaries

The secondary colors are those you get when you set two components to 255 and the third to 0.

a. What colors do you expect to get from `(rgb.new 255 255 0)`, `(rgb.new 255 0 255)`, and `(rgb.new 0 255 255)`?

b. Explore your answers by setting various pixels in `canvas` to the specified colors.

c. Explore your answers by using `rgb->cname` to find the names associated with these colors.

Exercise 3: Reduced Components

In the previous two experiments, we set the level of components to 255 or to 0. Let's consider what happens if the components are less than 255.

a. What colors do you expect to get from `(rgb.new 192 0 192)`, `(rgb.new 128 0 128)`, `(rgb.new 64 0 64)`, `(rgb.new 192 0 64)`, and `(rgb.new 64 0 192)`?

b. Confirm your predictions by setting various pixels in `canvas` to the specified colors.

c. Determine what names DrFu associates with these colors. Are they good names?

Exercise 4: Common Components

a. What do you expect to have happen if all three components of a color are 128? If all three components are 255? If all three components are 0?

b. Check your answers by setting a pixel to each color and by finding DrFu's name for that color.

c. Explain why those colors correspond to the different combinations. If you're not sure, check the notes on this problem.

Exercise 5: Your Favorite Colors

In the lab on raster graphics, you defined three favorite colors. Your definitions looked something like the following:

```
(define fave1 (cname->rgb "blood red"))
(define fave2 (cname->rgb "aquamarine"))
(define fave3 (cname->rgb "wheat"))
```

If all went well, you also saved those definitions in the file
/home/*username*/Desktop/fave-colors.sct. If so, try loading that file. If not, try creating that file now and then load it.

```
(load "/home/username/Desktop/fave-colors.sct")
```

a. Determine the red, green, and blue components of `fave1` using `rgb.red`, `rgb.green`, and `rgb.blue`.

b. Determine the red, green, and blue components of `fave1`, `fave2`, and `fave3` using `rgb->string`.

c. Determine the red, green, and blue components of `fave1`, `fave2`, and `fave3` using `rgb->rgb-list`.

Exercise 6: Nearby Colors

a. Pick one of your favorite colors (say `fave1`).

b. Create three colors that are similar to that color by varying each of the components by 16 or so. For example, you might create one color by adding 16 to each component, one by subtracting 16 from each component, and one by adding 16 to two components and subtracting 16 from the third. Name these three variants `variant1`, `variant2`, and `variant3`.

c. Draw pixels in each of those colors.

d. Find out what names DrFu assigns to each of those colors.

Exercise 7: Psuedo-Complementary Colors

a. Find psuedo-complements to your three favorite colors and name them `evaf1`, `efaf2`, and `evaf3`.

b. Set the top-two pixels in column 0 of `canvas` to `fave1`, the top-two pixels in column 1 to `evaf1`, the top-two pixels in the column 2 to white, the pixels in column 3 to `fave2`, the top-two pixels in column 4 to `evaf2`, and so on and so forth.

c. What do you note about the relationship between colors and their psuedo-complements?

Exercise 8: Psuedo-Complementary Colors, Revisited

Consider the color 128/64/0. One natural psuedo-complement to that color is 127/191/255, for reasons given in the reading. However, there are certainly others.

a. Find three other psuedo-complements that are not too near each other. If you are not sure how to do so, read the notes on this problem.

b. Explore those colors by setting pixels in `canvas` to each color.

c. What do you note about relationships between the various psuedo-complements?

Exercise 9: Generating Colors

It is useful to be able to visualize the relationship between the RGB numbers and the colors they represent. Sometimes it's just fun to see how well you can guess the components of a color. For each of the following colors, guess the RGB components, build a pixel of that color to see how close you got, and then try again.

a. The dominant color of your skin.

b. A color of your shirt.

c. The color of the desks in this room.

d. The color of the sky right before a storm.

e. A color of your choice (which you should describe first).

Explorations

Explorations are intended for students interested in further exploring the design aspects of these techniques. They also provide students who finish early with extra activities that may challenge them in different ways.

Exploration 1: Color Context

While the representation of colors using red, green, and blue components is relatively straightforward, it ignores a significant issue: Human perception. Perhaps not so surprisingly, the way we perceive a particular color depends greatly upon the surrounding colors. We can use the techniques we've learned so far to explore this issue.

a. Create and show two images, `canvas-light` and `canvas-dark`, each 9x5. The background color of the first canvas should be a light grey, about 64/64/64. The background color of the second canvas should be a dark grey, about 192/192/192. Zoom in on each canvas to the full 16:1 magnification.

b. Set the center pixel of each canvas to a medium grey, 128/128/128.

c. Do the center pixels look the same to you? Should they?

d. [Fill in some standard color example that shows the same thing.]

e. Choose a simple color and conduct an experiment similar to the two preceding experiments. In this case, you should first surround the color by its psuedo-complement and then surround it by a nearby color. Share your images with a neighboring group.

Exploration 2: Related Colors

Pick a technique for generating a group of eight or so related colors. For example, you might choose a variety of complements to some color, you might choose colors in which the ratio of red to green and of red to blue is consistent (e.g., red is always twice green and triple blue), or you might start with a single color and find variants by repeatedly adding 16 to each component.

Create an interesting 5x5 image using those colors.

Notes

Preparation

To create a blank canvas and name it, type the following in the interactions window.

```
(define canvas (image.new 11 9))
(image.show canvas)
```

Notes on Exercise 4: Common Components

It should make some sense that the color 0/0/0 is black, since black is the absence of any color. To many, though, it makes less sense that 255/255/255 is white. However, if you ask a physicist, she will tell you that white is the combination of every color of light, so saturation of all three components makes sense. Alternately, we can simply think of white as the opposite of black. Once we know white and black, we can expect that every color with common components in between white and black is some kind of grey, so 128/128/128 is medium grey. (Or is that "medium *gray*"?)

Notes on Exercise 8: Psuedo-Complementary Colors, Revisited

For example, you might compute the psuedo-complement that gives 128/128/128, the psuedo-complemens that gives 192/192/192, and the psuedo-complement that gives 224/224/244.