

Class 12: Transforming Colors

Held: Wednesday, 16 September 2009

Summary: Today we explore common ways in which to transform colors, pixels, and images.

Related Pages:

- EBoard.
- Lab: Transforming RGB Colors.
- Reading: Transforming RGB Colors.
- Due: Assignment 3: Drawing Generally and Concisely.

Notes:

- Lab Writeup 3 distributed. Transforming RGB Colors, exercises 5-7.
- Exam 1 distributed.
- Reading for Friday: Transforming Images.
- Don't forget the Grinnell HS Homecoming Parade tomorrow @ 5:30 downtown.
- I hear the play is in need of more male actors. Even if you're an extra, you get to hang with the way cool cast.
- Any EC/Support?
- EC/Academic: Gene Gaub plays Brahms. 11 a.m. Thursday, Herrick.
- EC/Academic: Dennis Vaccaro '11 presents *Graphical User Interface development using the Qt toolkit*, Thursday, 3821, at 4:30. Refreshments at 4:15 in commons.
- Thoughts or questions on yesterday's lecture?

Overview:

- Review: Color basics.
- Computing new colors from old.
- An example.

Review: RGB Colors

- A standard way to represent transmissive colors.
- Think of it as a combination of three colored lights of varying intensities
 - Red
 - Green
 - Blue
- Why these colors? It just works that way.
- How do we measure intensities?
 - Some people like a 0-1 scale

- Computers like a 0-255 scale (powers of two rock!)
- Important procedures:
 - (rgb-new *red green blue*) - create a new color
 - (image-set-pixel! *image col row color*) - set the color of a pixel
 - (image-get-pixel *image col row*) - get the color of a pixel
 - (rgb-red *color*) - get the red component
 - (rgb-green *color*) - get the green component
 - (rgb-blue *color*) - get the blue component
 - (color->rgb-list *color*) - get a list of the three components
 - (color->rgb *color*) - convert some representations to the RGB representation.

Transforming Colors

- Today we are exploring ways to transform colors.
- Why?
 - Deepens our understanding of colors.
 - We can apply a color transformation to a pixel in the image.
 - We can apply a color transformation to each pixel in an image. (More on that tomorrow.)

An Example

- MediaScript comes with a variety of built-in color transformations.
- However, we can also define our own (yay!).
- The normal framework for a color transformation is

```
(define color-transform
  (lambda (color)
    (rgb-new ____ ; computation of red component
              ____ ; computation of green component
              ____ ; computation of blue component
    )))
```

- Let's consider a simple one: We'll keep only the blue component of an image.
- How would we write this?

```
(define only-blue
  (lambda (color)
    (rgb-new ____
              ____
              ____)))
```

- Some tests
- While we'll spend today using this idea with pixels, for this example, we'll explore the effect on a sample image.

Copyright © 2007-9 Janet Davis, Matthew Kluber, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials copyright by John David Stone and Henry Walker and used by permission.) This material is based upon work partially supported by the National Science Foundation under Grant No. CCLI-0633090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.