

## Assignment 6: Color-Reducing Filters

**Due:** 4:00 p.m., Tuesday, 18 September 2007

*No extensions!*

**Summary:** In this assignment, you will build a number of simple filters.

**Purposes:** To reinforce the basic concepts of anonymous functions. (After all, practice makes perfect.) To bridge concepts from color theory into the course. To give you the opportunity to explore the algorithms for somewhat more complex transformations

**Expected Time:** 1-2 hours (for those who completed most of the lab on anonymous procedures); 2-3 hours (for those who did not)

**Collaboration:** You must work in a group of size two or three. You may discuss this assignment and possible solutions with anyone you wish. If you discuss this assignment with people other than group members, make sure to include a citation (e.g., “I consulted *this person*, who helped me *do this*”).

**Submitting:** Email me your answer. More details below.

**Warning:** So that this assignment is a learning experience for everyone, I may spend class time publicly critiquing your work.

### Contents:

- Assignment
  - Part A: Flattening an Image
  - Part B: Greyscale (or Grayscale)
  - Part C: Sepia
  - Part D: A Color-Blend Filter
  - Part E: Explorations
- Important Evaluation Criteria
- Submitting Your Homework

## Assignment

### Part A: Flattening an Image

Complete Exercise 6 from the lab on writing anonymous procedures.

## Part B: Greyscale (or Grayscale)

Complete Exercise 3 from the lab on writing anonymous procedures.

## Part C: Sepia

Older photographs are not black and white, but rather closer to brown and white, due to some combination of the chemicals used to make images and the ravages of time. (Sorry, I don't know what old photographs looked like when they were new.) Some people like the aged look of sepia-toned images enough to desire them.

To produce a sepia-tone image, start with a grayscale image you created with the code from Part B. Then, write a new instruction to transform the grayscale image so that black becomes sepia (a very dark brown), while white remains white.

How can we accomplish this? We want to adjust the colors in the image so that every pixel has a red component that is at least as large as the red component of sepia. But, so that we don't lose distinctions between bright pixels in the image, the red component should not be 255 (the maximum) unless it was 255 in the original image. Numbers in between should be scaled appropriately. Similarly for the green and blue components.

There is not a color named "sepia" in the color name list, but

```
(define sepia (cname->rgb "bakers chocolate"))
```

gives a pretty good approximation.

If you write your instructions well, you should be able to replace `sepia` with another dark color (and thus get a green-scale or blue-scale image, for example).

## Part D: A Color-Blend Filter

The last part specified that "white remains white." What if we used a light color instead of white, and a dark color instead of black? In effect, we transform white to that light color, black to a dark color, and shades of grey to intermediates between the light and dark colors..

Define `dark` to be a dark color of your choice and `light` to be a light color of your choice. Write a Scheme instruction to transform the grayscale image such that black is transformed to `dark`, white is transformed to `light`, and the colors in between are scaled appropriately. Your instructions from Part C should be a helpful starting place.

## Part E: Explorations

Find a compelling pair of colors to use with your color-blend filter from Part D. Try to describe the color scheme using the color theory terminology we learned from Professor Kluber.

## **Important Evaluation Criteria**

I will primarily look at whether your instructions are correct, automatic, and general. Since many of you have been negligent in writing general code, I will pay particular attention to the generality of your code: How easy will it be to change your program to do something similar?

I am also likely to consider at the formatting and readability of your code.

## **Submitting Your Homework**

Please submit this work via email. The email should be titled CSC151.02 Assignment 06 and should contain your answers to all parts of this assignment. Make sure to include your answer to Part E. You need not attach any images, but I may ask you to demonstrate them in class.

Please send your work as the body of an email message. I don't like attachments, and prefer not to receive them when they can be avoided.

---

Copyright © 2007 Janet Davis, Matthew Kluber, and Samuel A. Rebelsky. (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.