

Assignment 7: Overlapping Bordered Gradients

Due: 4:00 p.m.,
No extensions!

Summary: In this assignment, you will extend the ideas from the lab on iterating over positions in an image to build an interesting composite image.

Purposes: To give you further experience with the `region.compute-pixels!` procedure. To help you think about applications of recent techniques to making images.

Expected Time: 1-2 hours

Collaboration: I encourage you to work in groups of size three. You may, however, work alone or work in a group of size two or size four. You may discuss this assignment with anyone, provided you credit such discussions when you submit the assignment.

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:

- Background
- Assignment
- Important Evaluation Criteria
- Submitting Your Homework

Background

In a recent lab, you learned how to use the `region.compute-pixels!` procedure to create rectangles in which the color at any position is computed by a formula based on the position. We call such computed swatches of color *gradients*. For example, the following puts a 41x41 box with a simple gradient somewhere in the middle of the image called `canvas`.

```
(define square (lambda (x) (* x x)))
(region.compute-pixels!
 canvas
 80 80
120 120
 (lambda (pos)
  (rgb.new (* 0.5 (+ (square (- (position.row pos) 100))
                    (square (- (position.col pos) 100)))))
    (* 150 (abs (sin (* .2 (position.row pos))))))
  0)))
```

We've also seen how to put borders on those gradients. One simple technique is to draw a larger, single-colored, box before we draw the gradient. For example, we might precede the preceding code with

```
(region.compute-pixels!  
  canvas 76 76 124 124  
  (lambda (pos) (rgb.new 0 0 0)))
```

Assignment

Create an interesting image with three bordered rectangles. Each rectangle should have a different gradient, although you should use a similar formula to compute each gradient. (You might vary gradients by changing scale factors in your computation, for example.) At least two of the gradients should overlap. You may choose the size of the image, the size of the rectangles, the colors in your gradients, and the formula for the gradients.

Write a short paragraph that explains your choice of gradients and placement.

Important Evaluation Criteria

I will primarily judge your work on your success on the structure of the code for your gradients.

You are likely to receive a higher grade if you give a more careful analysis of how you arrived at your code and how the code relates to your goals for the gradient.

You are also likely to receive a higher grade if you find a way to generalize your code.

Submitting Your Homework

Please submit this work via email. The email should be titled CSC151.02 Assignment 07 and should contain your code for creating the image.

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.

You need not attach the image you've created - I can run the code to create the image. (Note, also, that the code is likely to be much smaller than the image.)

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.