CSC151.02 2013F, Class 09: An Introduction to the GIMP

Overview

- Preliminaries
  - Admin.
  - Preparation for quiz / confusing topics / etc.
  - Quiz
- Lab

Admin

- Warning! Friday the 13th falls on a Friday this month. (Today even.)
- The outline and what we’re doing don’t match very well today. We’ve rearranged the course, but I like leaving my old notes around.
- If you’re finding this difficult, know that you’re not alone - this stuff is challenging.
- I don’t think folks are taking enough advantage of the mentor sessions. If you’re having difficulty in the class and not going, please drop me a note to explain why.
- Reading for Monday: Programming the GIMP Tools
- EC Opportunities:
  - Submit a short paragraph giving reactions/thoughts/etc. (I don’t need a summary)
  - CS Table NEXT Friday (Pair programming)
  - Soccer 1:30 pm Saturday against people from a very flat state
  - Other?

Preparation for Quiz (and other discussion items)

I have a bit of code to show you that I hope resolves some recent questions. Then I’ll give you ten minutes for additonal questions.

```racket
#lang racket
(require gigls/unsafe)

;;; File:
;;; sample.rkt
;;; Author:
;;; Samuel A. Rebelsky
;;; Summary:
;;; A quick sample to illustrate a variety of points about drawings.

;;; Procedure:
;;; bigred
;;; Parameters:
;;; drawing, a drawing
```
;;; Purpose:
;;;   To confuse students
;;; Produces:
;;;   A drawing
;;; Props:
;;;   Taken from a clicker question designed by SR and JD
(define bigred
  (lambda (drawing)
    (drawing-recolor drawing "red")
    (drawing-scale drawing 10)))

;;; Procedure:
;;;   preview
;;; Parameters:
;;;   d, a drawing
;;; Purpose:
;;;   preview the drawing in a new window
;;; Produces:
;;;   img, the id of the preview image
;;; Preconditions:
;;;   (drawing-right d) >= 1
;;;   (drawing-bottom d) >= 1
;;; Postconditions:
;;;   img contains a rendering of d
(define preview
  (lambda (d)
    (image-show (drawing->image d (drawing-right d) (drawing-bottom d))))))

;;; Value
;;;   d1, a sample drawing
;;; Type:
;;;   drawing
;;; Contents:
;;;   A black, 50x50 circle centered at (25,25)
(define d1
  (drawing-hshift
    (drawing-vshift
      (drawing-scale drawing-unit-circle 50)
      25)
    25))

;;; Value:
;;;   d2
;;; Type:
;;;   drawing
;;; Contents:
;;;   A purple 50x50 circle, centered at (75,75)
(define d2
  (drawing-hshift
    (drawing-vshift
      (drawing-scale (drawing-recolor drawing-unit-circle "purple")
        50)
      75)
    75))

;;; Value:
;;;  d3
;;;  Type:
;;;     drawing
;;;  Contents:
;;;     A green rectangle
(define d3
  (drawing-hshift
   (drawing-vshift
    (drawing-scale
     (drawing-recolor
      drawing-unit-square
      "green")
    30)
   20)
  80))

;;;  Value:
;;;     scene
;;;  Type:
;;;     drawing
;;;  Summary:
;;;     A simple scene
(define scene
  (drawing-group d1 d2 d3))

;;;  Procedure:
;;;     drawing-munge
;;;  Parameters:
;;;     drawing, a drawing
;;;  Purpose:
;;;     Create a new drawing, based on the original drawing
;;;  Produces:
;;;     new-drawing, a drawing
(define drawing-munge
  (lambda (d)
    (drawing-recolor (drawing-scale d 1.1) "blue")))

* Puzzlement about what what happens with parameters and what happens when we don’t name drawings

* Disappearing values

3 + (4*5)

Just as intermediate values in a calculation disappear, so do intermedia drawings.

* Behind the scenes: Procedures

* When does order matter?
  - Scale vs. shift (traditionally scale first)
  - Group vs. almost anything
What’s the difference between drawing-bottom and drawing-height

Where should your detail be: In the parameters or the pre/post?
  ○ I like detail in the pre/post
  ○ But documentation is like an essay, not code - it need not be perfect at this stage of your career. As long as it makes sense, I’ll probably accept it
  ○ Is it always possible to express postconditions formally? NO

Can we derive immense satisfaction from watching SamR screw up while trying to do relatively easy things in DrRacket/Gimp?
  ○ Yes
  ○ And you should take that as a moral of "it’s okay to screw up"
  ○ [No, I don’t do so intentionally, just to make that point.] (No, I don’t do so intentionally, just to make that point.)

Quiz

Is the code in part 1 in the definitions pane or the interactions pane?
  ○ It shouldn’t matter - It will give the same result either way

Lab

Quick.