Fundamentals of Computer Science I (CS151.01 2006F)

Homework 11: A Procedure is Worth a Thousand Pictures

Assigned: Tuesday, 24 October 2006
Due: Friday, 27 October 2006
No extensions!

Summary: In this assignment, you will further explore the design of Script-Fu procedures for the GIMP and how you might parameterize those pictures.

Purposes: To give you further experience with Script-Fu. To help you think about parameterization.

Expected Time: One to two hours.

Collaboration: You may work in a group of any size between one and four, inclusive. You may consult others outside your group, provided you cite those others. You need only submit one assignment per group.

Submitting: Email me your work, using a subject of *CSC151 Homework 11*.

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

Background

One of the advantages of scripting simple images is that you can *parameterize* those images; drawing different images based on different choices. For example, a procedure that draws a face might take as parameters the "roundness" of the face, the color of the eyes, and the type of hair.

In some situations, it is equally interesting to make the parameters control aspects of the image, but in a less obvious way. For example, one might take a number as a parameter to the face-drawing routine and use that number to determine the type of hair. (The programmer might reveal the translation of number to hair type to the client, or may leave it as a secret.)

Assignment

In this assignment, you will write the latter kind of parameterized drawing procedure. In particular, you should write a procedure `(draw-something` *`val1 val2 val3`*`)` that takes three integers in the range 0-9 as parameters and draws different versions of the image based on the value of those parameters.

Once this procedure is available, you can then easily draw “random” pictures by selecting random values for the parameters.

```
(draw-something (random 10) (random 10) (random 10))
```

We recommend that you choose a straightforward category of image, such as a cartoonish face or a house.

Helpful Hints

There are a number of ways to convert numbers to aspects of your drawing. For example, if your drawing is of a face, you can use the first number to specify the width of the face with something like the following:

```
(let ((face-height 120)
      (face-width (+ 100 (* val1 4))))
  ...)
```

It is possible to use the number to select between different categorical values, such as eye colors.

```
(define eyecolors
        (list BLUE GREEN COPPER GREY BROWN
              DIM_GREY STEEL_BLUE PALE_BLUE DARK_BROWN VERY_DARK_BROWN))
...
(let ((eyecolor (list-ref eyecolors val2)))
  ...)
```

It is even possible to use one value to modify two different attributes. For example, in the following, `val3` is used to select both the brush used to draw the hair and the kind of hair drawn. Since there are five different brushes and two kinds of hair, each different number gives a different kind of hair.

```
(define hair-brushes
  (list "Calligraphic Brush" "Circle (03)" "Circle (05)"
        "Circle Fuzzy (03)" "Felt Pen"))
...
(set-brush (list-ref hair-brushes (modulo val3 5)))
(if (< val3 5)
    (draw-curly-hair)
    (draw-straight-hair))
```
