Laboratory: Raster Graphics: Images from Pixels and Colors

Summary: In this laboratory, you will explore some issues with the creation and modification of simple raster graphics, primarily in terms of the creation and analysis of colors.

Reference:

- `(cname.list)` - list all the available named colors
- `(cname->rgb` *`name`*`)` - given the name of a color, convert it to a representation that can be used by `image.set-pixel!` and other procedures
- `(image.get-pixel` *`image column row`*`)` - get the color of the pixel at the given position
- `(image.height` *`image`*`)` - determine the height of the specified image
- `(image.load` *`filename`*`)` - load an image from the specified file
- `(image.new` *`width height`*`)` - create an image of the specified width and height
- `(image.set-pixel!` *`image column row color`*`)` - set the color of the pixel at the given point
- `(image.show` *`image`*`)` - display the specified image
- `(image.width` *`image`*`)` - determine the width of the specified image
- `(list-colors` *`name`*`)` - list all the available named colors that contain a particular name
- `(rgb->cname` *`color`*`)` - given a color obtained from an image, find a common name for that color

Contents:

Preparation

Start DrFu (which starts The GIMP and the DrScheme interface for programming). If you've forgotten how to do so, review the first DrFu lab.

Exercises

Exercise 1: Creating Some Images

For this laboratory, you will need to work with two images.

a. Using the techniques described in the corresponding reading, create and show a 9x5 image called `landscape`.

Recall that `(image.new` ***`width height`***`)` creates an image and `(define` ***`name expression`***`)` assigns a name to a computed value.

b. As you may recall, the initial color associated with an image is the current background color. Change the background color to yellow in preparation for the next step.

Recall that `(envt.set-bgcolor!` ***`color`***`)` sets the background color and that `(cname->rgb` ***`name`***`)` lets you get a color by name.

c. Using the same technique that you used for step a, create and show a 5x7 image called `portrait`.

Exercise 2: Changing Selected Pixels

a. Make the top-left pixel of each of the two images red. Recall that you set pixels using `(image.set-pixel!` ***`image column row color`***`)` and that you can get a color from its name with `(cname->rgb "`***`name`***`")`.

b. Make the bottom-right pixel of each of the two images black.

c. You may have found that it is difficult to see the changes that you've made. To better observe these changes, let us enlarge the picture.

- Click on the picture to bring it to the front.
- From the View menu at the top of the window, select the Zoom submenu and then select 16:1 (1600%)

Exercise 3: A Simple Form

Pick one of your two images and write instructions to make a green plus sign in the exact center of the image. The plus sign should be three pixels wide and three pixels high.

Exercise 4: Selecting New Colors

As you may recall from the reading, the `(cname.list` *`name`*`)` procedure lets you find out all the color names that DrFu knows about that include *name*. For example, `(cname.list "red")` will tell you about various red-like colors.

Set the upper-right corner of each image to one of those variants of red.

Exercise 5: Selecting New Colors, Revisited

Of course, you should not be limited by our suggestions for possible colors. Write down the names of five of your favorite colors and then see if any of them are defined by using `cname.list`.

a. Using those results or other explorations of the DrFu color space, pick three color names that you like, find the corresponding RGB values, and name them `fave1`, `fave2`, and `fave3`.

For example, if spicy pink were my favorite color, I might write

```
(define fave1 (cname->rgb "spicy pink"))
```

You may find it easiest to look at colors using `envt.set-fgcolor!`. For example, if I wanted to see what spicy pink looked like, I might use

```
(envt.set-fgcolor! "spicy pink")
```

b. Set some pixels to your favorite colors.

c. Create a new file, `/home/`*`username`*`/Desktop/fave-colors.sct`, that contains these three definitions. You may want to review the DrFu lab for how to save definitions.

Exercise 6: Representing Colors

Recall that you can find out the value DrFu associates with a name by typing it in the interactions window and then hitting enter.

a. Find out what values DrFu associates with your three favorite colors.

b. Are those values clearer or less clear than the names you've used?

c. (Optional) Hypothesize what those values represent.

Exercise 7: From Colors to Names

As you may recall from the reading, one way we deal with the lack of clarity of color values is to use the `rgb->cname` procedure to find the name of a similar color.

Verify that `rgb->cname` successfully finds the names of your three favorite colors.

Exercise 8: Working with Existing Images

The `(image.load "`*`file-name`*`")` procedure allows us to load an image from a file. As is the case with `create-image`, `load-image` does not immediately show the image it has loaded. And, as is the case with `image.new`, you'll need to name the image you've just loaded.

a. Load the image `/home/rebelsky/glimmer/samples/rebelsky-stalkernet.jpg`, name it `sample`, and show it in a new window.

b. Save your StalkerNet picture to `/home/`*`username`*`/Desktop/me.jpg`, load that picture, name it `self-portrait`, and show it in a new window.

How do you save your StalkerNet picture? You first find yourself in StalkerNet. Next, you right click on the picture, select Save As ... from the menu that appears, navigate to the desktop, and choose the name `me.jpq`..

Although Grinnell College owns the copyright in your StalkerNet picture and tells us so repeatedly in strong language, our use of that picture for this lab and elsewhere in the class clearly falls under the rubric of *fair use*.

c. Set the bottom-left pixel in each of the two new images to white.

Exercise 9: Getting Information from Existing Images

a. Set the color of the center pixel in `portrait` to the color of the center pixel in `self-portrait`. (If there is not a center pixel in `self-portrait` because it has even width or height, use one near the center.)

b. Find out the name of the color of some pixel in `self-portrait` using `image.get-pixel` and `rgb->cname`.

Explorations

Explorations are intended for students interested in further exploring the design aspects of these techniques. They also provide students who finish early with extra activities that may challenge them in different ways.

Create a new 9x5 image called `canvas`. Fill the pixels of the canvas with variations of blue to produce something you find compelling.

If those instructions are too vague, you might consider how to use those shades of blue to make an image that induces some mood, such as relaxation or playfulness.

---

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.