

Program #3 - Image Pal (ette)

Prof Bill - May 2018

Program #3 logistics:

- Due: **Thu May 17, 2018** at the beginning of class (> 2 weeks)
- Worth: **8 points** (8% of your grade)
- Learn: algorithm design, image processing, Big-O, Comparator, hash table, JavaFX, and JCF (ArrayList, Collections, HashMap)

1. Description

The different colors used in an image are called it's palette. The goal of P3 is to:

1. Reduce the number of colors in an image's palette, and
2. While having the least impact on image quality

Well, how are we going to do that?

I have no idea.

thanks... yow, bill



Source: en.wikipedia.org/wiki/List_of_color_palettes

1. Background

OK, I have some idea of how to do this. (ha)

But still, P3 will not just be coding. A big part of P3 will be your research and experimentation with different ideas and algorithms to solve our palette problem.

Rationale

Why do this? What is the benefit of reducing the palette size of an image?

- **History** - Monitors of the past could only display 256 colors at a time. This saved time and space by representing each pixel with 8 bits, which is 1 byte.
- **Web browser** - The web palette defined 256 colors that were “safe” on all the devices on which a web browser ran.
- **Today** - We can still reduce the size of some image files by reducing the number of different colors in an image. How much? Let’s find out.

Images

Terms: image coordinates, RGB pixels, distance between two colors, image palette

JavaFX: Image, ImageView, Color

Palette size

To start, we’ll want to determine how many different colors there currently are in our image. This is the palette size.

Please do this three ways: 1) sequential search and 2) get() using a hash table. We should get the same results with each structure. Processing large images, we should be able to “see” (and feel) the Big-O of these structures.

Reduce palette size

Reducing the number of colors means combining two or more colors into one. This is our primary research topic. A couple points immediately come to mind:

- Combine colors that are not used much
- Combine colors that are similar (the distance between them is small)

That’s just scratches the surface. We’ll happily spend class time to brainstorm P3 ideas.

Images collection

We'll gather a collection of images to share for this program. These will give you fodder for your trial and error experiments.

2. Feature list

Here we go:

- Read and display an image in your GUI
- Report on image basics: height, width, num pixels
- Each student will contribute an image to the 210 collection (for playing/testing).
Decide on categories and choose: selfie, nature, man made, black and white?
- Count the number of different colors in an image
 - Sequential search with an ArrayList, $O(n)$
 - `get()` with a HashMap, $O(1)$
 - Track (benchmark) your performance to compare the 2 structures
- Reduce the palette size to 256 colors
 - Other approaches?
 - Show palette stats (% of colors in top 256, etc)
- Save image after your changes

Creativity: Your creative component will be the algorithms you use to choose colors and reduce palette size. This and your experiments to get good/interesting results.

Remember

How to succeed (writing any program):

1. Start early!
2. Ask questions in class. Email me. Come to office hours.
3. Small bites. Divide and conquer your program into small, manageable tasks.
4. Always be green. Always be working. Your program should always compile and run. Use the debugger. Never leave your work in disarray.

3. Grading

Create a **program3** folder on your k: drive. This folder should contain:

- All your Java source files
- Your program3 executable (jar or class files)
- Any test input and output files that you have

Special for P3 - Include a **brief write-up** of your algorithm, performance, and results. Save your favorite images where the palette size was reduced to 256 colors.

All your code must follow our class **Coding Guidelines**. Ugly code will be severely penalized. A program that doesn't even compile is probably worth 0 points.

Remember our **plagiarism** guidelines as well. Getting help from google or stackoverflow or a friend is OK, but:

1. You must acknowledge any help you receive with a comment in your code
2. You must understand any code in your solution
3. Get help on program components, not the assignment (the tic tac toe philosophy)
4. If you have any questions in this area, contact me **before** you turn in your work, not after (when it's too late)

thanks... yow, bill

PS - For these guys, P3 = Protein, Protein, Protein. Creativity!

www.proteinproteinprotein.com

