CSC 204 Lab 9: Playing with Pictures

Goals

After doing this lab, you should be able to:

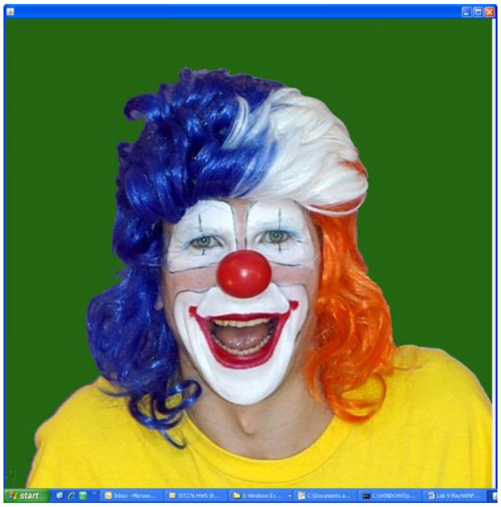
- write Java programs that manipulate pixels of a picture through our textbook's Picture class as can be found in Chapter 2's and Chapter 6's Worked Example. The API for the Picture class is located in our Big Java folder, ch02, worked_example_2, api folder.
- do creative 'things' to pictures.

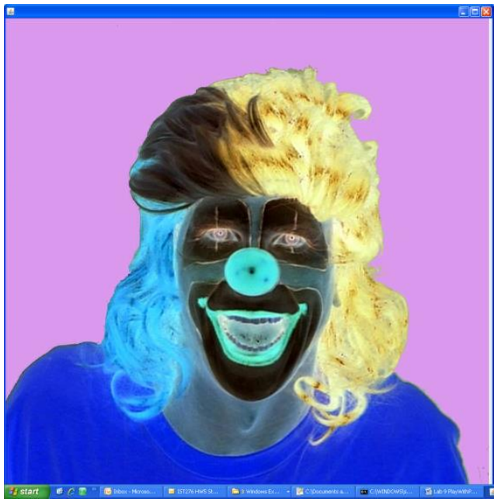
Lab Preparation

Read through this lab. Take a look at the Picture class API. Create a new Eclipse Java Project named Lab9 in your Orion folder on Blackhawk. Copy the two Java files from our courses Lab9 folder on blackhawk into the src folder of this project on Eclipse. Copy the image files from our courses folder into your Lab9 project folder in Eclipse (NOTE, pictures get copied into the Lab9 folder, not the src folder). Feel free to copy some of your own pictures into this folder, too. :)

Getting Acquainted with the Picture Class

Run the program and become familiar with how it works. Look at the code and understand how the nested loops accomplish the picture transformations. Understanding this code will assist you in writing your code described below.





Part 1

Now it is your turn to write new picture transformations. For each of the following picture transformations you should write the transformation method as described below.

- flipTB() – Flip Top to Bottom. This method should be very similar to flipLR(). The only difference is that you want to swap the corresponding top and bottom pixels.
- mirrorLR() – Mirror Left to Right. This method should alter thePicture so that the right side mirrors the left side. That is, imagine an invisible line drawn down the center of thePicture. Think of that line as a mirror and every pixel on the right 'reflects' the corresponding pixel on the left. This code should be very similar to flipLR() – copy pixels, don't swap pixels.
- mirrorTB() – Mirror Top to Bottom. This method mirrors (hehehe) mirrorLR(). The only difference is that this time draw the imaginary line from left to right half way down thePicture and let the bottom pixels mirror the corresponding top pixels.
- rgb2gbr() – Shift the RGB values. This method should look a lot like the makeNegative() method. For each pixel in thePicture, shift the red, green, and blue components as follows:

    1. get the three integer intensities for each pixels red, green, and blue,
    2. assign red the intensity from green,
    3. assign green the intensity from blue, and
    4. assign blue the intensity from red.

Part 2

Finally, you get to be creative and Razzle Dazzle your friends. Come up with your own transformations and add them to this program. Here are some suggestions… but you should try to be more creative and come up with something very creative.

- Put a border around the picture.
- Give the picture some checkerboard effect.
- Mirror around a diagonal.
- Cut the image into four quadrants and duplicate the picture in each.
- Soften the image colors (average each pixel's RGB values with those around it).
- Make all very dark pixels lighter (or vice versa).
- Make every other pixel its negative.
- "Blueify" the image (or any colorify).

Create your best Razzle Dazzle photo and save that picture to turn in. Feel free to add some of your own pictures to this assignment!!

Deliverables

When finished copy your PlayWithPictures.java file, and your best Razzle Dazzle picture, into a Lab9 folder in your shared Google drive. If your Razzle Dazzle picture started with a picture other than those given with this assignment, please put a copy of your original picture in the shared folder, too. Have fun and be creative!!!