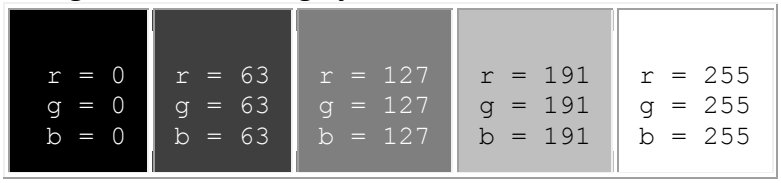
CS 111 – Program Design I, Fall 2015

Lab 6

Images Using a GrayScale

A black and white picture uses various shades of gray. These shades of grey all have the same amount of red, green and blue color. The higher the color value, the lighter the shade of gray is. The lower the color value, the darker the shade of gray is. For example, look at the following squares for various shades of gray. The amount of red, green and blue for the color is displayed as text in each square. Note that the amount of red, green and blue is the same in each square. Also note that the larger the amount of color, the lighter the color of gray.



To see the entire range of gray scale images, refer to the picture below. The left side of the picture has colors set to black (r = 0, g = 0, b=0). The right side of the picture has colors set to white (r=255, g=255, b=255). The middle colors shift from black to white based on X coordinate of the pixel.

Changing the color in a pixel

The color in a pixel can be changed by using the **setRed()**, **setGreen()** and **setBlue()** functions in the Pixel class as part of the Java bookClasses.

Determining the amount of gray

One method to determine which grayscale color to use for a pixel when creating a black and white picture from a color picture is to average the amount of red, green and blue color the corresponding pixel in the colored picture. This method works fairly well and is fairly easy to understand and use. This way uses the intensity of each color to determine the grayscale value. However, this assumes that the human eye sees the "brightness" or "luminance" of red, green and blue equally well. However, this is not the case. The human eye perceives green as brighter than red and it perceives red as brighter than blue. So a standard average calculation is NOT the best way to determine the grayscale amount.

Another method is to take a weighted average of the amount of red, green and blue color instead of an evenly-weighted average. A weighted average would give more weight to certain values and less weight to other values. For example, a weighted average will be used when determining the grade for this course. Exams have a higher weight (i.e. more impact on the final grade), while lab assignments have a lower weight (i.e. less impact on the final grade).

For determining the amount of brightness for the grayscale color, the human eye needs to use a weighted average. The original amount of green used should account for almost twice as much as the 33.3% which would be used in a normal averaging. The original amount of red used should be just a bit less than the 33.3% which would be used in a normal average. The original amount of blue should only account for a third of the expected 33.3% which would be used in a normal averaging. Below us the percent amounts for each color that should be used for this lab. These values were determined according to some research done on luminance.

- Red : 29.9%
- Green : 58.7%
- Blue : 11.4%

Using this weighted value is close to what was done for the code written in class for Lect1008d.java.

Lab Assignment 6

Due: Wednesday, 10/14/2015 by 11:59 pm

Write two Java programs that will do the following:

1. **Program 1: Make Black And Green**
    - The Java program is to be named: NetidLab6a
    - Print out your name and your net-id
    - Allow the user to select a picture from a file stored on the local machine
    - Call a method that will change the image to a "black-to-green scale" image. This is done in a similar manner as the creation of a grayscale image, except the values for red and blue are kept at zero. Only the green value for each pixel will range from 0 to 255 based on the luminance of the original pixel. Thus, where the grayscale image is black, the green scale image will also be black. Where the grayscale image is white, the green scale image will be green. Where the grayscale image is gray, this image will range from black to green (the middle values will be a shade of dark green). To see the variations of color from black to green, check out the following picture:

      Note: **This is NOT done by only setting the red and blue color values to 0 and leaving the green value unmodified. You must modify all three color values in every pixel! (The red and blue values get set to 0, while the green value gets set to the weighted average of the original red, green and blue values.)**

    - Display the modified image
    - Save the modified image to a file on the local machine

2. **Program 2: Make Green And White**
    - The Java program is to be named: NetidLab6b
    - Print out your name and your net-id
    - Allow the user to select a picture from a file stored on the local machine
    - Call a method that will change the image to a version of a green scale image. This is done in a similar manner as the creation of a grayscale image, but with a twist. Where the grayscale image is black, this image will be green. Where the grayscale image is white, this image will also be white. Where the grayscale image is gray, this image will range from green to white (the middle values will be light green). Look at the table below to determine which color value need to change and which one need to remain constant. Also check out the following picture to see the colors range as its changes from green to white.

      The green value should be set to the maximum value, while the red and blue values are modified to reflect the luminance/grayscale amount.

      Note: **This is NOT done by only setting the green color value to 255 and leaving the red and blue color values unmodified. You must modify all three color values in every pixel! (The green values gets set to 255 while the red and blue values get set to the weighted average of the original red, green and blue values.)**

    - Display the modified image
    - Save the modified image to a file on the local machine

3. You must write your programs using good programming style which includes:
    - Good variable names
    - in-line commenting
    - header block commenting for the program and each method written
      Be sure to include the following with the header block comment for the program.
        - your name
        - day and time of your CS 111 lab section (i.e. Monday at 3:00)
        - A description of the project.
    - proper indentation of program statements
    - use of blank lines to separate blocks of code.

4. You are to submit both Java files electronically via the assignment link in Blackboard for Lab 1. You can upload and submit two files in blackboard.

   Please only submit source code files (the **.java** file, not the **.class** or the **.java~**).

   Also, if you have any comments about your program, please write it down in the header block comment of the .java file; please do NOT write in the "Comments" field on the submission page (this will go into a different file and will be easily missed).

The discussion below is to help understand the differences between a black and white picture versus a black and green versus a green and white picture. Using the below information, we can determine some of the before and after colors. The "lum" value below is the average for the the r, g and b values listed. Notice how the lum value in the first row is used in the each of the other 3 rows.

original color values (note: the "lum" value is the important value)	r = 0 g = 0 b = 0 lum = 0	r = 9 g = 150 b = 30 lum = 63	r = 255 g = 0 b = 126 lum = 127	r = 150 g = 168 b = 255 lum = 191	r = 255 g = 255 b = 255 lum = 255
makeBlackAndWhite()	r = 0 g = 0 b = 0	r = 63 g = 63 b = 63	r = 127 g = 127 b = 127	r = 191 g = 191 b = 191	r = 255 g = 255 b = 255
makeBlackAndGreen()	r = 0 g = 0 b = 0	r = 0 g = 63 b = 0	r = 0 g = 127 b = 0	r = 0 g = 191 b = 0	r = 0 g = 255 b = 0
makeGreenAndWhite()	r = 0 g = 255 b = 0	r = 63 g = 255 b = 63	r = 127 g = 255 b = 127	r = 191 g = 255 b = 191	r = 255 g = 255 b = 255

The following are some images created based on the ideas in this lab. The following image is the original full color picture.

The following image is the black and white version of the original image.

The following image is the black and green version of the original image.

The following image is the green and white version of the original image.