

CS 111 – Program Design I, Fall 2016

Lab 6

Images Using a ColorScale

A black and white picture uses numerous shades of gray. These shades of gray 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. Since our color values range from 0 to 255, this gives us 256 shades of gray.

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
-------------------------	----------------------------	-------------------------------	-------------------------------	-------------------------------

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 JES library.

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.

Another method is to take a weighted average of the amount of red, green and blue color instead of an evenly-weighted average. This is shown in the code [lect1006a.py](#). 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). When looking at colors, the human eye perceives green as a brighter color than red and red as a brighter color than blue. This “brightness of color” is often call “luminance”. For determining the amount of color for the grayscale color according to luminance, the original amount of green is given a higher weight while the original amount of blue is given a lesser weight. Based on research done in the how human perceived

luminance, the following percent weights of each color that should be used when determining the proper spot of the Color Scale:

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

For this assignment, you are to use the above weighted values when calculating the luminance of a color in a pixel for the Color Scale amount. You are NOT to use the simple average calculation as was done in lect1004d.py.

Lab Assignment 6

Due: Wednesday, 10/12/2016 by 11:59 pm

Write two python functions that will do the following in a program file called netidlab6.py:

1. Function 1: Make Black And Red

- The python function is to be named: makeBlackRed ()
- 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-red 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 red 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 red scale image will also be black. Where the grayscale image is white, the red scale image will be red. Where the grayscale image is gray, this image will range from black to red (the middle values will be a shade of dark red). To see the variations of color from black to red, check out the following picture:



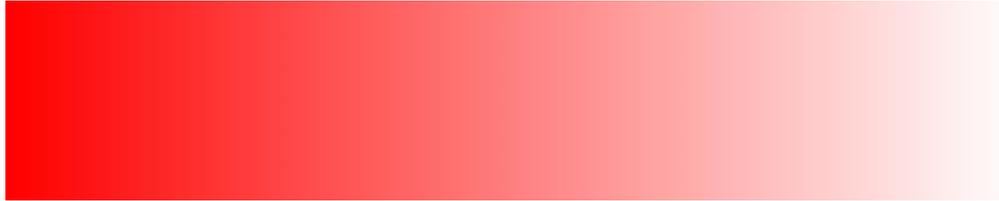
Note: **This is NOT done by only setting the green and blue color values to 0 and leaving the red value unmodified. You must modify all three color values in every pixel! The green and blue values do get set to 0, but the red value gets set to the weighted average of the original red, green and blue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

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

2. Program 2: Make Red And White

- The python function is to be named: makeRedWhite ()
- 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 “red-to-white” 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 red. Where the grayscale image is white, this image will also be white. Where the grayscale image is gray, this image will range from red to white (the middle values will be light red). 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 red to white.



The red value should be set to the maximum value, while the green and blue values are modified to reflect the luminance/grayscale amount. Note: **This is NOT done by only setting the red color value to 255 and leaving the green and blue color values unmodified. You must modify all three color values in every pixel! The red values gets set to 255, while the green and blue values get set to the weighted average of the original red, green and blue values).**

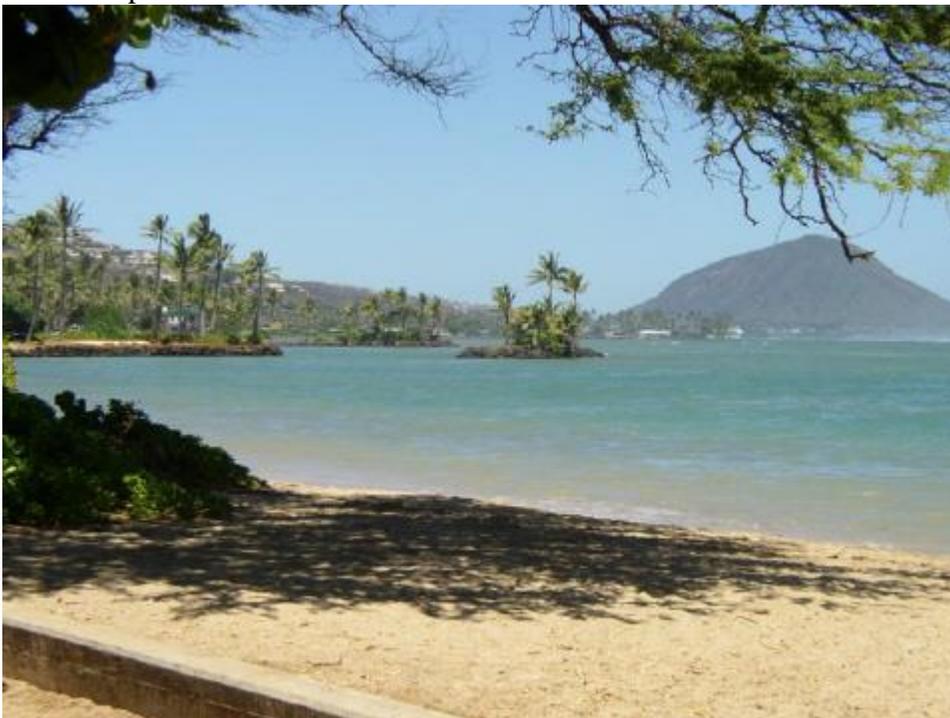
- Display the modified image
 - Save the modified image to a file on the local machine
3. You must write your programs using good programming style which includes:
- Good variable names
 - in-line commenting
 - header block commenting for the program and each method written
- Be sure to include the following 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 the python file electronically via the assignment link in Blackboard for Lab 1.

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-red versus a red-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)	<code>r = 0</code> <code>g = 0</code> <code>b = 0</code> <code>lum = 0</code>	<code>r = 9</code> <code>g = 150</code> <code>b = 30</code> <code>lum = 63</code>	<code>r = 255</code> <code>g = 0</code> <code>b = 126</code> <code>lum = 127</code>	<code>r = 150</code> <code>g = 168</code> <code>b = 255</code> <code>lum = 191</code>	<code>r = 255</code> <code>g = 255</code> <code>b = 255</code> <code>lum = 255</code>
<code>makeBlackAndWhite()</code>	<code>r = 0</code> <code>g = 0</code> <code>b = 0</code>	<code>r = 63</code> <code>g = 63</code> <code>b = 63</code>	<code>r = 127</code> <code>g = 127</code> <code>b = 127</code>	<code>r = 191</code> <code>g = 191</code> <code>b = 191</code>	<code>r = 255</code> <code>g = 255</code> <code>b = 255</code>
<code>makeBlackAndRed()</code>	<code>r = 0</code> <code>g = 0</code> <code>b = 0</code>	<code>r = 63</code> <code>g = 0</code> <code>b = 0</code>	<code>r = 127</code> <code>g = 0</code> <code>b = 0</code>	<code>r = 191</code> <code>g = 0</code> <code>b = 0</code>	<code>r = 255</code> <code>g = 0</code> <code>b = 0</code>
<code>makeRedAndWhite()</code>	<code>r = 255</code> <code>g = 0</code> <code>b = 0</code>	<code>r = 255</code> <code>g = 63</code> <code>b = 63</code>	<code>r = 255</code> <code>g = 127</code> <code>b = 127</code>	<code>r = 255</code> <code>g = 191</code> <code>b = 191</code>	<code>r = 255</code> <code>g = 255</code> <code>b = 255</code>

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 red version of the original image.



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

