

Assignment 3: Basic Filters

Due: Tuesday, March 1, 11:59 PM

Make sure you read the directions carefully. This assignment consists of several pieces and you need to complete all of the pieces for full credit.

- This assignment consists of the creation of several methods as outlined below. These methods should all be saved in a file called hw4.py
- To test these methods described below you may use any image file of your choice. This can be one of the files we have used in class, or it may be one that you found on the Internet or took yourself. Do not use large images: JES was not built for speed.

This assignment is worth 40 points, which accounts for approx. 4.3% of your final grade.

Task 1 (20 points)

In the textbook readings for this week you learned that gray is any color where the red, green, and blue values in a pixel are all equal.

For example,

```
setRed(pixel, 128)
setGreen(pixel, 128)
setBlue(pixel, 128)
```

produces a middle of the scale gray. Since you can set each color channel to any value from 0 to 255 there are 256 gray colors in the representation of color we use in this class (notice this definition means that both black and white are, in fact, versions of gray).

We extend this idea by defining that grayscale images are any picture where every pixel is set to some version of gray. There are a variety of techniques we can use to decide which version of gray each pixel should be set to. In your textbook you read that one way is to use the concept of a pixel's luminance - the average of its three color components. This is, however, only ONE way to determine which version of gray to select for a pixel. In this task, you will explore different methods for selecting the intensity value to which you set each pixel.

Put the following versions of makeGrayscale into your file.

1) **makeGrayscaleV1(picture)** should use the value in the red channel to determine the luminance of a particular pixel. That is, if the color of a pixel is (R=111, G=24, B=216) then the pixel's grayscale color should be (R=111, G=111, B=111).

2) **makeGrayscaleV2(picture)** should use the following formula:

```
lum = 0.3 * red + 0.59 * green + 0.11 * blue
```

That is, if the color of a pixel is (R=111, G=24, B=216), then the pixel's grayscale color should be:

```
111 * 0.3 + 24 * 0.59 + 216 * 0.11 = 182 or (R=182, G=182, B=182)
```

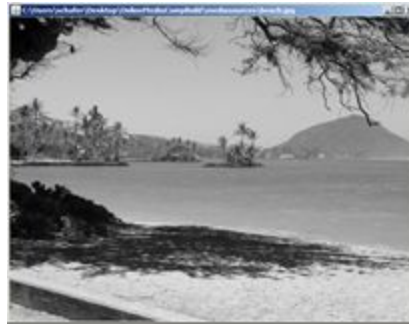
Task 2 (10 points)

If you think of "old fashioned" photographs like cowboy photos from the American West of the 1800s, you probably don't actually think about grayscale photos but instead you think of sepia-toned photos. Sepia tone was a printing technique that gave photos a sort of yellowish color rather than gray.

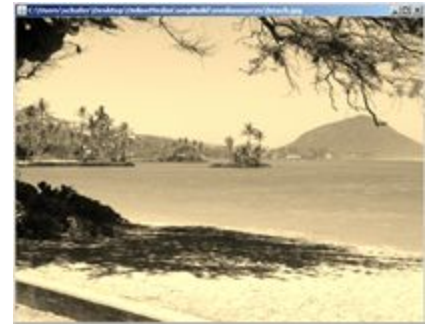
For example, here is the same photo in color, grayscale, and sepia-toned:



color version



grayscale version



sepia-toned version

Using the color values [referenced on wikipedia](#) create a method called `makeSepiaTone()` which takes a picture as a parameter and converts that photo to sepia tone.

```
newRed = (R * 0.393 + G * 0.769 + B * 0.189)
newGreen = (R * 0.349 + G * 0.686 + B * 0.168)
newBlue = (R * 0.272 + G * 0.534 + B * 0.131)
```

NOTE: If you look at this closely you will see that this formula does not always produce "valid" results. That is, it will produce RGB values greater than 255 if the original color is close to white.

For example, if my color was (R=250, G=248, B=239) then the formula above would produce a NEW red value of

```
newRed = (250 * 0.393 + 248 * 0.769 + 239 * 0.189)
newRed =      98.25      +      190.712      +      45.171
newRed =                               334.133
```

If you use `setRed(pixel,334)` then JES will automatically change any values greater than 255 to 255 (it will also set any values less than zero to zero, but that won't happen here).

Finally, please note that your textbook contains a program for Sepia much later in the book. This version has absolutely nothing to do with your task in this assignment. Using the code from the book will result in an automatic score of zero for this assignment.

Task 3 (10 points)

UNI has a slogan that says “Purple for Life” For this task you will write a function called purple4Life() that takes in one picture as a parameter and tints it “purple” You should experiment with what modifications to the Red, Green, and Blue channels (if any) are necessary to give your photo a nice purple color while retaining the basic original image.

For example, here is a photo that I tinted purple:



Submissions

Your file hw3.py submitted to eLearning. Your file should contain the following functions:

```
makeGrayscaleV1 (picture)
```

```
makeGrayscaleV2 (picture)
```

```
makeSepiaTone (picture)
```

```
purple4Life (picture)
```