

Assignment 3
Image Manipulation — using Java
Due: 12 noon on Friday 26th May, 2006

Outline

Your aim in this assignment is to write some computer image manipulation tools for **Portable PixMap** (ppm) images. This is essentially the same as Assignment 1, but in this case you are to use the Java programming language.

A number of Java classes to represent channels, pixels and images, and a driver program to test your work, are provided in `/dept/dcs/comp1100/public/assts/asst3/` and on the course web site.

Some sample ppm images are available in `/dept/dcs/comp1100/public/assts/asst1Images/` and on the course web site. If you wish to make your own images (make sure they're not too big) you can convert from other formats to ASCII ppm with a tool like Photoshop or the Gimp. *Eye Of Gnome* (eog) is a useful image display tool usually distributed with Linux, but there are others. Be sure to turn off interpolation to see true pixels when zooming in.

Required Tasks

You are required to design, develop and test several image manipulation tools. Since they are naturally classified as PPM methods, they will belong in the PPM class. Some of them need to manipulate individual pixels, so those methods will be included in the Pixel class.

1. Desaturate

Write a method that converts an image to *greyscale*, that is, shades of grey. This is often called *desaturation*. The method will be in class PPM and have type:

```
void desaturate()
```

Hint: Begin by writing a method (in class Pixel) to convert pixels to greyscale. To convert a pixel to greyscale, each of its RGB values is changed to the average of the RGB values rounded down to an Integer. For example, (30, 101, 200) converts to (110, 110, 110).

Note: In Java, the division operator (/) performs integer division (like Haskell's div operator) when applied to Integer values. For example, 13/2 gives 6.

2. Double Scale

Include a method `void doubleScale()` in class `PPM` which doubles the dimensions of an image (in both horizontal and vertical directions).

Hint: Begin by implementing a function:

```
ArrayList<Pixel> doubleRow(ArrayList<Pixel> row)
```

which returns a list with each pixel repeated.

Free Choice Tasks

Successful completion of the required tasks above, to the satisfaction of your tutor and lecturer is sufficient to achieve up to 80% on this assignment. For marks better than that, you will need to develop some image manipulation tools of your own design.

There are two basic categories: *pixel-based*, where you modify each pixel in some manner (such as desaturation); and *image-based*, where the structure of the image is modified in some way (such as scaling). You should develop one of each kind. I recommend that you choose some of the same image manipulation tools that you developed for Assignment 1.

Submission

You should submit all the modules that you have written and wish to be included in your assessment, by the published deadline. Late submissions will be penalised 10% for each day (or part thereof) past the deadline.

Assignment submission will be electronic. Submit your assignment with the command:

```
submit comp1100 Asst3 <file names>
```

You should only have modified `Pixel.java` and `PPM.java`, so only submit those files. Make sure the two required methods have exactly the same names as given above.

You should check that your submissions have been successful by clicking the **View Marks** button in **StReaMS** (<http://cs.anu.edu.au/streams/>). You should see a list of files together with the time and date of submission.

Assessment

This assignment forms 10% of your total assessment for COMP1100.

Successful completion of the *Required Tasks* to the satisfaction of your tutor and lecturer is sufficient to achieve a mark of 80% on this assignment. For marks better than a that, you will need to develop some image manipulation tools of your own design, as explained in *Free Choice Tasks*.