Homework 3

September 26th 2019

Return your solutions (i.e., .cpp/.h files) to tuomas.lunttila@uef.fi by October 9th.

Implement a class for doing stuff with 8-bit greyscale images. The class should include member functions for reading and writing images from/to file. I will provide an example image in the homework folder of the Moodle pages. You can use either binary or text IO: the text file `test_asc.dat` starts with two lines with one number on each line; these tell the x and y dimensions of the image (both 512 in this case). The subsequent 512 lines all have 512 values between 0 and 255; these are the pixel (brightness) values of the grayscale image. The binary file `test.dat` starts with two 32-bit unsigned integers containing the dimensions followed by 512*512 bytes of image data (8-bit unsigned integers).

Your class should keep the image data in an array allocated with `new` (just to practice using that, so don't use C++ vector class this time). Implement copy constructor and copy assignment operators that do the appropriate (deep) copying as well as a destructor that deallocates arrays.

So that your image class does something useful, implement a member function that calculates the pixel histogram (i.e., it should return a vector which tells how many times each of the 256 possible pixel values occurs in the image).

You can get up to three bonus points for implementing functions to do other (non-trivial) image processing stuff (one point for each more or less correctly implemented function), for example Gaussian blur (`https://en.wikipedia.org/wiki/Gaussian_blur`), histogram equalisation (`https://en.wikipedia.org/wiki/Histogram_equalization`), or median filtering (`https://en.wikipedia.org/wiki/Median_filter`).