CSE274 Fall 2012
HW1 – Making Images the Hard Way

What is the point of this assignment?
You should get your programming environment set up, and also learn how to do, in C++, things you already know how to do in Java. In particular, this assignment teaches you:

- How to get the Visual Studio IDE setup
- How to use Git
- How to use variables, arrays, file input and output, and the command-line in C++
- How images are represented by a computer, using row-major order.

(10 pts) Phase 0: Getting set-up, checked-in. Due Mon, August 27, by noon.
Goals: Your program should change the background color of the screen in an interesting way. You should learn how to create a project, and check it in to github.

Steps:

1. Follow the first 4 videos at http://www.youtube.com/playlist?list=PL04769D8290155B15

What is DUE:

1. Your "CatPicture" app should be checked in to GitHub.
2. You should email me your github user id. This will allow me to make a branch of your repository, write my comments in your code, and send you a pull request with my comments added to your file.
3. Your CatPicture app should change the background color each time through the update/draw loop, but it does not have to be just like mine. Have some fun playing around with the different color options.

(70 pts) Phase 1: Making pictures. Due Monday, Sept. 3, by noon.
Goals: You should create some methods that draw or modify a picture by modifying the pixel array directly. See below for more details. **It is up to you to design and document your function parameters.** This is required to get points for a particular function. Use the same "Javadoc" type formatting you would use in Java. Your function should be fairly general, so there ought to be some way to draw objects of different sizes, different colors, and in different positions on the screen.

Main goals (max 60 pts):

A. Implement 4 of the following (10 pts each)

1. rectangle – Draw a rectangle (filled or empty, your choice) on the Surface. Must access the Surface's pixel array directly.
2. circle – Draw a circle (filled or empty, your choice) on the Surface. Must access the Surface's pixel array directly.
3. line – Draw a line segment on the Surface. Must access the Surface's pixel array directly.
4. gradient –Fills the entire surface with a gradient that smoothly changes from the first color to the second one. Must access pixel array directly. The shape and direction of the gradient can be whatever you wish.
5. copy – Copy a rectangular region of pixels from one place in the image to another.
6. tint – Apply a colored tint to the whole image. For example, if the image was a blue circle on a white background, you could add a red tint, to create a blue-purple circle on a pinkish background.
7. triangle – Given three points in the image, draw a triangle with those three points as the corners. May be filled or empty, your choice. Must access the Surface's pixel array directly.

B. Implement 1 of the following (10 pts)

1. Blur – For each pixel in the image, change it to be the average of itself with its 8 neighboring pixels. Note: You may want to clone your surface, and then work from the cloned surface … otherwise you will overwrite data that you need later on! This corresponds to a convolution with the following kernel:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

2. Edge detect – Do a convolution, just like in blur, but use the following kernel instead:

- 1/8	0	1/8
-1/4	0	1/4
-1/8	0	1/8

   Note that this can give values between -127 and +127, so you will need to add 127 to the result before storing the result. Note that you may wish to convert the image to grayscale before doing this, or you may wish to do each color (red, green, blue) individually. You will get different results depending on what you do. This kernel detects vertical edges, but it isn't hard to modify it to detect horizontal edges.
3. sharpen – Do a convolution, just like in blur and edge detect, but use the following kernel instead:

- 1/16	-1/16	-1/16
-1/16	8/16	-1/16
-1/16	-1/16	-1/16

   Note that this can give values between -127 and 127, so you will need to add 127 to the result before storing it. This is only useful if your image contains some blurry edges (for example, if you start by loading a photograph as in our CatPicture example).

4. Invent your own convolution-based filter, or find an interesting one online, to implement in your project.

C. Use your methods to draw a picture (5 pts)

D. Save your image to an image file, so it can be easily shared (5 pts)
Note: If you make your work interactive or animated, I'd love to get a video instead of an image. Please post to YouTube, and then send me the link.

Stretch goals (max 10 pts):

E. Implement 2 of the following (5 pts each)

1. anti-aliasing – For any of your drawing methods (rectangle, circle, line, etc), use anti-aliasing to make them look smooth.
2. Transparency – For any or all of your methods, add support for transparency in the color input variable. This means you will have to blend the new color with the existing color in each pixel.
3. Rotation – Add an input variable to your rect method that specifies an angle of rotation, in degrees.
4. 3-D – Use red/blue colors so that your image looks 3-D when viewed with red/blue 3-D glasses. You can borrow a pair from me (first come, first served until I run out).
5. Animation – Use the update/draw loop to make your picture animate in some interesting way.
6. Mouse interaction – Make your artwork interactive, using mouse clicks in some interesting way

What is DUE:

1. HW01 must be committed and pushed to github.
2. Email your image file to me by the due date. (Or if posting a video on YouTube, the YouTube link.) I will be showing them during class.

(20 pts) Phase 2: Code review. Due Wednesday, Sept. 5, by noon.

Goals: You will be assigned a partner. Fork your partner's CatPicture repository, and review the code. Make any changes you think useful, and then submit a pull request to your partner, asking them to accept your changed into their repository. You should spend about 2 hours on this. If you run out of time, even though there is more to fix, it is okay to stop. Please leave your partner's program in a working condition, however, so that accepting your pull request does not break their program.

What is DUE:

1. Your pull request must be submitted. I would like to see comments/discussion on the pull request, but this means you will need to have your pull request submitted before the dealine, to allow time for discussion. It is okay to start work on this BEFORE the due date for Phase 1!