ITSC 1212 – Assignment 3 – Pixel Tinting!

The main idea behind this assignment is to demonstrate your understanding of for-each loops and while loops and basic pixel color manipulation. After completing Labs 1-8, and programs 1-15 in the textbook you should have all the skills you need to complete this assignment. Remember that assignments are to be done individually.

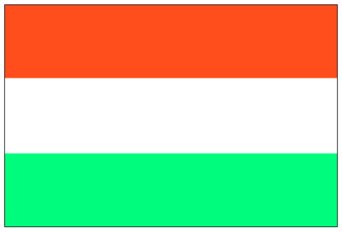
Part 1 – Go psychedelic! Make a tinting animation of your own headshot.
Part 2 – Go Dutch! Merge your headshot with a transparent Royal Dutch Navy flag.
Part 3 – Go blurry! Write an averaging function to blur your headshot.
Part 4 – Animate! Make an animation of your headshot that goes psychedelic for awhile, then goes Dutch, and then slowly gets more and more blurred.
Appendix A. Grading Rubric.
Appendix B. averagePixelsHorizontally() Starting Point

Part 1 – Go psychedelic! (20%)

For Part 1 of the assignment, use the base code Assignment3Part1.java as a starting point. In the Main() method, there is already code that will load in a picture from the command line or allow you to choose a picture to load. You must change the program to call at least three different color methods to tint your headshot different shades. There are already calls to the Thread.sleep(<milliseconds>) method to pause the image for awhile, so you can see your changes. In the end, you should have an animation of your own photo, changing colors. The color-tinting methods should be written in your Picture.java class. Make sure that you have three different methods that tint your face at least three different colors. Whether or not you use parameters to be able to tint at varying levels is up to you (in other words you could have a method that always tints your picture red by 50%, or you could have a method that tints your picture red by a percentage as a parameter).

Part 2 – Go Dutch! (25%)

For Part 2 of the assignment, you are to use base code Assignment3Part2.java as a starting point. The objective is to create a program that merges your headshot with the historical flag of the Royal Dutch Navy:



Hint: Use a while loop to loop through the pixels in your image, tinting the first third of the pixels orange, the second third of the pixels white, and the last third of the pixels green. Note that you want to move the RGB values <u>towards</u> orange, white, or green but not completely because then it would just look like the flag image above.

In the Main() method, write code that will load in a headshot image called myHeadShot.jpg (you will have to rename your photo to this). You should accomplish everything you need to do in the Main() method of Assignment3Part2.java.

Part 3 – Go blurry! (20%)

In this part of the assignment, your objective is to write a method in Picture.java that will blur an image by averaging each pixel with the pixels that come before and after it in the pixel array. So, for a random pixel with index “i" in your pixelArray, the red value of i should be the average of the red values of pixel i (the pixel i points to), pixel i-1 (the pixel to its left), and pixel i+1 (the pixel to its right). The same should be true of the blue parts of the pixel and the green parts of the pixel.

Assignment3Part3.java has some starting code in it for you. In the Main() method, write code that will load in a headshot image called myHeadShot.jpg (you will have to rename your photo to this). The important code we have provided for you in Appendix B below is a separate method called averagePixelsHorizontally(). Copy this method into your Picture.java file and work on it there.

Note that there are two special cases in this method that you have to think about:

- The very first pixel in the array can only be averaged with the pixel after it, since there is no pixel before it.
- The very last pixel in the array can only be averaged with the pixel before it, since there is no pixel after it.

We have created an IF statement that checks for these special conditions. All you have to do is fill in the code for what happens in the two special conditions and in the normal condition of a regular pixel anywhere else in the array. Note that these special conditions at the beginning and end of an array are often called ‘edge cases’. Processing edge cases properly is an important part of programming.

We suggest getting the standard case to work first, and then going back and writing the code for the special edge cases.

Once you have written the averagePixelsHorizontally method, you will want to test it, perhaps using the picture.explore() functionality to see the results. However, this averaging may be hard to detect visually after a single averaging call. We suggest that in your Main method() you call the averagePixelsHorizontally method at least three times so that you can see the blurring effect. You should probably write a loop to do this. You might also consider calling the averaging method many more times to see what the effect is.

Part 4: Animate! (20%)

Make an animation of your headshot that goes psychedelic three or four times, then goes Dutch, and then slowly gets more and more blurred.

You can use the `Thread.sleep(<milliseconds>)` method (see Part 1) to pause the image for a while between each change (remember: 1,000 milliseconds is 1 second).

Note: For this part, create a new class file named Assignment3Part4.java with a main method that executes the animation. You can add methods to your picture class file as you see fit.

Coding Style – (15%)

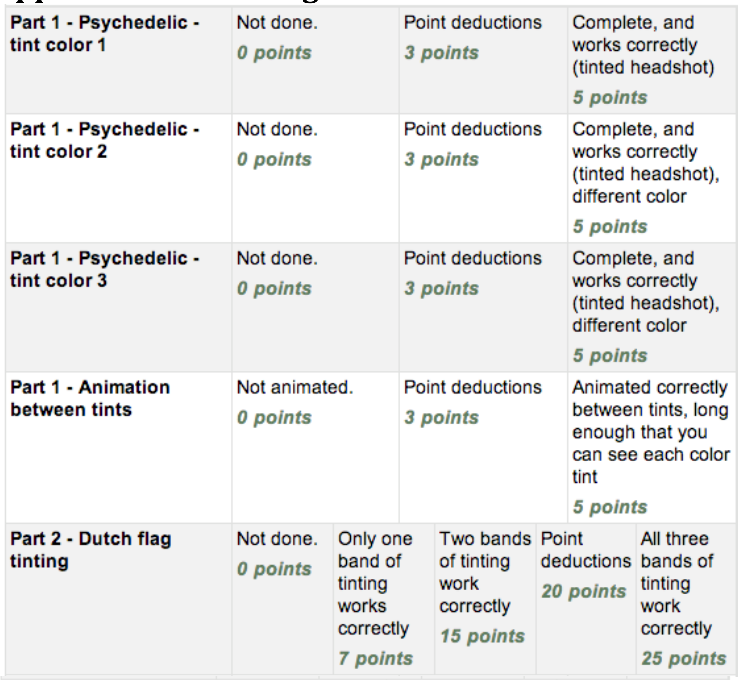
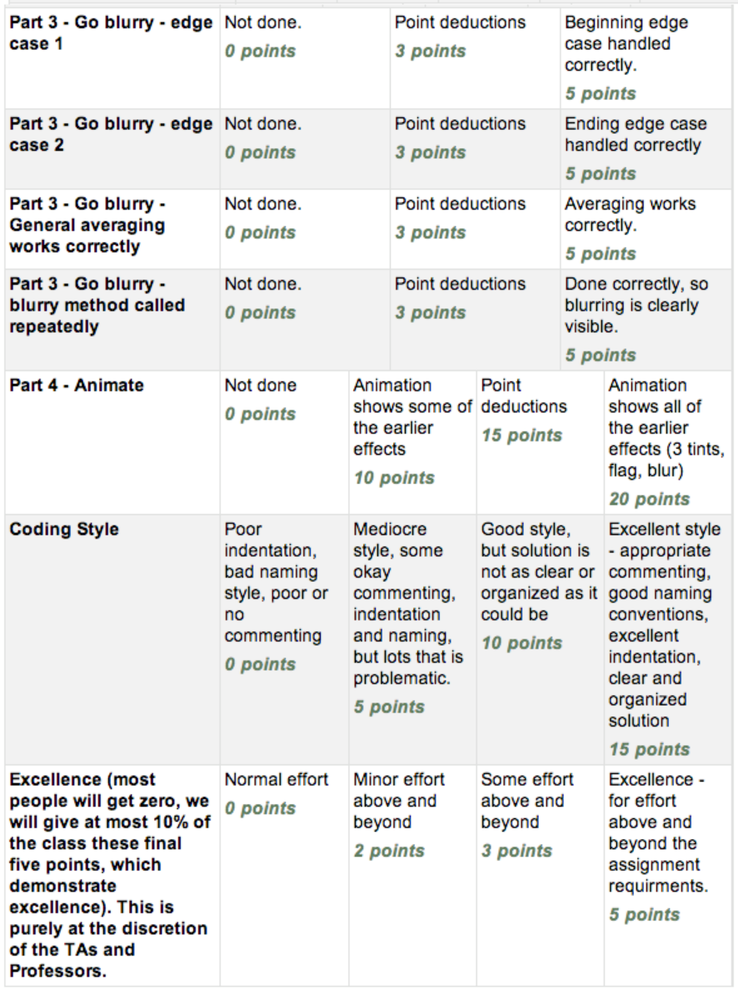
This grade is awarded for proper coding style. This includes:

- Appropriate method, variable, field, object and class names
- Proper indentation
- Good commenting (explains what code is doing)
- Well-organized, elegant solution

What to submit:

- Assignment3Part1.java
- Assignment3Part2.java
- Assignment3Part3.java
- Assignment3Part4.java
- Picture.java
- myHeadshot.jpg (the image you used for this assignment)

Appendix A. – Grading Rubric

Part 1 - Psychedelic - tint color 1	Not done. *0 points*		Point deductions *3 points*		Complete, and works correctly (tinted headshot) *5 points*
Part 1 - Psychedelic - tint color 2	Not done. *0 points*		Point deductions *3 points*		Complete, and works correctly (tinted headshot), different color *5 points*
Part 1 - Psychedelic - tint color 3	Not done. *0 points*		Point deductions *3 points*		Complete, and works correctly (tinted headshot), different color *5 points*
Part 1 - Animation between tints	Not animated. *0 points*		Point deductions *3 points*		Animated correctly between tints, long enough that you can see each color tint *5 points*
Part 2 - Dutch flag tinting	Not done. *0 points*	Only one band of tinting works correctly *7 points*	Two bands of tinting work correctly *15 points*	Point deductions *20 points*	All three bands of tinting work correctly *25 points*
Part 3 - Go blurry - edge case 1	Not done. *0 points*		Point deductions *3 points*		Beginning edge case handled correctly. *5 points*
Part 3 - Go blurry - edge case 2	Not done. *0 points*		Point deductions *3 points*		Ending edge case handled correctly *5 points*
Part 3 - Go blurry - General averaging works correctly	Not done. *0 points*		Point deductions *3 points*		Averaging works correctly. *5 points*
Part 3 - Go blurry - blurry method called repeatedly	Not done. *0 points*		Point deductions *3 points*		Done correctly, so blurring is clearly visible. *5 points*
Part 4 - Animate	Not done *0 points*	Animation shows some of the earlier effects *10 points*		Point deductions *15 points*	Animation shows all of the earlier effects (3 tints, flag, blur) *20 points*
Coding Style	Poor indentation, bad naming style, poor or no commenting *0 points*	Mediocre style, some okay commenting, indentation and naming, but lots that is problematic. *5 points*		Good style, but solution is not as clear or organized as it could be *10 points*	Excellent style - appropriate commenting, good naming conventions, excellent indentation, clear and organized solution *15 points*
Excellence (most people will get zero, we will give at most 10% of the class these final five points, which demonstrate excellence). This is purely at the discretion of the TAs and Professors.	Normal effort *0 points*	Minor effort above and beyond *2 points*		Some effort above and beyond *3 points*	Excellence - for effort above and beyond the assignment requirments. *5 points*

Appendix B. averagePixelsHorizontally() starting code

```
// TODO: copy this method into your Picture.java file and work on it there.
public void averagePixelsHorizontally() {

    // To get you started, these are the local variables you will need for this method
    Pixel[] pixelArray = this.getPixels();
    Pixel this_pixel = null;
    Pixel prev_pixel = null;
    Pixel next_pixel = null;
    int index = 0;
    int red;
    int green;
    int blue;


    while (index < pixelArray.length) {
      if (index == 0) {
        // special case, first pixel, no prev_pixel to average
        System.out.println("special case, first pixel");

        //TODO: put code here to average first pixel (index) with second pixel (index + 1)

      } else if (index == pixelArray.length - 1) {
        // special case, last pixel, no next_pixel to average
        System.out.println("special case, last pixel");

       // TODO: put code here to average last pixel (index) with second-last pixel (index – 1)

      } else {
        // standard case

         prev_pixel = pixelArray[index - 1];
         pixel = pixelArray[index];
         next_pixel = pixelArray[index + 1];

         // red pixel averaging
         red = (prev_pixel.getRed() + pixel.getRed() + next_pixel.getRed())/3;
         pixel.setRed(red);

         // green pixel averaging
         green = (prev_pixel.getGreen() + pixel.getGreen() + next_pixel.getGreen())/3;
         pixel.setGreen(green);

         // blue pixel averaging
         blue = (prev_pixel.getBlue() + pixel.getBlue() + next_pixel.getBlue())/3;
         pixel.setBlue(blue);


      } // end else
      index++;

    } // end while

  }// end method
```