Hands-on Experiment # 9-2 : Worksheet

Section_____2_____ Date_____07/04/2018_____

No more than 3 students per one submission of this worksheet.

Student ID _____6031851521_____ Name___Sarun Nuntaviriyakul_____

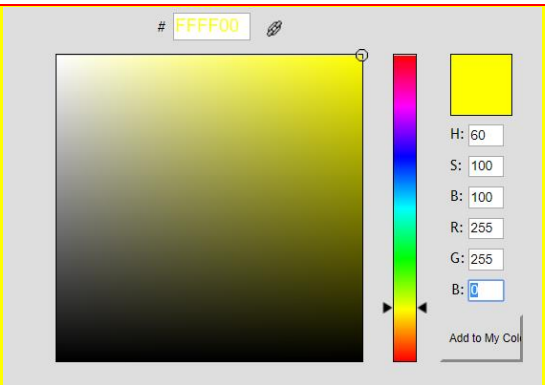
Student ID _____6031848721_____ Name___Watcharin Kriengwatana_____

Student ID _____6031847021_____ Name___Wasuthon Klyhirun_____

Part A: Understanding Given Resources/Backgrounds

1) Get yourself familiar with the RGB color model. Play around with the color picker on http://www.colorpicker.com/ and answer the following questions.
   a. What color is it that has the maximal value in R, the maximal value in B, and 0 in G? Capture the picture of the color and post it here.

Yellow.



   b. What are the requirements on the RGB values for all shades of gray?

The value of R,G and B must be the same.

2) Read the API specification of the class *Java101ImageUtil* in *Hands-on Experiment 9-2.pdf*.
   a. How many static methods are there in the class?

6

   b. How many overloaded methods are there in the class?

3

   c. Write the “method signatures” of all the overloaded methods. (** Write only the signature)

```
void showViewer(int[][][] a, String b)

void showViewer(int[][][] a, int[][][] b, String c)

void showViewer(int[][][][] a, String b)
```

3) Read the source code of *Java101ImageUtilExample.java* and try executing the program. Briefly explain what the program does. (** It is recommended NOT TO open big images. The program was not optimized in any ways. Try the program on some images with a few hundreds of pixels in their width/height)

First, the program will get a rgb array from the picture uploaded, then it will modify that array to the choice we choose. Finally, the program will convert from array to picture to show the picture and a modified picture.

Part B: Creating RGB arrays for Desired Images

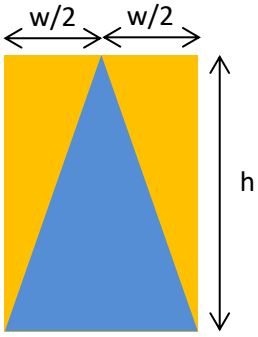
1) Write a program performing the following steps.
   a. Create a 3-D array of `int` that when used with `showViewer(int [ ][ ][ ],String)`, the program shows a 64-pixel x 128-pixel all-white image.
   b. Show the image with `showViewer(int [ ][ ][ ],String)`

   List your source code here.

```
public class WhiteImage{

    public static void main(String[] args) {

            int [][][] whiteImage = new int[64][128][3];

            for(int i=0;i<64;i++){

                    for(int j=0;j<128;j++){

                            whiteImage[i][j][0] = 255;

                            whiteImage[i][j][1] = 255;

                            whiteImage[i][j][2] = 255;

                    }

            }

            Java101ImageUtil.showViewer(whiteImage,"white image");
```

```
    }

}
```

2) Write another program performing the following steps.
   a. Ask the user to input the value of w and h, which are integers in the range of 100 to 200.
   b. Show an image of a triangle as shown in the figure below. Use the colors of your choice.



List your source code here.

```
import java.util.Scanner;

public class RGB{

    public static void main(String[] args) {

            int w ,h ;

            Scanner kb = new Scanner(System.in);

            System.out.print("enter width ");

            w = kb.nextInt();

            System.out.print("\nenter height ");

            h = kb.nextInt();

            double slope = (double) h / w * 2;

            int[][][] color = new int[w][h][3];

            for(int i = 0 ; i < w ; i++){
```

```
                    for (int j = 0 ; j < h ; j++ ) {
                        if( j < h-(i*slope) || j < (i*slope)-h){
                            color[i][j][0] = 0;
                            color[i][j][1] = 255;
                            color[i][j][2] = 255;

                        }
                        else {
                            color[i][j][0] = 0;
                            color[i][j][1] = 155;
                            color[i][j][2] = 155;
                        }
                    }
                }
                Java101ImageUtil.showViewer(color,"Triangle");
        }
}
```

Part C: Image Manipulation

Modify *DesaturateIt.java* to obtain a Java program performing the following steps.

1) Ask the user to select a gif or a jpg file.
2) Show the original image and its "desaturated" (grayscale) version using `showViewer()`.

Explain how the grayscale values are computed.

The value of new R,G and B can be calculated from the sum of its original R,G and B value divided by 3

[(R+G+B)/3]

List your source code here.

```
public class DesaturateIt{

	public static void main(String [] args){

		int [][][] rgb = Java101ImageUtil.getRGBArrayFromFile();

		if(rgb==null){return;}

		int [][][] gray = desaturate(rgb);

		Java101ImageUtil.showViewer(rgb,gray,"Desaturate It!");

	}

	public static int [][][] desaturate(int [][][] rgb){

		int[][][] gray = new int[rgb.length][rgb[0].length][3];

		for(int x=0;x<rgb.length;x++){

			for(int y=0;y<rgb[0].length;y++){

				int sum = (rgb[x][y][0] + rgb[x][y][1] + rgb[x][y][2])/3;

				gray[x][y][0] = sum;

				gray[x][y][1] = sum;

				gray[x][y][2] = sum;

			}

		}

		return gray;

	}

}
```

Submit this worksheet (by only one member of the group) via http://www.myCourseVille.com (Assignments > Hands-on Experiment # 9-2) **within the day after your lecture.**