**Picture Lab: Student Guide – Activity One and Two**

**Read through this word document, completing the activities and submitting this word document in Edmodo with filename: LastnamePictureOneTwo.docx.**

**Introduction**

In this lab you will be writing methods that modify digital pictures. In writing these methods you will learn how to traverse a two-dimensional array of integers or objects. You will also be introduced to nested loops, binary numbers, interfaces, and inheritance.

**Activities**

You will be working through a set of activities. These activities will help you learn about how:

*   digital pictures are represented on a computer;
*   the binary number system is used to represent values;
*   to create colors using light;
*   Java handles two-dimensional arrays;
*   data from a picture is stored; and
*   to modify a digital picture.
* **Set-up**
* You will need the pixLab folder and a Java Development Kit, also known as a JDK (see http://www.oracle.com/technetwork/java/javase/downloads/index.html). A development environment is also useful. Just open the files in the classes folder and compile them. Please note that there are two small pictures in the classes folder that need to remain there: leftArrow.gif and rightArrow.gif. If you copy the Java source files to another folder you must copy these gif files as well.
* Keep the images folder and the classes folder together in the pixLab folder. The FileChooser expects the images to be in a folder called images, at the same level as the classes folder. If it does not find the images there it also looks in the same folder as the class files that are executing. If you wish to modify this, change the FileChooser.java class to specify the folder where the pictures are stored. For example, if you want to store the images in “r://student/images/,” change the following line in the method getMediaDirectory() in FileChooser.java:
* URL fileURL = new URL(classURL,"../images/");
* And modify it to
* URL fileURL = new URL("r://student/images/");
* Then recompile.

I recommend keeping the default set-up of the classes folder and images folder in the pixLab folder. Rename your folder LastnamePixLab.

**A1: Introduction to digital pictures and color**

If you look at an advertisement for a digital camera, it will tell you how many *megapixels* the camera can record. What is a megapixel? A digital camera has sensors that record color at millions of points arranged in rows and columns (Figure 1). Each point is a *pixel* or *picture (abbreviated* ***pix****)* ***el****ement*. A *megapixel* is one million pixels. A 16.2 megapixel camera can store the color at over 16 million pixels. That’s a lot of pixels! Do you really need all of them? If you are sending a small version of your picture to a friend's phone, then just a few megapixels will be plenty. But, if you are printing a huge poster from a picture or you want to zoom in on part of the picture, then more pixels will give you more detail.

How is the color of a pixel recorded? It can be represented using the RGB (Red, Green, Blue) color model, which stores values for red, green, and blue, each ranging from 0 to 255. You can make yellow by combining red and green. That probably sounds strange, but combining pixels isn’t the same as mixing paint to make a color. The computer uses light to display color, not paint. Tilt the bottom of a CD in white light and you will see lots of colors. The CD acts as a prism and lets you see all the colors in white light. The RGB color model sometimes also stores an alpha value as well as the red, green, and blue values. The alpha value indicates how transparent or opaque the color is. A color that is transparent will let you see some of the color beneath it.

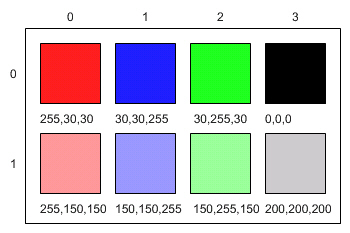


Figure 1: RGB values and the resulting colors displayed in rows and columns

How does the computer represent the values from 0 to 255? A decimal number uses the digits 0 to 9 and powers of 10 to represent values. The decimal number 325 means 5 ones (100) plus 2 tens (101) plus 3 hundreds (102) for a total of three hundred and twenty-five. Computers use *binary numbers*, which use the digits 0 and 1 and powers of 2 to represent values using groups of bits. A *bit* is a **b**inary dig**it,** which can be either 0 or 1. A group of 8 bits is called a *byte*. The binary number 110 means 0 ones (20) plus 1 two (21) plus 1 four (22), for a total of 6.

**Answer the following questions:**

1. How many bits does it take to represent the values from 0 to 255?   \_\_\_8\_\_\_\_\_\_\_\_
2. How many bytes does it take to represent a color in the RBG color model?  \_\_4\_\_\_\_\_\_\_\_\_\_\_
3. How many pixels are in a picture that is 640 pixels wide and 480 pixels high?  \_307200\_\_\_\_\_\_\_\_\_\_\_\_

**A2: Picking a color**

Run the main method in ColorChooser.java. This will pop up a window (Figure 2) asking you to pick a color. Click on the RGB tab and move the sliders to make different colors.

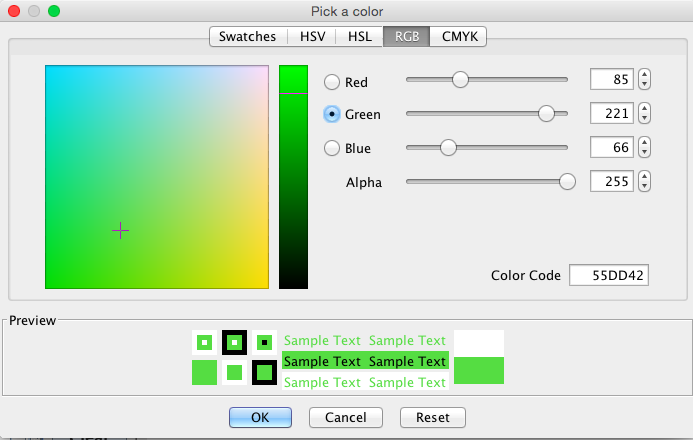


Figure 2: The Color Chooser

When you click the OK button, the red, green, and blue values for the color you picked will be displayed as shown below. The Color class has a toString method that displays the class name followed by the red, green, and blue values. The toString method is automatically called when you print an object.

java.awt.Color[r=85,g=221,b=66]

Java represents color using the java.awt.Color class. This is the *full name* for the Color class, which includes the *package* name of java.awt followed by a period and then the class name Color. Java groups related classes into *packages*. The *awt* stands for Abstract Windowing Toolkit, which is the package that contains the original Graphical User Interface (GUI) classes developed for Java. You can use just the short name for a class, like Color, as long as you include an import statement at the beginning of a class source file, as shown below. The Picture class contains the following import statement.

import java.awt.Color;

Use the ColorChooser class (run the main method) to answer the following questions.

**Answer the following questions:**

1. How can you make pink? State your RGB and Alpha values:  \_\_255,0,255,192\_\_\_\_\_\_\_\_\_
2. How can you make yellow?   State your RGB and Alpha values:  \_\_255,255,0,255
3. How can you make purple?   State your RGB and Alpha values:  \_\_128,0,255,255
4. How can you make white?   State your RGB and Alpha values:  \_\_255,255,255,255\_\_\_\_\_

5. How can you make dark gray? State your RGB and Alpha values:  \_\_32,32,32,255\_\_\_\_\_\_\_\_\_\_\_\_\_

Submit this word document in Edmodo with filename: LastnamePictureOneTwo.docx.