

Lab 2: ADT Simple RGB

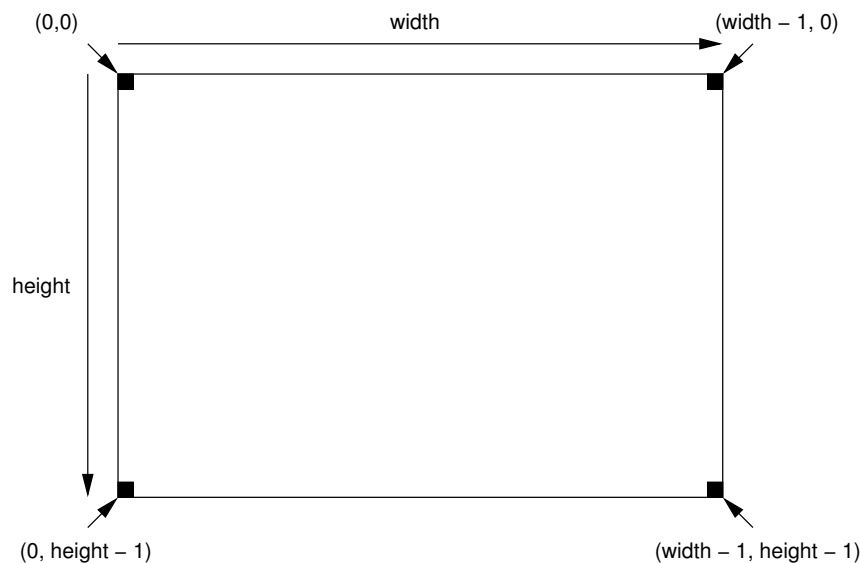
Location: CourseWeb -> Labs/Recitations -> Lab 2: Simple RGB

Download the following files:

1. SimpleRGB.java
2. SimpleRGBTester.java
3. ImageViewer.java
4. RGBComponent.java
5. pgh_640x480.jpg

Introduction

Imagine that you need to write an image processing program to manipulate RGB images. An RGB image consists of a series of pixels. Each pixel consists of three colors, red, green, and blue. Each color is represented by a number from 0 to 255. The number of pixels in an image of size width times height is the value of width multiplied by the value of height. A location of a pixel can be referred by (x, y) where $(0, 0)$ is the top left corner and $(width - 1, height - 1)$ is the bottom right corner as shown below:



To create an image processing application, you need a way to store images in your program. One way to do so is to imagine that an image is an object. This object should contain the width of the image, the height of the image, and red, green, and blue values of every pixel in the image. Let's assume that there is an ADT called Simple RGB. The data is organized like an image. It can be constructed by a specific width and height. Once it is constructed, it contains the value of width, the value of height, and red, green, and blue of every pixel in the image are initialized to 0s. Users can set the value of each color of a specific pixel by supplying x and y values that specify the pixel location and a color value (0 to 255). Users can read the value of each color in the similar manner. Methods for this ADT Simple RGB are shown below:

```
public int getWidth();
public int getHeight();
public void setRed(int x, int y, int aRed);
```

Lab 2: ADT Simple RGB

```
public void setGreen(int x, int y, int aGreen);
public void setBlue(int x, int y, int aBlue);
public int getRed(int x, int y);
public int getGreen(int x, int y);
public int getBlue(int x, int y);
public SimpleRGB getRedImage();
public SimpleRGB getGreenImage();
public SimpleRGB getBlueImage();
public SimpleRGB getGreyImage();
```

What to do

You are going to implement the ADT Simple RGB by creating a class named `SimpleRGB`. What you have to do is to modify the starting code (`SimpleRGB.java`). You need to add instance variables, implement the constructor, and implement all methods that marked `TO DO`.

A grey color is a color where red, green, and blue values are exactly the same. To convert an RGB image into a greyscale image, the red, green, and blue values of each pixel must be the same. For example, suppose the red, green, and blue value of the pixel (x,y) are R , G , and B , use the following formula

$$\text{newValue} = (0.21 \times R) + (0.72 \times G) + (0.07 \times B)$$

and use the `newValue` for all red, green, and blue of the grey scale image at (x,y).

Test Class

A test class (`SimpleRGBTester.java`) is provided. Simply compile and run this test class. This program will test your `SimpleRGB.java` and show your total points (out of 10). If you do not get the full 10 points, you should keep trying to fix your code until you get 10 points. **Note** that this test class is not perfect. It cannot tell you why your program is incorrect. You may have to look at the source code of `SimpleRGBTester.java` and see why it says `FAIL` and trace your code.

Example of Using the ADT Simple RGB

The program `ImageViewer.java` open an image file (`pgh_640x480.jpg`) and create an array of size 5 of `SimpleRGB` objects named `rgb`. The `rgb[0]` will be constructed using the width and the height of the image (`pgh_640x480.jpg`) and colors of each pixel will be set from the image. The rest of images will be initialized by calling `getRedImage()`, `getGreenImage()`, `getBlueImage()`, and `getGreyImage()`. The fraction of the code performing the above task is shown below:

```
final SimpleRGB[] rgb = new SimpleRGB[5];
rgb[0] = new SimpleRGB(width, height);

for(int i = 0; i < height; i++)
{
    for(int j = 0; j < width; j++)
    {
        Color c = new Color(originalImage.getRGB(j,i));
```

Lab 2: ADT Simple RGB

```
        rgb[0].setRed(j, i, c.getRed());  
        rgb[0].setGreen(j, i, c.getGreen());  
        rgb[0].setBlue(j, i, c.getBlue());  
    }  
}  
  
rgb[1] = rgb[0].getRedImage();  
rgb[2] = rgb[0].getGreenImage();  
rgb[3] = rgb[0].getBlueImage();  
rgb[4] = rgb[0].getGreyImage();
```

Then the program will show the original full color image as shown below. User can select to see just red, green, or blue color by clicking one of the button.



Due Date and Submission

For the due date, please check the lab in the CourseWeb. Submit your `SimpleRGB.java` to the CourseWeb under this lab by the due date. **No late submission will be accepted.**