Name: Andrew Martin  
Date: 4/6/21

**08.07 Picture Lab Worksheet**

**Directions**: Make note of your responses to the following questions as you work through the activities and exercise in the lesson.

**Activity 5 Questions**

|  |  |  |
| --- | --- | --- |
| **Question** | **Yes** | **No** |
| 1. Is the method getPixels2D in the Picture.java class? | ✓ |  |
| 1. Is the method getPixels2D in the SimplePicture.java class? | ✓ |  |
| 1. Will the following code compile?  DigitalPicture p = new DigitalPicture(); |  | ✓ |
| 1. Assuming a no-argument constructor exists for SimplePicture, will the following code compile?  DigitalPicture p = new SimplePicture(); | ✓ |  |
| 1. Assuming a no-argument constructor exists for Picture, will the following code compile?  DigitalPicture p = new Picture(); | ✓ |  |
| 1. Assuming a no-argument constructor exists for Picture, will the following code compile?  SimplePicture p = new Picture(); | ✓ |  |
| 1. Assuming a no-argument constructor exists for SimplePicture, will the following code compile?  Picture p = new SimplePicture(); |  | ✓ |

**Activity 5 Exercise Results**

1. Describe your method for keepOnly red, blue, or green.

The method keepOnlyBlue() loops through all the pixels and sets the red and green values to zero

1. For the negate method, paste your code related to calculating and setting the values for red, blue, and green.  
   Pixel[][] pixels = this.getPixels2D();

for (Pixel[] row : pixels) {

for (Pixel pixelObj : row) {

pixelObj.setRed(255 - pixelObj.getRed());

pixelObj.setGreen(255 - pixelObj.getGreen());

pixelObj.setBlue(255 - pixelObj.getBlue());

}

}

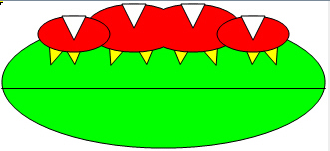
1. Paste a copy of the image that is the result of calling the grayscale.



1. For the method fixUnderwater, describe the algorithm you'd propose to accomplish the task.

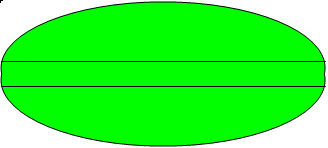
I would likely write an algorithm that would set each pixel’s blue value to a fraction of itself, because most of the noise in the picture is due to the overblown blue tint.

**Activity 6 Exercise Results**

1. Paste the image that is the result of calling the method mirrorVerticalRightToLeft.  
   
2. Describe the algorithm for the method mirrorHorizontal works.

This method loops through each row, looping through each pixel in the row and setting the color equal to the bottom of the picture.

1. Paste the image that is the result of calling the method mirrorHorizontalBotToTop.



**Activity 7 Questions**

* 1. How many times would the body of this nested for loop execute? \_90\_

for(int row = 7; row < 17; row++)

for(int col = 6; col < 15; col++)

* 1. How many times would the body of this nested for loop execute? \_90\_

for(int row = 5; row <= 11; row++)

for(int col = 3; col <= 18; col++)

**Activity 7 Exercise Results**

1. What value is displayed for count after the nested loop ends in the mirrorTemple method? \_4551\_
2. Paste the image that is the result of calling the method mirrorArms.  
   
3. Paste the image that is the result of calling the method mirrorGull.



PROJECT TITLE: 08.07\_Picture\_Lab

PURPOSE OF PROJECT: To explore and edit pictures and matrices

VERSION and DATE: 2.0 4/8/2021

AUTHORS: Andrew Martin

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* P M R \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<+s>: Mirroring the images along a vertical line was rather easy

<-s>: I did not plan ahead when I modified the vertical mirror to a horizontal one, so it took me a while to get it to work

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In the future: When modifying a method, I should make a list of changes that need to be made before implementing them.