Name:   
Date:

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

**Activity 5 Exercise Results**

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

I made a method called keepOnlyBlue that only keeps blue pixels. This method uses a 2D array to cycle through all the pixels and sets red and green pixels to 0.

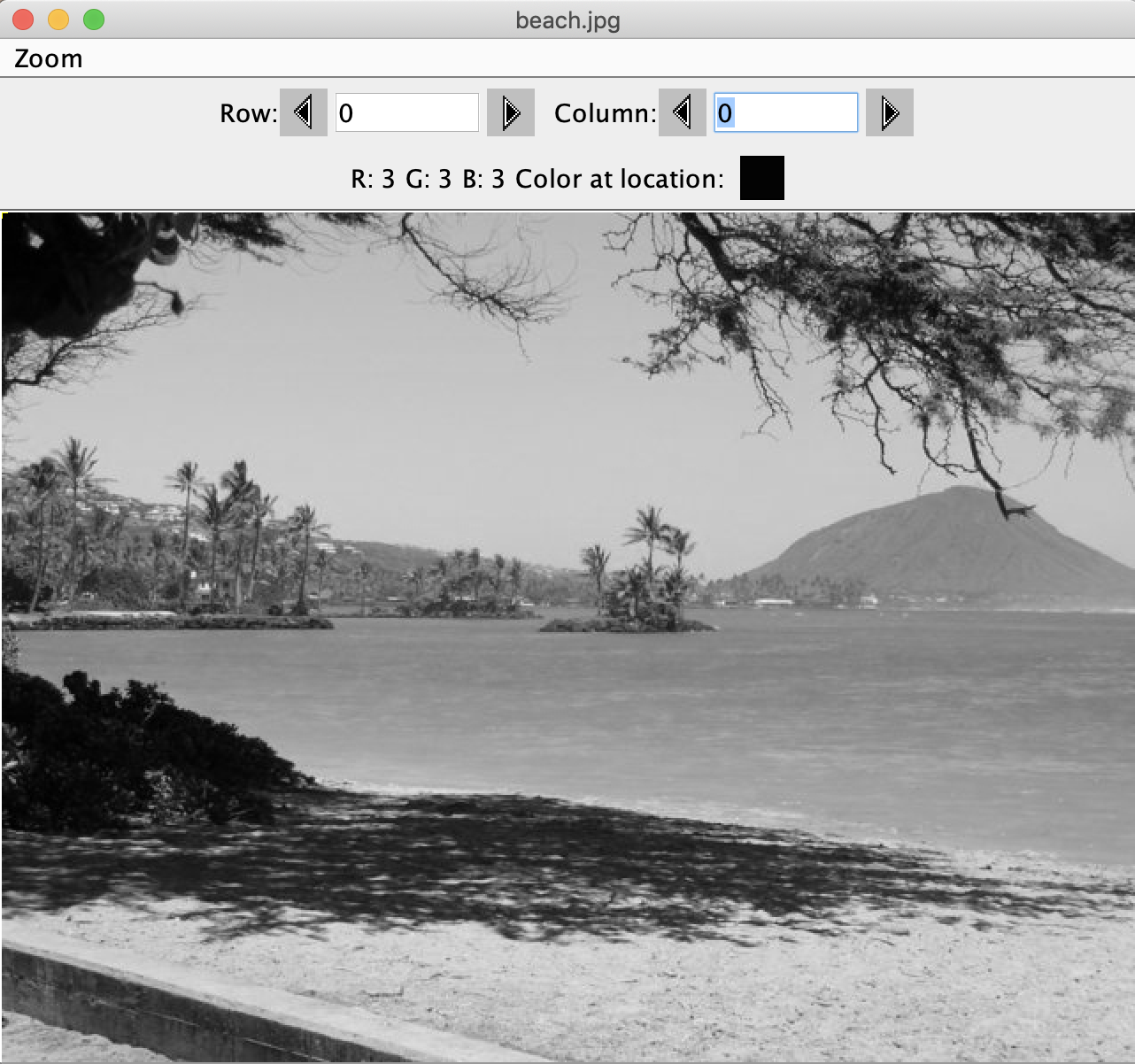
1. For the negate method, paste your code related to calculating and setting the values for red, blue, and green.

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

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

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

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.

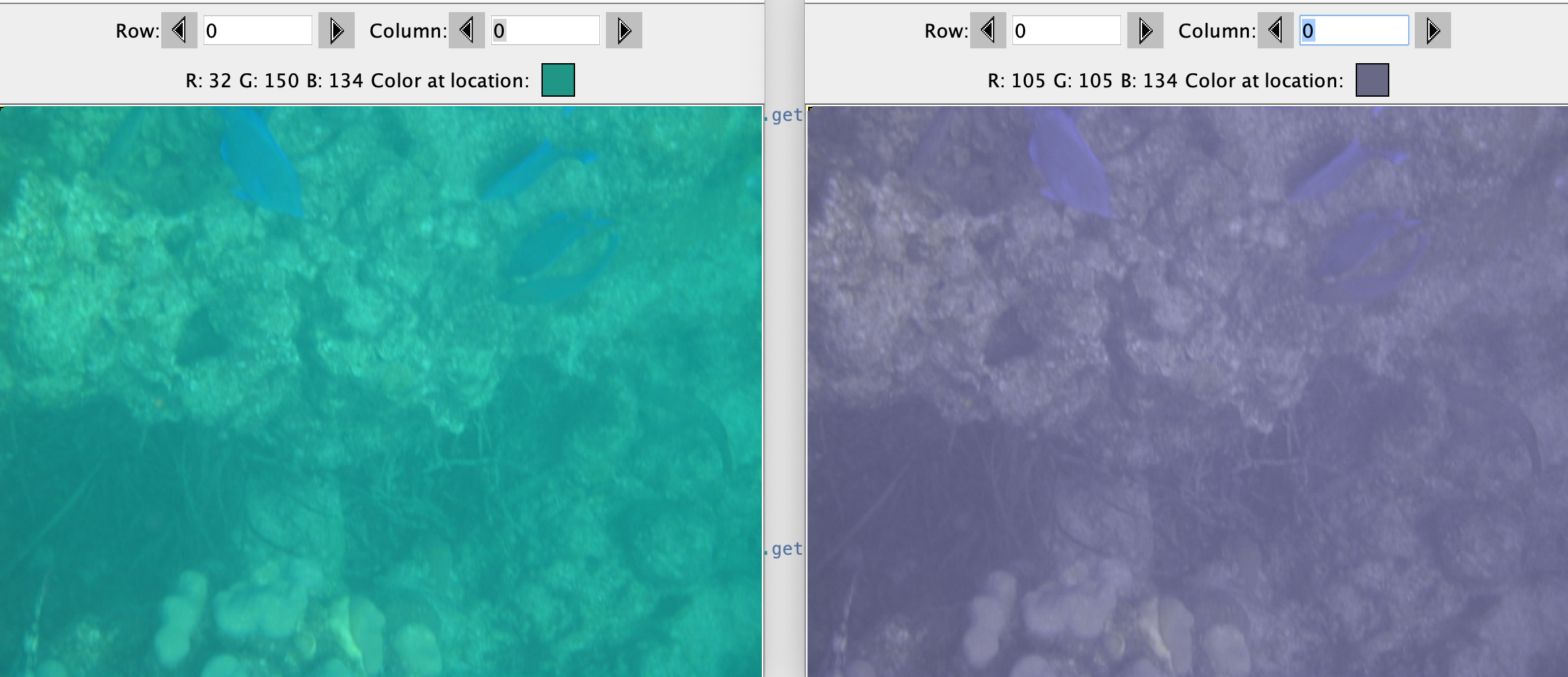
int sumPix = pixelObj.getRed() + pixelObj.getGreen() + pixelObj.getBlue();

int averagePix = sumPix/3;

pixelObj.setRed(averagePix);

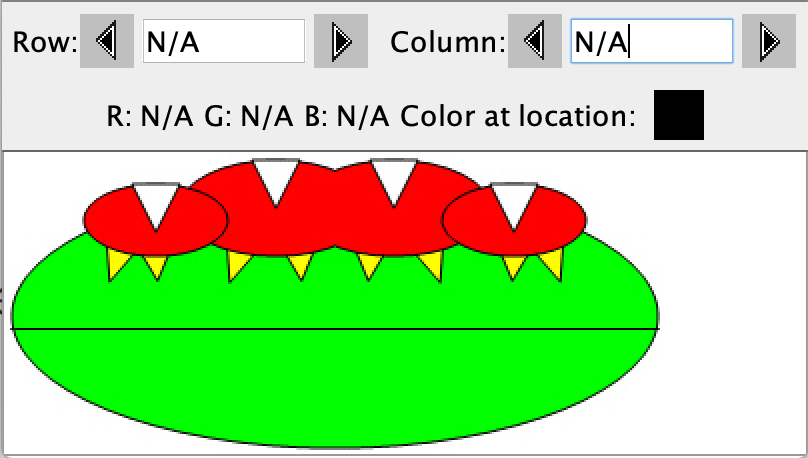
pixelObj.setGreen(averagePix);

this method kind of uses the grayscale method and alters the red and green pixels to the average pix. The blue pixel has the same value and red and green pixels are shifted. It is slightly easier to see the fish.



**Activity 6 Exercise Results**

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

public void mirrorVerticalRightToLeft()

{

Pixel[][] pixels = this.getPixels2D();

Pixel leftPixel = null;

Pixel rightPixel = null;

int width = pixels[0].length;

for (int row = 0; row < pixels.length; row++)

{

for (int col = 0; col < width / 2; col++)

{

leftPixel = pixels[row][col];

rightPixel = pixels[row][width - 1 - col];

leftPixel.setColor(rightPixel.getColor());

}

}

}

1. Describe the algorithm for the method mirrorHorizontal works.

public void mirrorHorizontal()

{

Pixel[][] pixels = this.getPixels2D();

Pixel upPixel = null;

Pixel downPixel = null;

int width = pixels[0].length;

for (int col = 0; col < width; col++)

{

for (int row = 0; row < pixels.length/2; row++)

{

upPixel = pixels[row][col];

downPixel = pixels[pixels.length - 1 - row][col];

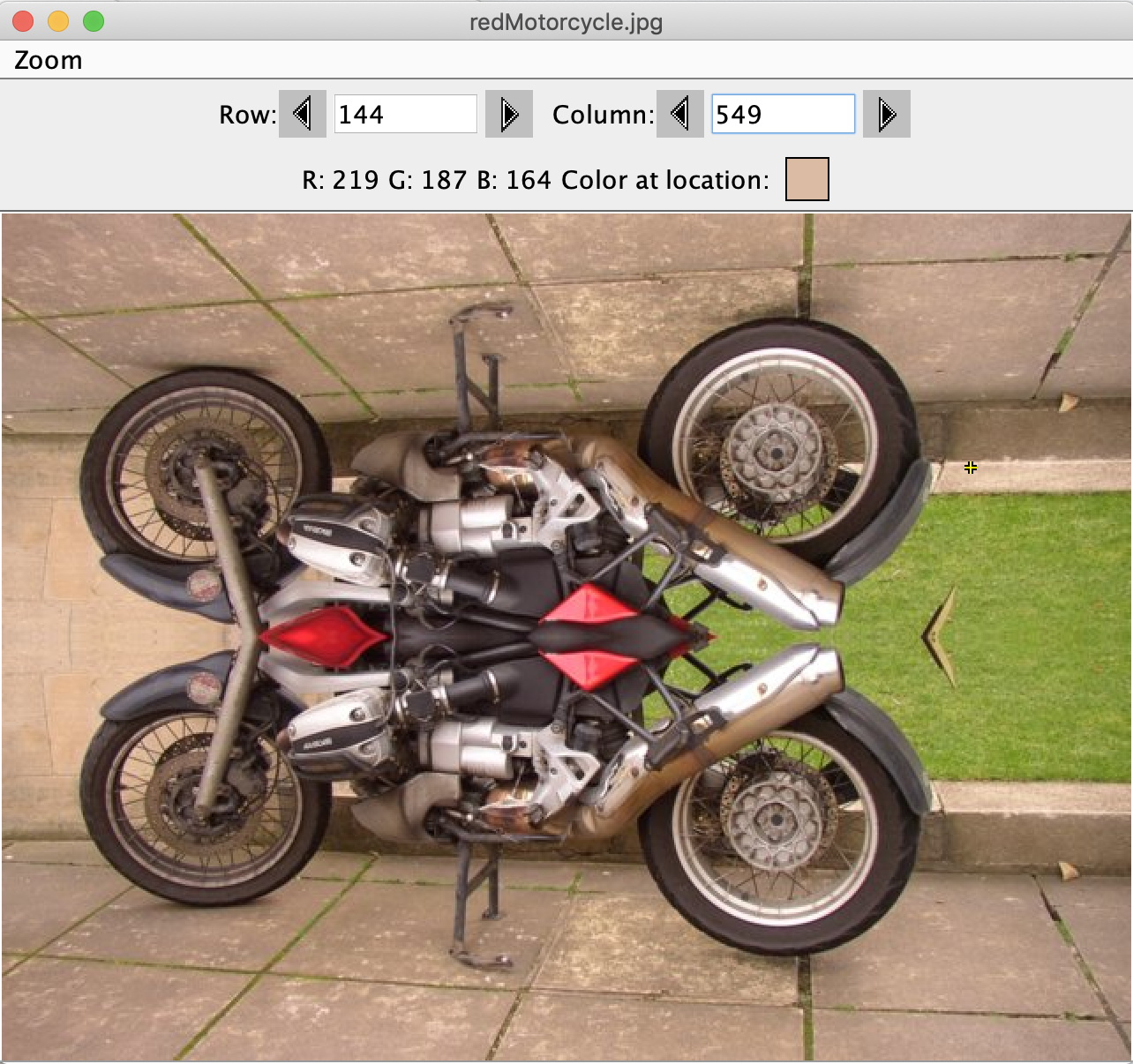
downPixel.setColor(upPixel.getColor());

}

}

}

This method uses a nested loop that loops the rows within the columns. As the columns go from 0 to the width, each row from the 0th row to half of the rows have an upPixel assigned and a down pixel assigned. The down pixel uses rows as **pixels.length - 1 - row** to mirror horizontally. Then downPixel is set to the color of upPixel. This gives the desired effect.

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

same as above, but the last line is changed to this:

upPixel.setColor(downPixel.getColor());

**Activity 7 Questions**

* 1. How many times would the body of this nested for loop execute? \_(17-7)(15-6) = 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? \_(11-5+1)(18-3+1) = 112\_\_

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? (97-27)(276-13) = 18410
2. Paste the image that is the result of calling the method mirrorArms.

public void mirrorArms()

{

Pixel[][] pixels = this.getPixels2D();

int mirrorPoint = 193;

Pixel upPixel = null;

Pixel downPixel = null;

//int width = pixels[0].length;

for (int col = 107; col < 298; col++)

{

for (int row = 165; row < mirrorPoint; row++)

{

upPixel = pixels[row][col];

downPixel = pixels[mirrorPoint - row + mirrorPoint][col];

downPixel.setColor(upPixel.getColor());

}

}

}

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

