**Picture Lab – Activity 6: Mirroring Pictures**

**Exercises -** Copy and paste the methods from the Picture class below once you have them working.

• Write the method mirrorVerticalRightToLeft that mirrors a picture around a mirror placed vertically from right to left. Hint: you can copy the body of mirrorVertical and only change one line in the body of the method to accomplish this. Write a class (static) test method called testMirrorVerticalRightToLeft in PictureTester to test this new method and call it in the main method.

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());

}

}

}

• Write the method mirrorHorizontal that mirrors a picture around a mirror placed horizontally at the middle of the height of the picture. Mirror from top to bottom as shown in the pictures below (Figure 8). Write a class (static) test method in PictureTester to test this new method and call it in the main method.

public void mirrorHorizontal(){

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

Pixel topPixel = null;

Pixel bottomPixel = null;

int height = pixels.length;

for (int row = 0; row < height / 2; row++)

{

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

{

topPixel = pixels[row][col];

bottomPixel = pixels[height - row - 1][col];

bottomPixel.setColor(topPixel.getColor());

}

}

}

• Write the method mirrorHorizontalBotToTop that mirrors the picture around a mirror placed horizontally from bottom to top. Hint: you can copy the body of mirrorHorizontal and only change one line to accomplish this. Write a class (static) test method in PictureTester to test this new method and call it in the main method.

public void mirrorHorizontalBotToTop(){

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

Pixel topPixel = null;

Pixel bottomPixel = null;

int height = pixels.length;

for (int row = 0; row < height / 2; row++)

{

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

{

topPixel = pixels[row][col];

bottomPixel = pixels[height - row - 1][col];

topPixel.setColor(bottomPixel.getColor());

}

}

}

Figure 8: Original picture (left) and mirrored from top to bottom (right)

• Challenge — Work in groups to figure out the algorithm for the method mirrorDiagonal that mirrors just a square part of the picture from bottom left to top right around a mirror placed on the diagonal line (the diagonal line is the one where the row index equals the column index). This will copy the triangular area to the left and below the diagonal line as shown below. This is like folding a square piece of paper from the bottom left to the top right, painting just the bottom left triangle and then (while the paint is still wet) folding the paper up to the top right again. The paint would be copied from the bottom left to the top right as shown in the pictures below (Figure 9). Write a class (static) test method in PictureTester to test this new method and call it in the main method.

public void mirrorDiagonal(){

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

Pixel pixel1 = null;

Pixel pixel2 = null;

int width = pixels[0].length;

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

{

for (int col = 0; col < pixels[0].length; col++)

{

if (col < pixels.length)

{

pixel1 = pixels[row][col];

pixel2 = pixels[col][row];

pixel1.setColor(pixel2.getColor());

}

}

}

}

Figure 9: Original picture (left) and mirrored around the diagonal line with copying from bottom left to top right (right)