**Mirror Vertical**

public void mirrorVertical() {

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

int halfIndex = matrix[row].length / 2;

for (int col=0; col<halfIndex; col++) {

Pixel numToReflect = matrix[row][col];

int newCol = matrix[row].length-col-1;

matrix[row][newCol] = matrix[row][col];

}

}

}

**Mirror Horizontal**

public void mirrorHorizontal() {

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

int halfIndex = pixels.length / 2;

for (int row=0; row<halfIndex; row++) {

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

int newRow = pixels.length-row-1;

Pixel originalPixel = pixels[row][col];

pixels[newRow][col].setColor(originalPixel.getColor());

}

}

}

**Mirror Horizontal Bottom to Top**

public void mirrorHorizontalBotToTop() {

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

int halfIndex = pixels.length / 2;

for (int row=pixels.length-1; row>halfIndex; row--) {

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

int newRow = pixels.length-row-1;

Pixel originalPixel = pixels[row][col];

pixels[newRow][col].setColor(originalPixel.getColor());

}

}

}

**Mirror Diagonal**

public void mirrorDiagonal() {

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

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

int limit = row;

if (row > pixels.length) {

limit = pixels.length;

}

for (int col=0; col<limit; col++) {

int newRow = col;

int newCol = row;

Pixel originalPixel = pixels[row][col];

pixels[col][row].setColor(originalPixel.getColor());

}

}

}

**Mirror Rectangle**

public void mirrorRectangle(int startX, int startY, int width, int height, int endMirrorX, int endMirrorY) {

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

int endX = startX + width;

int endY = startY + width;

for (int row=0; row < width;row ++) {

for (int col=0; col < height; col ++) {

int oldRow = startY + row;

int oldCol = startX + col;

int newCol = (width-col-1) + endMirrorX;

int newRow = row + endMirrorY;

pixels[newRow][newCol].setColor(pixels[oldRow][oldCol].getColor());

}

}

}