Vincent Lee
[redacted]

Color Swap Outline

- First we need to open file one for color swap.
- Second open file number two for the background.
- Next we input the input file into a stream by way of BufferedReader.
- Input the background file into a stream by way of BufferedReader.
- Once those two streams are open, we need to create a for loop that will sample each color and store it into an array, with pixel color, and pixel count.
- By using getRGB a color array with two columns will be created to keep a pixel count for the extra credit. The first column would be for the number or color value that comes out of getRGB, and the second column, a counter for how many times that particular color appears in the image.
- Color at (0, 0) will be set in the first slot of the color array, and the count will be set to one.
- Next a for loop will have to be written that will stop at pixel number (0, 1), and it will search through the color array column one to check if that color has already been registered for. If it has been registered for, the second column will have its count increased by one, to represent that color at pixel (0, 1). If not, the color array, will move to the next available spot and log the color value in column 1 and a one will be set for the count in the second column.
- The pixel information has all been logged. Now we need to find the color that has the highest count, most likely the background we would like to replace.
- This is done by, setting variable high to be the first row and second column in the color array. This sets the high value to the first row column two. Then a for loop can start at position 2 and if that number is higher, set high to that current position, and also log the color in a background color variable.
- Now the previous for loop should have given us a RGB value that we will use for color swap.
- The next step is to make a for loop for the amount of pixels in the row, and columns of the image. If pixel one equals background color, swap out the pixel for the same location as the original image. This is accomplished by setRGB.
- Continue doing this for loop until every pixel has been accounted for.
- Once this is finished, write the image to modified image, to a file with either extension PNG, or JPG. This is done by ImageIO.write.
- Close the opened file, and check the modified image for color replacement success!

-Vincent