CptS 122 Lab #8

In this lab, we will once again be working with PPM files. This time around, we will combine two images using a "green screen" effect. Without any constraints to the problem, performing a green screen overlay effect can be quite difficult as there are a number of factors that must be considered. For this lab, we will simplify the green screen effect's requirements in the following way:

1. We will assume that both the green screen image and the original image have the same dimensions.
2. We will assume that both images also have the same number of pixels per line. Note that the PPM specification doesn't say how many pixels are on each line of any given file.

Green Screen Algorithm

1. Let target_file refer to the original image.
2. Let key_file refer to the "green screen" image.
3. Let final_file refer to the combination of the original and green screen image.
4. Let key_color refer to the RGB value in the 0,0 column/row of key_file.
5. Simultaneously loop through each line in both target_file and key_file.
6. For each pixel in the line:
    a. Determine if the RGB value matches key_color. If there is a match, output the pixel from target_file into final_file. If there is not a match, output the pixel from key_file into final_file.

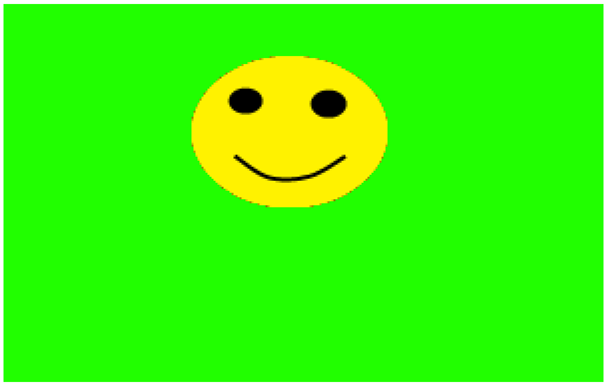
Implementation

For this lab, I'll leave the particulars of the lab's implementation up to you. However, I would recommend reusing the Point class from Lab #7 / HW #4.

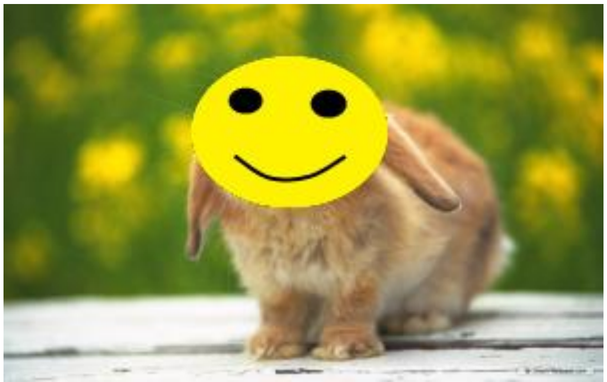
Sample Images and Output

We will use the bunny picture from HW4 as the target file:

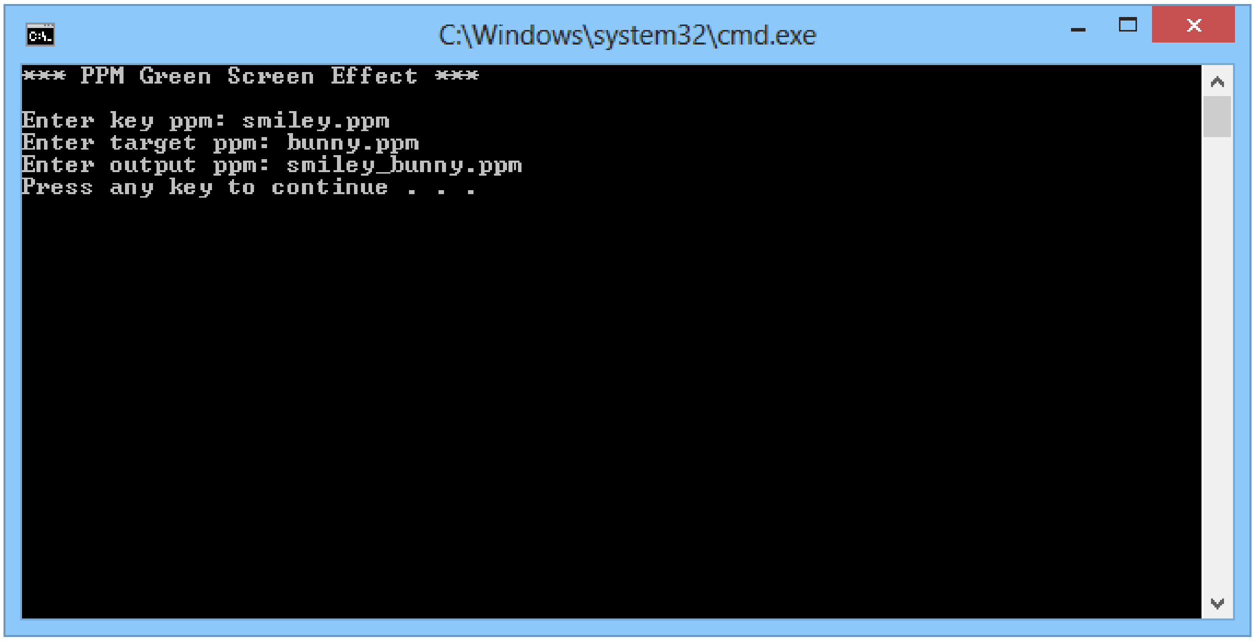
We will use my crudely-drawn smiley face as the key file:



Note that the green background in the above image serves as the key color. When combined, we get the following image:



Here is the output of my program:



```
*** PPM Green Screen Effect ***

Enter key ppm: smiley.ppm
Enter target ppm: bunny.ppm
Enter output ppm: smiley_bunny.ppm
Press any key to continue . . .
```