CMSC141: Object-Oriented Programming

Lab 8: Software Mirror

due: November 9th or 10th, 2017

This assignment asks you to create a series of image effects: a software photo-booth.

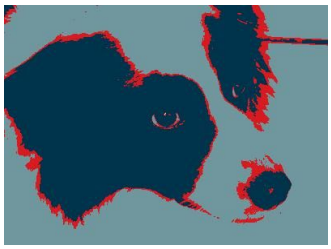
Warm-Up: Tinting a Photo

1. Download an image and use `loadImage()` to bring it into Processing.
2. Tint the image red using Processing's `tint()`.

Your "software mirror" sketch should load an image from a file or the webcam (see included code), and then based on user input, transform the image with different effects. Starting with the code provided you should trigger specific effects when the user presses a key. Each effect (including those provided) should be fully explained in comments. All the effects should operate at the pixel level and not use `filter`, `tint` or `blend`.

1. **brighten** – 'b' – adds a constant amount to each color channel, or scales each by some factor.
2. **dim** – 'd' – subtract a constant amount to each color channel, or scales each by some factor.
3. **swap** – 'w' – swaps the red, green, and blue channels.
4. **gray** – 'g' – turn the image into a gray scale by averaging the red, green, and blue channels.
5. **threshold** – 't' – turn the image into a binary, black-and-white image using a conditional statement.
6. **sepia** – 's' – applies a sepia effect, which raises the red and green channels, and lowers the blue. In particular it adds twice the `sepiaAmount` to the red, adds `sepiaAmount` to green, and subtracts `sepiaAmount` from blue. 20 is a good value for `sepiaAmount`.

7. **fairey** – 'f' – applies an effect similar to Shepard Fairey's iconic Obama "HOPE" poster. Each pixel is colored one of four colors depending upon the sum of RGB values. It assigns roughly equal intervals for each of the four colors.



rgb sum	Color
0 – 181	darkBlue (0, 51, 76)
182 – 363	red (217, 26, 33)
364 – 545	lightBlue (112, 150, 158)
545 – 765	yellow (252, 227, 166)

8. **your effect** – 'y' – an image effect of your choice (e.g. mirror, green-screen, slit-scan, glitch art, pixelation).

Learning Objectives

- ❏ Practice for loops.
- ❏ Analyze images at the pixel level.
- ❏ Work with image coordinate systems.

Deliverables

- ❏ Your program should start with a comment that includes your name, email, date, assignment description, collaboration statement, and reflection.
- ❏ Bring a hardcopy of your program (the source code, not the graphics) to your next lab period.
- ❏ Also turn-in the original textual design document.
- ❏ Be prepared to run the Processing sketch and demonstrate your "Theory of the Program."

```
import processing.video.*;
Capture video;

void setup() {
  size(640, 480);
  video = new Capture(this, width, height);
  video.start();
}

void draw() {
  if (video.available()) {
    video.read();
    video.loadPixels();
    for (int i = 0; i < width; i = i + 1) {
      for (int j = 0; j < height; j = j + 1) {
        int idx = j * video.width + i;
        color px = video.pixels[idx];
        float r = red(px);
        float g = green(px);
        float b = blue(px);
        if (key == 'i') {         // invert image
          px = color(255-r, 255-g, 255-b);
        } else if (key == 'r') {  // make image all red
          px = color(r, 0, 0);
        }
        video.pixels[idx] = px;
      }
    }
    video.updatePixels();
    image(video, 0, 0);
  }
}
```