Challenge: Rainbow Filter

As a challenge project, add a button to give an image a rainbow-colored filter. Try creating seven horizontal stripes, evenly spaced across the image, and give each one a different hue filter, such as red, orange, yellow, green, blue, indigo, and violet.

For example, here is the lion.jpg image with a rainbow filter.

Rainbow filter algorithm

For each pixel in the image, first calculate the average of the pixel's RGB values, then find the appropriate formula in the table:

Color	avg < 128	avg >= 128
Red	R 2*avg G 0 B 0	R 255 G 2*avg - 255 B 2*avg - 255
Orange	R 2*avg G 0.8*avg B 0	R 255 G 1.2*avg - 51 B 2*avg - 255
Yellow	R 2*avg G 2*avg B 0	R 255 G 255 B 2*avg - 255
Green	R 0 G 2*avg B 0	R 2*avg - 255 G 255 B 2*avg - 255
Blue	R 0 G 0 B 2*avg	R 2*avg - 255 G 2*avg - 255 B 255
Indigo	R 0.8*avg G 0 B 2*avg	R 1.2*avg - 51 G 2*avg - 255 B 255
Violet	R 1.6*avg G 0 B 1.6*avg	R 0.4*avg + 153 G 2*avg - 255 B 0.4*avg + 153

How to make a filter in any color

The values in the table were determined by plotting a piecewise linear function for each of the R, G, and B values vs. the average value. To keep black pixels black and white pixels white, the desired functions are pinned at (0,0) and (255,255). The third point (where the function changes slope) is determined by plotting the R, G, or B value of

the desired color vs. 127.5.

Consider any color any color with R value = Rc, G value = Gc, and B value = Bc. The filtered pixel has R value:

```
R = Rc/127.5*avg                      for avg < 128
    (2 - Rc/127.5)*avg + 2*Rc - 255   for avg >=128
```

Similarly for G and B, where you would substitute Gc or Bc for Rc in the above formula.

To apply this formula and create a colored filter of your choice, use a color picker tool to determine the RGB content of any color you would like to use, such as teal (17,170,153).

Since for teal, Rc = 17, Gc = 170, Bc = 153, so

```
R = 17/127.5*avg                      = 0.13*avg        for avg < 128
    (2 - 17/127.5)*avg + 2*17 - 255   = 1.87*avg - 221  for avg >=128

G = 170/127.5*avg                     = 1.33*avg        for avg < 128
    (2 - 170/127.5)*avg + 2*170 - 255 = 0.67*avg + 85   for avg >=128

B = 153/127.5*avg                     = 1.2*avg         for avg < 128
    (2 - 153/127.5)*avg + 2*153 - 255 = 0.8*avg + 51    for avg >=128
```

Now you can use this formula to make a filter in any color!

Challenge: Blur Filter

One way to create a simple blur filter is to scramble pixels that are near each other. For example, you could turn this image:

into this image:

Let us consider how to achieve this effect.

We begin by creating a blank image. For each pixel we will do one of two things: half the time, we will simply copy the pixel from that location in the old picture. The other half of the time we will find a pixel nearby and copy that pixel instead. We will do this by generating a random number between 0 and 1. If the random number generated is less than 0.5 (which it will be approximately half the time), we will copy the pixel from the old picture. Otherwise, we will find and copy a nearby pixel.

We must figure out how to find a "nearby" pixel. We will define some value for how far away the new pixel will be (say, 10 pixels) and then we write a function that will give x and y coordinates that are a random amount between 0 and 10 pixels away. For example, it could give a pixel that is 5 pixels to the left and 3 pixels higher.

Before we use these coordinates to get a new pixel, we must check that the new coordinates still give a valid pixel in the image. For example, imagine we are finding a pixel to replace a pixel at the very top of the image. The function that generates coordinates gives us a point that is 3 pixels up, but since we are on the top of the image (y = 0) we cannot go up by three pixels (y would be -3)! If the random number is too big (larger than the width-1 or height-1) or too small (less than 0) then we will just use the closest number that is valid.

Once we have a valid pixel that is some amount away, we use its red, green, and blue values as the new pixel's values.

Note about CSS Filters

Some image filters can be implemented in CSS, using the CSS filter property (see http://www.w3schools.com/cssref/css3_pr_filter.asp). For example, you can make an image grayscale using CSS. We have shown grayscale in JavaScript to give an idea of what types of things you can do with programming. JavaScript also allows you to go far beyond the filters offered by the CSS filter property and build or customize a filter to do exactly what you want it to.

标记为完成