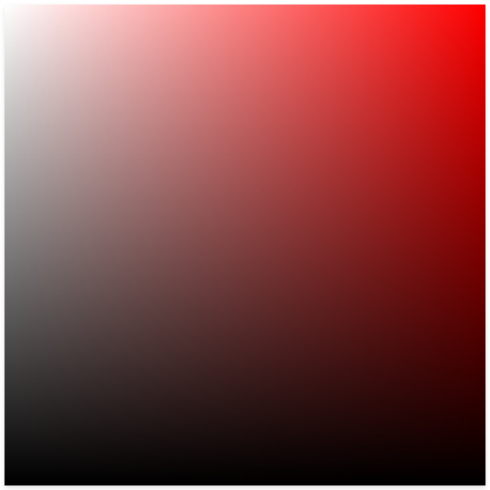
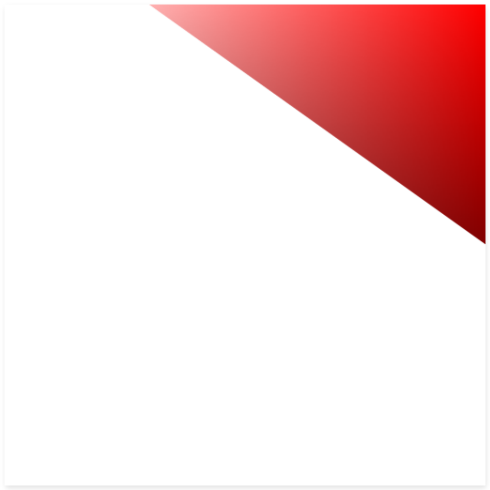
Finding Key-Color Boundaries

This week I wanted to work on finding the boundaries on what defines a “key-color”. In english, I described a key-color as a colorful color, or one that is both bright and saturated. At first, just to try things out, I used the HSB (Hue, Saturation, Brightness) colorspace. I defined the key-color boundary as a linear line relating brightness and saturation (regardless of hue). The test is defined as:

```
self.brightness + ((5.0f/7.0f) * self.saturation) <= (17.0f/14.0f)
```

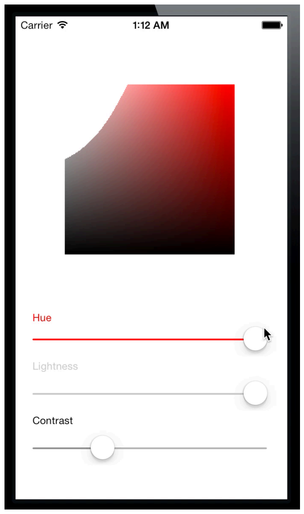
This divides the HSB colorspace in two, key-colors and not key-colors.


All Colors in HSB Colorspace (Where H=0º)

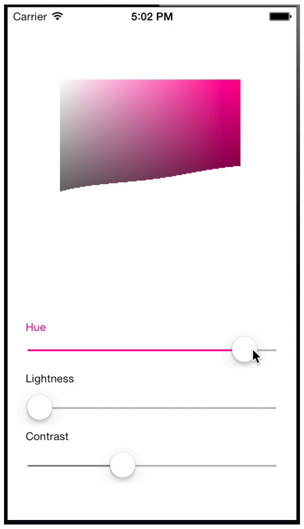

Key-Colors in HSB Colorspace (Where H=0º)

Going into this, I knew it wasn’t going to work perfectly because certain hues, like yellow, appear lighter than others, like blue, even at 100% lightness and 100% saturation in this colorspace. I later learned there is a term for this effect and it’s called the Helmholtz-Kohlrausch effect. While I thought taking advantage of the CIELAB colorspace would solve this issue, but it does not. However, the CIELAB colorspace does help out quite a bit with numerous other issues. For determining a key-color, I am going to start with the CIELAB colorspace and then adjust myself. I have not been able to find great information on solving (accounting for) the Helmholtz-Kohlrausch effect.

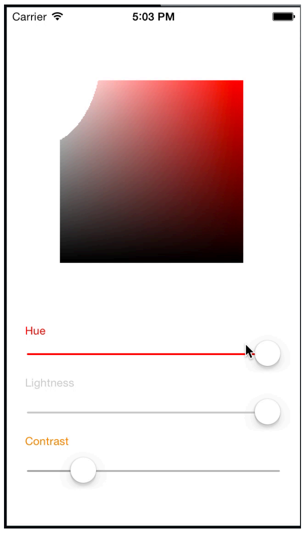
To see how CIELAB handles my definition of a key-color, I made a demonstration app. This app has three sliders, hue, lightness, and contrast as well as one HSB colorspace viewer. The hue slider controls which hue is displayed on the HSB colorspace viewer. The lightness slider controls the grayscale value which is used for contrast comparison (left being black and right being white). The contrast slider controls how much contrast is required from the greyscale value in order for the color value to be drawn.



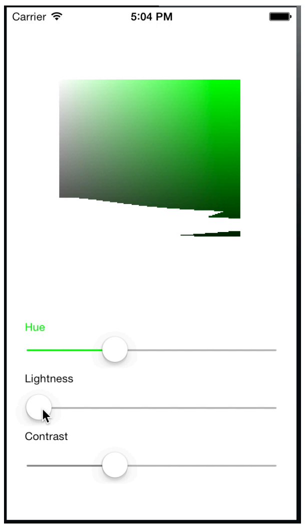
Demonstrating the Key-Color Tester Application



Changing hue to contrast against black



Changing hue to contrast against white

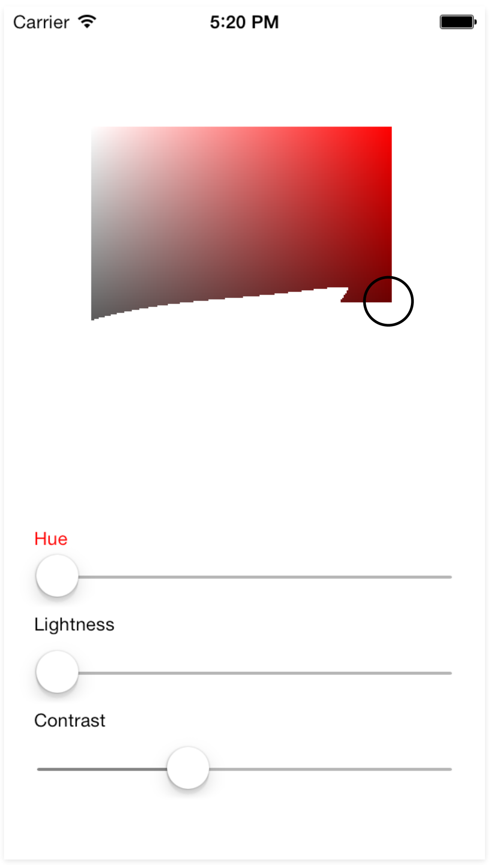


Changing greyscale color to contrast against

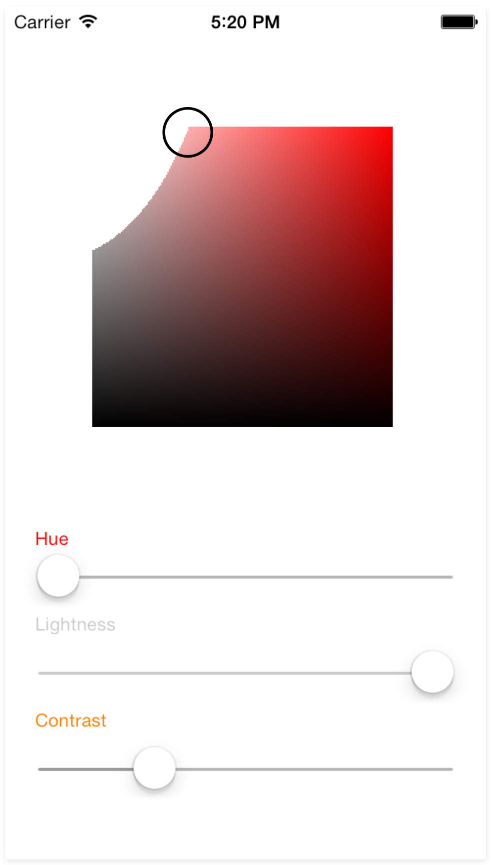
After playing around with the sliders to get an understanding of the relationships between the components, I wrote code to get two pieces of information out of this:

I. The relationship between hue and the minimum saturation point at 100% brightness with a specified contrast against white and the relationship between hue.
II. The minimum lightness point at 100% saturation with a specified contrast against black.

This sounds complicated but it's best when shown visually. I want to find the two circled intersection points for each hue value.

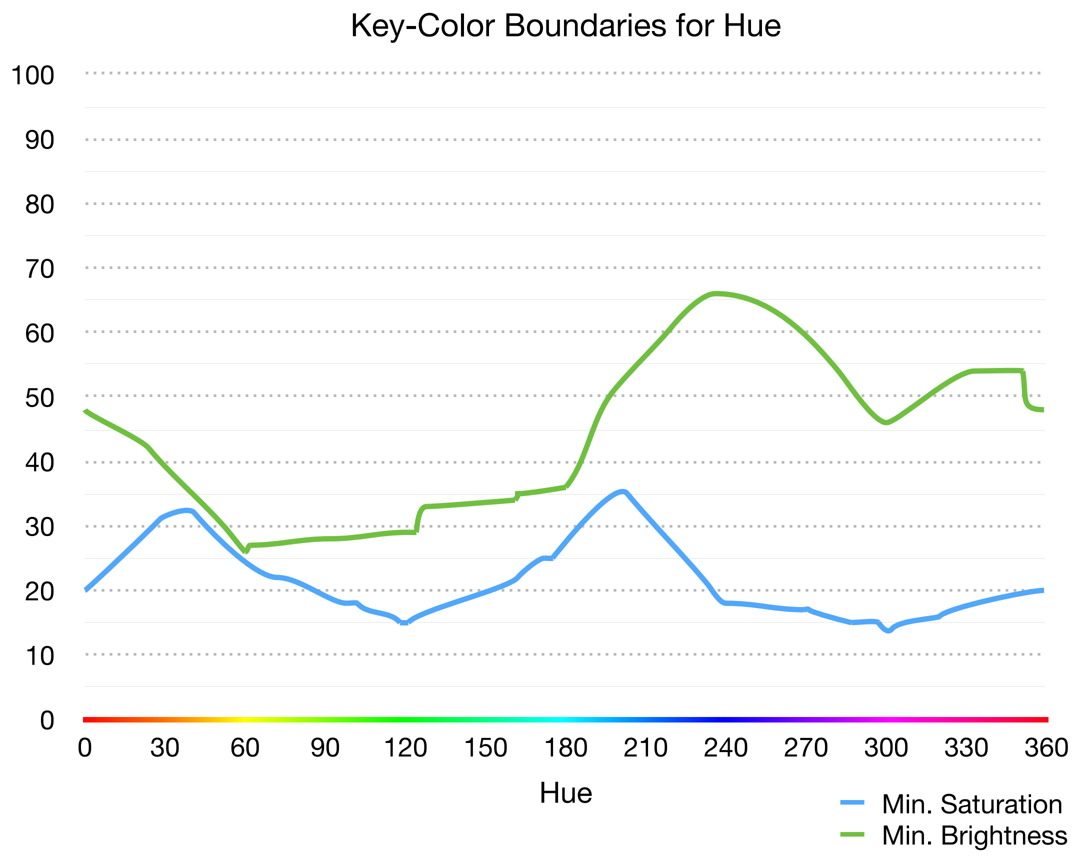


The minimum lightness point
(contrasts against black)



The minimum saturation point
(contrasts against white)

I found these two points for 360 degrees of hue and graphed them.



The minimum saturation is calculated based on the specified minimum contrast from RGB white. The minimum contrast is calculated based on two times the specified minimum contrast from RGB black. The minimum contrast shown here is 18.4. The contrast (color difference) is calculated using the CIEDE2000 algorithm.

This shows that colors like yellow appear brighter and therefore require more saturation to appear visually distinct against white. Blue appears darker and needs more brightness to appear distinct from black.

What do I do from here?

I have to find a mathematical equation to estimate the two lines shown in the graph. I will then use this to display boundary colors. I then may manually tweak the results so that all boundary colors contrast well enough against a white background and don’t appear too dark (like black).

After that, a curve needs to be drawn between these two points. I ideally would like to express this in the CIELAB colorspace instead of on the HSB colorspace.