Lab 12: Color Palette Generator DevLog

Pluto Zitek – https://github.com/GemedetAdept (https://github.com/GemedetAdept)

<div align="center">

Goals

</div>

✓ Fix incorrect conversion function implementation

    ✓ RGB to HSL

    ✓ RGB to HSV

    ✓ HSV to RGB

❏ Implement remaining conversion methods

    ❏ HEX

        ❏ HEX to HSL

        ❏ HEX to HSV

        ❏ HEX to RGB

    ❏ HSL

        ❏ HSL to HEX

        ✓ HSL to HSV

        ✓ HSL to RGB

    ❏ HSV

        ❏ HEX

        ✓ HSL

    ❏ RGB

        ❏ RGB to HEX

Implementation and Progress

Using Tuples vs Using Objects for Color Data

Given the scope of the course curiculum, I've opted to use tuples to store the color data rather than creating constructors for each color mode.

Fixing Broken/Incorrectly Implemented Formulas

I believe there are two issues which are causing the conversions between RGB and HSL/HSV to break:

1. An incomplete and/or incorrect understanding of the formulas

2. Ambiguity when referencing the written formulas, due to the use of single-letter variables

So far, I have resolved the second issue by seeking out multiple sources for the formula. Although they largely still used the same, single-letter variables, it was helpful to have it explained in different words. I would like to have the time to dig into the theory behind the formula(s) to resolve the first issue, but I am focusing on making the program work first.

Lingering Formula Bugs

I do not know why (for the same reason as bullet point 1, above) but the piecewise formula for getting the Hue value for HSV/HSL from RGB (Fig. 1) does not implement cleanly. When `max(R,G,B)` is equal to the value of `R`, the formula for hue is $60° \cdot \left(0 + \frac{G-B}{C}\right)$. However, when implemented into code (Fig. 2), this formula return the exact *inverse* of the correct value. This is easily solved by adding 360 to the value, but I would like to fix the formula itself if I get the time.

Fig. 1.

$$H := \begin{cases} 0, & \text{if } C = 0 \\ 60° \cdot \left(0 + \frac{G-B}{C}\right), & \text{if } V = R \\ 60° \cdot \left(2 + \frac{B-R}{C}\right), & \text{if } V = G \\ 60° \cdot \left(4 + \frac{R-G}{C}\right), & \text{if } V = B \end{cases}$$

Fig. 2.

```
else if (maxRGB == redPrimeRGB) {
 // I do not know this is producing the inverse of what it should, but un-inverting it makes it work, so.
 hueHSL = 60*(0 + (greenPrimeRGB - bluePrimeRGB)/chroma);
 hueHSL = 360 + hueHSL;
}
```

Remaining Tasks

Chad is writing the methods for converting between base-16 and base-10, so I was not able to fully implement the methods for converting between HEX and RGB/HSL/HSV. For the time being, I will write an outline of the required conversions and use a placeholder variable which can be replaced later.

Last updated 2022-12-01 17:25:00 -0700