Drayton Redick
Nicholas Foster
Lawson Willard

Chapter 1

The project we chose was Tanaguru Contrast Finder. The Contrast Finder is a useful tool for designers, as it takes two colors inputted by the user, either with RGB or hexadecimal values, and determines whether or not the colors provide a good contrast to each other. The most interesting part, however, is what happens when the system determines the colors do not match. Automatically, the program produces a list of suggested changes to the colors to produce a more visible and easily accessible visual style. The user may select whether they want the foreground or the background to be edited, and if they want a few color suggestions or many. For example, if a user were to have bright yellow foreground text and a white background color, and that user has the "edit foreground" option selected, Tanaguru Contrast Finder will produce a list of better color options with a visual sample for each item. In this case the contrast finder might suggest darker colors such as black or navy blue to better contrast with a white background. This program is a very useful tool for creating websites and UI alike.

While trying to build the program on our machines, we ran into a slight problem. As it turns out, the contrast finder depends on many different libraries. It would seem like everytime we attempted to run the program we would encounter yet another error telling us that a library was missing. Fortunately, Tanaguru provides an in-browser demo that we could test while we were downloaded various libraries. The intended goal of the program is relatively simple, and so testing was as well. We used numerous color combinations to test the system's limit, but it would always deliver on its promised result. Colors that did not contrast would give us a large amount of superior options to work with, and using colors that already have good contrast simply printed an “everything is ok!” message.

Overall testing included bombarding the system with a number of ridiculous color combinations to really push its limits. While the system has no functionality to fix tastes, it never failed to produce more accessible and readable results. Our testing, as well as the test cases provided on Tanaguru’ Github, failed the stump the system. It would seem that the Tanaguru Contrast Finder has some pretty airtight code, provided that you have the proper libraries installed, that is.