Colour-Chisel

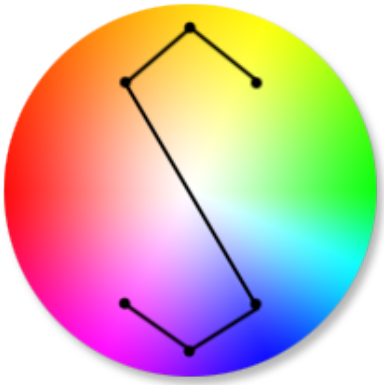
CPSC 448 Project Proposal by Christopher Powroznik

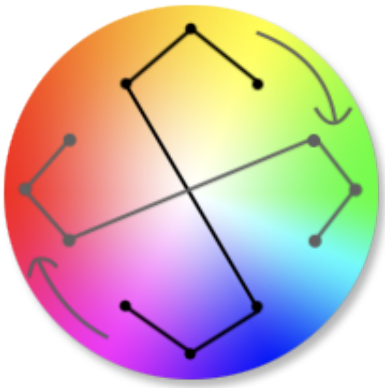
Overview

This is my submission as a proposal for CPSC 448 Directed Studies. My proposed project is to create a language and library for navigating the higher dimensional space of the colour wheel in order to determine colour patterns and schemes. I intend to work under the supervision of Elisa Baniassad as I believe her expertise in DSL and language design will be greatly beneficial, and also she is a great person.

Background

Colour scheming is a common process in nearly any front end application being built. Choosing colours requires experience and a lot of intuition. Many developers will choose a colour scheme of 5 or 6 colours and build applications with the idea of primary, secondary and tertiary in mind. However, many more experienced UI/UX designers will define large colour palettes based movements on the colour wheel. For example, defining an ‘S’ on a colour wheel using 6 points gives a good representation of 2 complementary colours with an adjusted hue for each. Manually making these shapes and defining colour schemes is tedious, especially if the shape wasn’t simple like an ‘S’. The ability to create these shapes and also provide transformation tools will allow developers to experiment and design more dynamic colour schemes.





Goals

1. **Create a functional DSL for representing shapes and operations on the colour wheel**
2. **Create a pattern creation tool for developers to programmatically choose colours**
3. **Create a library that interprets and compiles the language simply for developers**
4. **Design a new file type and syntax that will compile down into javascript or SCSS**
5. **Package as a usable npm module**
6. **Create an online tool for people to compile in their browser without dependencies**

Specifications

I want to create a language for the colour wheel and give access to common tools like creating relative colours (compliments, analogous), defining shapes and transforming these shapes to adhere to different palettes. I also want to create this as a language alongside a library because not all UI/UX designers have JS experience. My language would be simple to learn and implement into projects without the overhead of learning javascript, but still provide the functionality of direct implementation if needed.

Milestones

1. Create an EBNF

   Construct EBNF and define the functionality of the language. This will be the blueprint for the implementation process.

2. Initialize Project Structure and Test Suite.

   Initialize my environment with continuous deployment for the npm package upon release and integrating a testing suite.

3. Create interface for library and use it to create tests.

For the javascript library, define all the different functions in an interface. I will write a test suite using this interface before implementation, where I have manually done all the math to compare against easily for when I start coding the library.

4. Implement language directly into runtime compiler.

   The language will initially be called via passing statements into a runtime compiler as so: colourChisel(“Program”) => [#hex, ...]

5. Get Feedback and adjust library.

   At the first working implementation, I want to do use case testing with the project and get feedback from various sources in order to solidify the functionality and vision of the project.

6. Extend Library for procedural usage.

   For the javascript implementation of the library it will be intuitive to provide a procedural interface for developers to programmatically use the language as JS functions. An early example would be the following:

   colourChisel(“#hex”).analogous(30).compliment()=> #hex

   colourChisel.shape(“#hex”, “#hex”, colourChisel(“#hex).compliment()).rotate(-20) => [#hex, #hex, #hex];

7. Make compilable file to JS.

   I will create my own file type that can be compiled down to give exports from a js file of each of the colours to be used.

8. Make compilable file to SCSS.

   I will also create an export to SCSS to be used in creating more robust styles.

9. Get Feedback and adjust library.

At this point I want to conduct more user testing to start finalizing my project's release, by gathering feedback and making changes to my language and library.

10. Make online tool to create JS and SCSS exports without a dependency.

    I will provide an online tool where users many write the language into a code editor and see an export into JS and SCSS to be copied and pasted and used directly in a project without installing the library as a dependency.

11. Release as npm module.

    I will package the entire library as an npm module and release it.

12. Syntax highlighting (Stretch Goal).

    If time permits I will also create a syntax highlighting for when writing my language in code editors.

Deliverables

Language	NPM Module	Website Tool	Validation Report
I will provide a fully implemented library with compiler.	I will package the language in a javascript library and release it on npm.	I will create a web tool for users to write into an online code editor compile into js and scss.	I will do a report on my findings, learning experience and feedback from users. I will also provide a detailed overview of my journey creating this project.

About Me

I'm currently in my **last semester** at UBC completing a combined major of Business and Computer Science (BUCS). I will be done taking courses in December and I really want to get the chance before I leave to learn about creating releasable products that benefit developers

everywhere. I believe I have the pre-requisite experience from CPSC 310 and previous work experiences to build a language, library and web application. I really want to experience developing a solo project from beginning to end without a boilerplate as none of my software engineering courses have offered this thus far. I believe this would be a valuable learning experience for starting my career in the new year and a high note to finish my undergrad on.