A Thesis Paper On

COLOR CONTRAST PROCESSOR AND COLOR SCHEME GENERATOR

By

Dhanad Islam
Student ID: 03101077

Department of Computer Science and Engineering
December 2007

DECLARATION

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of Supervisor

Signature of Author

ACKNOWLEDGEMENTS

Special thanks to my supervisor Abdussamad Ahmed Muntahi who encouraged me to pursue my work and guided me throughout the whole process of my research and development and to think out of the box. I would like to convey my thanks to Matin Saad Abdullah for inspiring me to work with colors initially, to my peers for believing in me and for their support and to the member of the advisory committee Dr. Mumit Khan for taking time out of their busy schedules to consider this work.

ABSTRACT

Color contrast processor and color scheme generator is an integrated tool developed to aid user in finding a set of colors that have a notable level of contrast and brightness between them, and to help find a combination of colors that are harmonious to any user defined color using concept of Color Theory. It is merely a tool that computes user inputs (primarily color) and provides analysis and alternative suggestions and choice of colors.

This application was designed with the purpose of but is not limited to helping web and software developers/designers. It comprises of 3 primary features, namely the Color Contrast Analyzer which analyzers the contrast between any two colors and suggest alternative (if any) choices. The second part of the tool is the Color Scheme Generator, which based on user input color produces a list of other colors that are harmonious or said to be a good combination of colors to each other. The last part includes the Color Scheme Library which consists of predefined color schemes and custom saved schemes for quick reference that can be used by the user.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

In the world of technology and multimedia, the demand for high quality and reliable solutions is always the top priority. Whether the solution concerned is desktop, web-based or mobile integrated; the need for appealing and professional outlook on any of these interfaces is one of the most essential elements. Often than not, a good product is negatively received by the end user due to its usability and readability issues, overall appearance and often due to bad color choices.

1.1 Background

Web-based solution or desktop, needless to say it is important that the developers and designers should follow proper design guidelines and meet some standard specifications. What we need is a tool that can help the developers/designers in the following ways:

1. Assess user choice of colors and suggest if they meet some standard guidelines or not.
2. Suggest alternative choices to the user.
3. Suggest a list of color combination based on a user defined base color.
4. Library containing predefined color schemes and user generated schemes which can be used later for reference, etc.

Fortunately there are plethora of useful tools, guidelines and applications available but not organized or assembled in a way that can be useful for the end-user, in this case the developers and the designers.

1.2 Current Tools

Throughout my research I have managed to come across a variety of the tools that can to some extent aid the end-user I had in mind. Some of the most required elements were either missing in one or the other, but it is worth mentioning these wonderful applications that formed the basis of my thesis topic.

1. Color Wheel Pro [1]
2. Color Schemer Studio [2]
3. Color Combos [3]
4. WORQX [4]
5. Accessibility Color Wheel [5]

1.3 Thesis Objectives

An important discovery was made during analysis of the current applications, i.e. everything is already available but not integrated into one-stop solution and it seemed most tools were not aimed at any specific end-user or type. My thesis topic was then directed towards producing a solution to only a specific category of end-user, i.e. the designers and web-developers. This is where the ***Color Contrast Processor & Color Scheme Generator*** comes into the picture.

1.3.1 Primary Objectives

The following modules were targeted as primary objectives of my thesis, later integrated into the ***Color Contrast Processor & Color Scheme Generator*** application.

1. Color Contrast Analyzer
    - Process any two given colors and suggest whether they are contrasting enough or not.
2. Color Scheme Generator
    - Based on user input color (base color) generate all other color combinations that goes well with the base color.

1.3.2 Secondary Objectives

Secondary goal was to develop an automated feature built into the ***Color Contrast Processor & Color Scheme Generator*** for looking up pre-defined or custom saved color schemes created by the end-user and being able to read from various sources like images and analyze its color contents. The following sections were the sub goals:

1. Color Scheme Library
    - Inventory for referring to predefined color schemes and previously user created schemes.
2. Color Information Extraction & Analysis from various sources
    - Extract color information from sources like, Flash or Adobe Illustrator objects, or even web-pages. Then analyze the information to suggest alternative color choices.

CHAPTER 2

SYSTEM ANLYSIS & DESIGN

System design became the next phase of the thesis project, where important elements such as objectives and scopes were required and needed to be understood in full before proceeding any further.

2.1 System Design Overview

A design of the overall system was made from the specifications described in section 1.3. The figure below shows a top down view of the complete system.

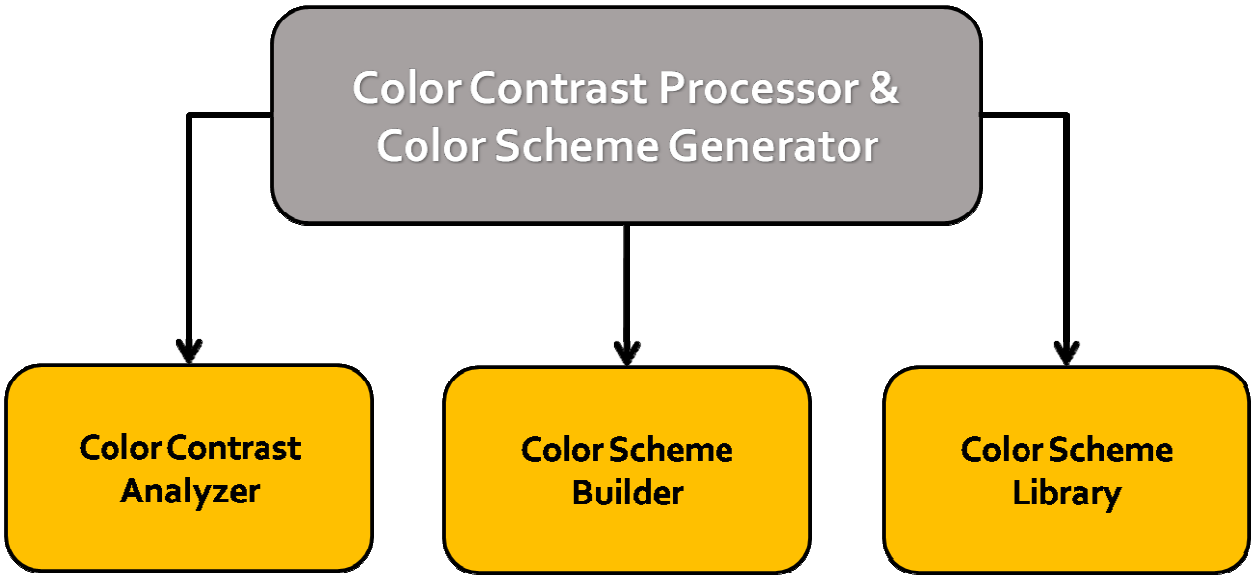

Figure 2.1 - System Design of Color Contrast Processor & Color Scheme Generator

2.1.1 Color Contrast Analyzer

This module as mentioned in the thesis primary objective is responsible for analyzing contrast between two colors specified by the user. Additional function of this area is to suggest any alternative choice of colors that is also contrasting enough against any of the selected input color. The diagram below describes the functionality of ***Color Contrast Analyzer*** module.

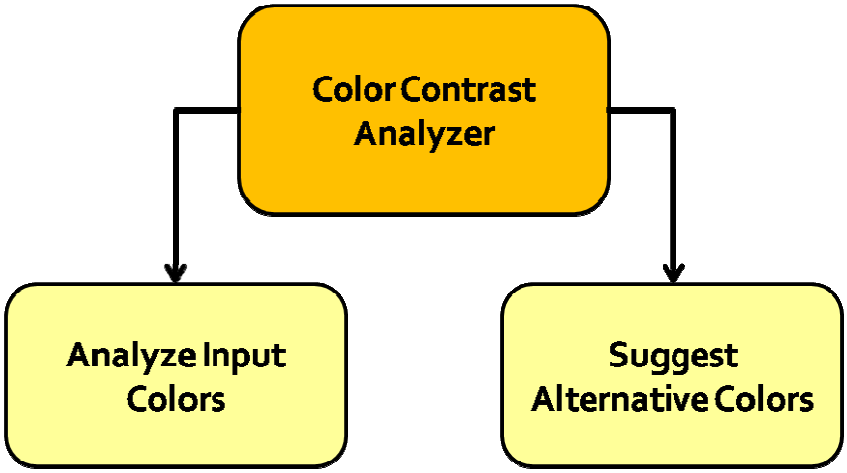

Figure 2.2 - Subsection functions of the Color Contrast Analyzer module

2.1.1.1 Analyze Input Colors

User will give two input colors, one as foreground color and the other as a background as shown in the diagram below.

2.1.1.2 Suggest Alternative Colors

In this section user is given a list of all possible combination of alternative colors that also have a good contrast against a certain input color. For example, if the user wants to list all possible alternative colors instead of the blue (in section 2.1.1.1) that can be used against a white text, then this part will compute using some algorithm (see algorithm 3.1) the list for the user.

2.1.2 Color Scheme Builder

This module is responsible generating color schemes and combinations for the user based on Color Theory (see section 4.2). Not only does this part generates a list of colors that are said to be good combinations against a user selected color (base color), it also lets the user pick any 5 colors from the generated list and save it as a custom favorite scheme for later reference which can be found in the list entry under the ***Color Scheme Library*** module.

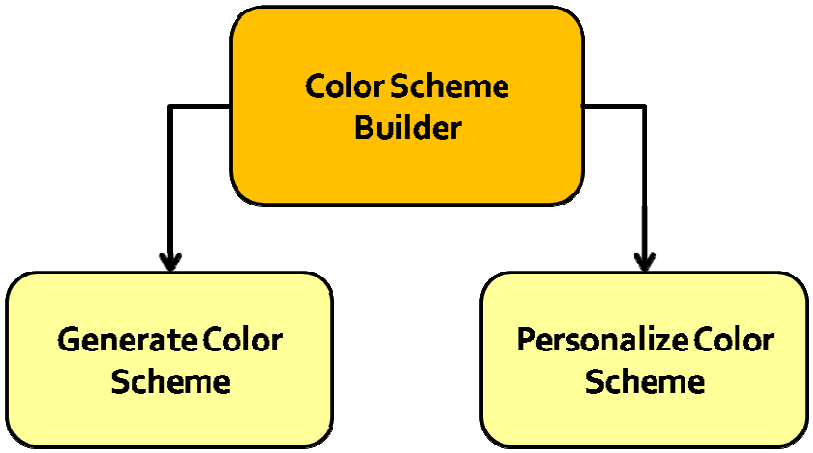

Figure 2.3 - Sub functions of Color Scheme Builder

2.1.2.1 Generate Color Scheme

A color scheme will be created based on what color the user has given and what type of scheme has been selected. There are several basic scheme types derived from the Color Wheel (see chapter 4).

2.1.2.2 Personalize Color Scheme

This is where the user can pick any 5 colors from the generated list of colors produced by the ***Generate Color Scheme*** function (see section 2.1.2.1). The picked color can then be saved for later reference found in ***Color Scheme Library***.

2.1.3 Color Scheme Builder

This is the last section of the application where we can find all the predefined color schemes and also user saved schemes (if any). This is basically an inventory system dedicated to aid user in looking up already available schemes, or retrieve color values of any certain color from any chosen scheme.

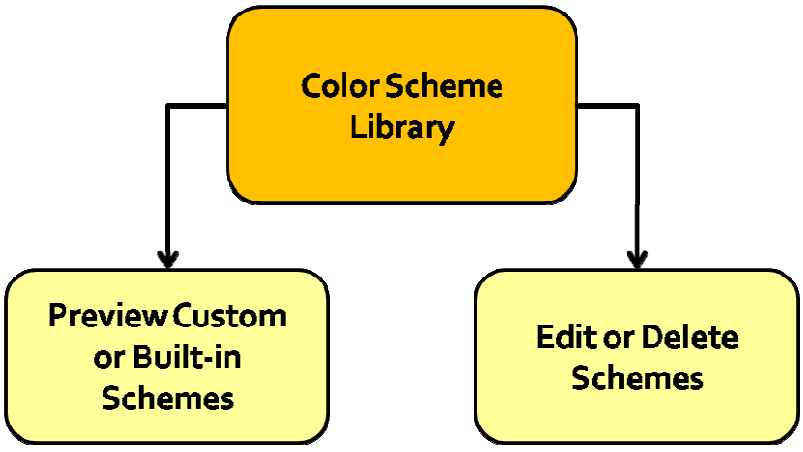

Figure 2.4 - Sub functions of Color Scheme Library

2.1.3.1 Preview Custom or Built-in Schemes

This is where all the saved schemes, i.e. built-in or custom are listed for reference. User can preview the schemes by selecting them and retrieve color information such as RGB values or hexadecimal codes.

2.1.3.2 Edit or Delete Schemes

Allows user to delete any of the save schemes, modify their content, etc.

CHAPTER 3

RGB COLOR MODEL

Color and its manipulation is one of the key task in this thesis project, hence it was necessary to understand how color is represented in our typical display unit (i.e. monitor) and what are the basic steps to create, change colors. Before we understand how the Color Contrast Processor and Color Scheme Generator works we need to understand the basics of how color is translated into digital information and vice versa.

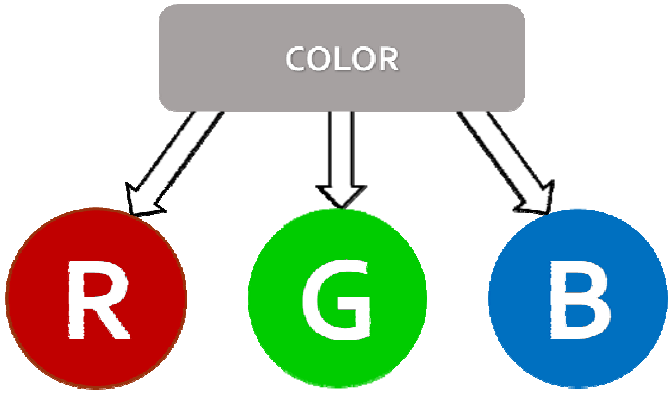

Figure 3.1 - Color is made up of Red, Green and Blue component

Every color we see in our monitor is made up of three important components. They are called the R, G and B components (short for **RGB**). Every monitor contains three light emitters, i.e. Red, Green and Blue. Depending on what color we want, these emitters emit light at varied intensities specified by the instructions we send.

The highest level of intensity that can be set for each of the emitters is 255, while 0 being the lowest. Hence we get a range of 0 to 255, i.e. 256 units effectively.

Let us see how color is created digitally.

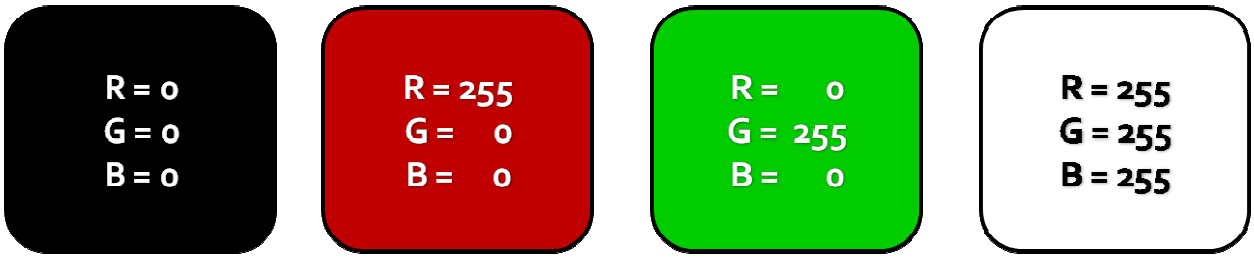

Figure 3.2 - Representation of colors in RGB Color Model

As you can see, when we send instructions to the three emitters with zero intensity we are basically informing them not to emit any light at all from any of the emitters, thus ending up with the color *black*. When we instruct the red emitter to emit light while setting the green and blue emitter to zero, we are only projecting red light, hence getting the color red. When we only tell green emitter to project light and others to remain at zero intensity we end up with the color green. If we tell all three emitters to emit light at its maximum intensity (i.e. 255) we get pure white.

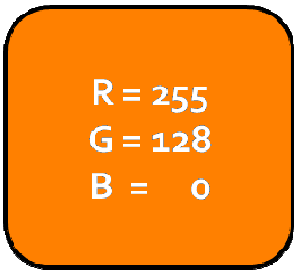

Figure 3.3 - Orange color in RGB Color Model

Here’s another example showing how to get orange. We set the red emitter to its maximum intensity and green to half intensity and blue to zero. The mixture of all three emitted light with specified intensities produces orange.

By varying the intensities of these three emitters we can achieve a huge number of various colors, i.e. over 16 millions of different color combinations to be exact.

This is how we get them, as we know there are three emitters working in tandem to produce various combinations of colors, hence for we get by:

Red X Green X Blue = 256 X 256 X 256

= 16777216

= 16.7 Million colors

CHAPTER 4

COLOR CONTRAST ANALYZER

This part of the application is responsible for analyzing any two colors as input. It will then produce an output that will suggest if the two selected colors are contrasting enough to each other or not. How this is done is discussed later in this chapter. Another feature of this segment of the application is to create a list of alternative color choices for anyone (base color) of the two given colors.

4.1 Algorithms Implemented for Color Contrast Analyzer

In order to find out how well two colors are contrasting to each other, a mathematical approach was required. Fortunately W3C (World Wide Web Consortium) [1] has developed an algorithm to do just that. The following algorithm forms the integral part of finding if two colors are contrasting enough or not.

1. Color Contrast Analyzer
2. Color Difference Algorithm
3. Color Brightness Algorithm

The basic algorithm for ***Color Contrast Analyzer*** is a follows

If **Color Difference** > **500** And **Brightness Difference** > **125** Then (4.1)

"**We Have A Good Contrast**"

End if

The algorithm ***Color Contrast Analyzer*** compares two values, one evaluated from a function called Color Difference which returns a numerical value. The other is another function called Brightness Difference. If Color Difference returns a value greater than 500 and Brightness Difference returns a value greater than 125, then we can say those two colors have a very good contrast between them.

Let's verify how this works.

4.1.1 Color Difference Algorithm

This algorithm takes the RGB values of the two input colors and finds their difference in terms of a numerical value representation.

$$\text{Red'} = (\mathbf{max}(\text{Red}_1 , \text{Red}_2) - \mathbf{min}(\text{Red}_1 , \text{Red}_2)) \quad (4.2)$$

$$\text{Green'} = (\mathbf{max}(\text{Green'}_1 , \text{Green'}_2) - \mathbf{min}(\text{Green'}_1 , \text{Green'}_2))$$

$$\text{Blue'} = (\mathbf{max}(\text{Blue}_1 , \text{Blue}_2) - \mathbf{min}(\text{Blue}_1 , \text{Blue}_2))$$

$$\mathbf{Color\ Difference} = \text{Red'} + \text{Green'} + \text{Blue'} \quad (4.3)$$

Let us take two colors, for example let Color_1 (R,G,B) be equal to Color_1(255,255,255) meaning we get the color white. Let Color_2(R,G,B) be equal to Color_2(0,0,160) where we end up with a nice royal blue as shown below.

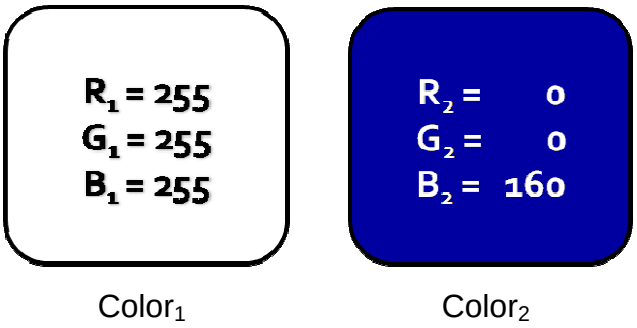

Figure 4.1 – RGB representation of two colors

We take the maximum and the minimum for each component of the two given colors. Let us see how this works.

Consider the first line of the algorithm, Red' = (**max**(Red_1 , Red_2) – **min** (Red_1 , Red_2)). This means out of the two red component of each colors i.e. R_1 = 255 and R_2 = 0, the maximum of these two is obviously R_1. For the minimum of these two red component R_2 = 0 which is the minimum value. Now we subtract the R_2 value from R_1 to get Red' to get 255.

We do this for the remaining components, till we get Red', Green' and Blue'. In this example Green' = 255 as well, and Blue' = (255-160) = 95. The summation of the three component yields the color difference as given below.

Color Difference = Red' + Green' + Blue' (4.3)

= 255 + 255 + 95

= 605

We get 605, which is greater than 500. We are done with the first part of our analysis. Next have to find if these two colors have the brightness difference of greater than 125.

4.1.2 Brightness Difference Algorithm

The Brightness Difference algorithm checks the brightness of the two input colors and returns a numerical value to depict the difference. If the value is greater than 125, then we have a good difference.

Let us see how this algorithm works:

$$Color_1 = ((Red_1 X\ 299) + (Green_1\ X\ 587) + (Blue_1\ X\ 114)) / 1000 \qquad (4.4)$$

$$Color_2 = ((Red_2\ X\ 299) + (Green_2\ X\ 587) + (Blue_2\ X\ 114)) / 1000$$

Brightness Difference = **Abs** ($Color_1 - Color_2$)

We simple take the two colors and apply their respective components to the mathematical equation called the Brightness Difference [2] formula suggested by the W3C WCAG 1.0 and WCAG 2.0 group (Web Content Accessibility Guideline) [3].

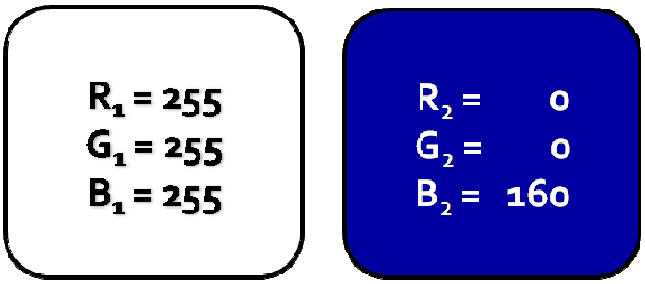

Figure 4.2 – RGB representation of two colors

For the same two colors shown above, we would get a Brightness Difference of 236. This is greater than the suggested minimal value 125. We are done with the second part of the ***Color Contrast Analyzer*** algorithm. Now let us put all this together.

Recall the algorithm (4.2), (4.3) with which we got our Color Difference of 605. Using algorithm (4.4) we got our Brightness Difference of 236. Now using these values in Color Contrast Analyzer algorithm (4.1) we get:

If **Color Difference** > **500** And **Brightness Difference** > **125** Then (4.1)

"**We Have A Good Contrast**"

Else

"Bad Contrast"

End if

We suggest the user that the contrast is good and conforms to the W3C's guidelines.

4.2 Enhancements to the Color Contrast Analyzer Algorithm

Research indicated that a Color Difference of greater than 500 and Brightness Difference of greater than 125 is good for finding strong or very high contrasting colors. This excludes all the colors that have soft or medium contrasts. Medium contrast colors are still very much readable against each other. Hence through trial and error a new minimal color difference value was discovered that can provide us with lighter colors but yet contrasting enough. Instead of 500 we can use 350 or greater if and only if we are looking to find softer contrast.

If **Color Difference** > **500** And **Brightness Difference** > **125** Then

"**We Have A Good Contrast**" (4.5)

Else If **Color Difference** > **350** And **Brightness Difference** > **125** Then

"We Have A Medium Contrast"

Else

"Bad Contrast"

End if

4.3 Sample Output of Color Contrast Analyzer

Here are few samples to show exactly what user gave as input and what the Color Contrast Analyzer produces as its output.

Good Contrast !		Sample Text !	Color Difference = 605 Brightness Difference = 236
Bad Contrast !		Sample Text !	Color Difference = 382 Brightness Difference = 67
Bad Contrast !		Sample Text !	Color Difference = 127 Brightness Difference = 15
Bad Contrast !		Sample Text !	Color Difference = 127 Brightness Difference = 38

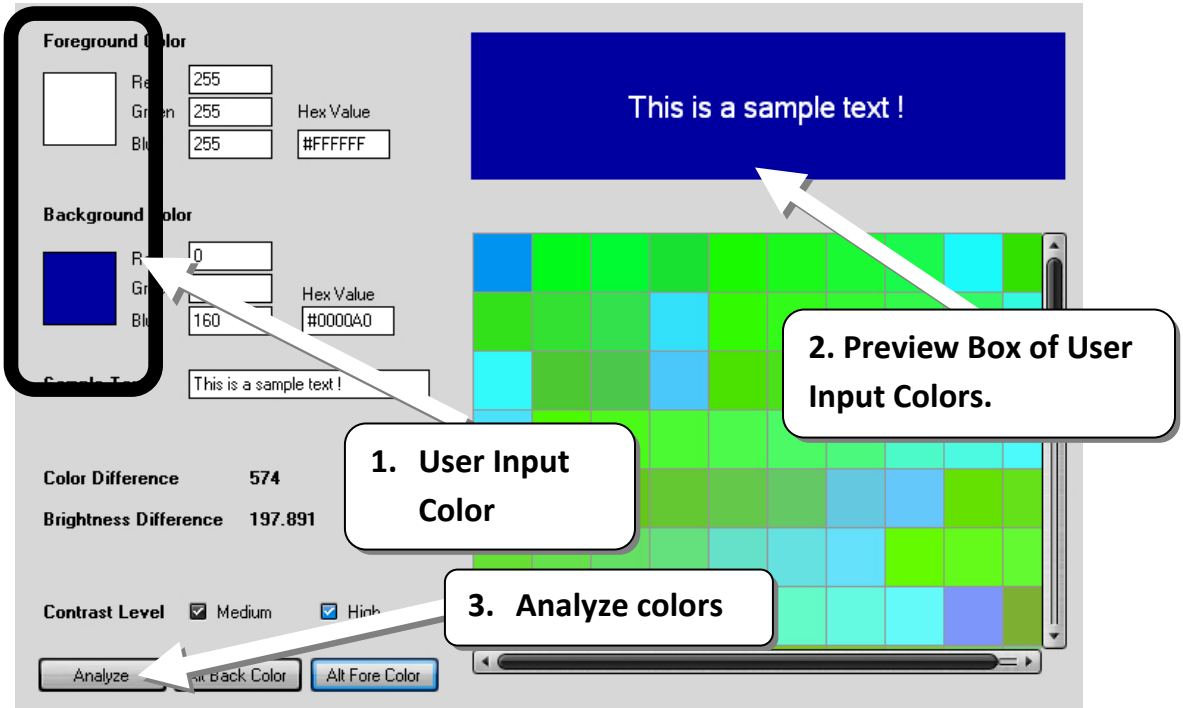

Figure 4.3 – How it all works in Color Contrast Analyzer

4.4 Alternative Color Palette Generator

An neat feature added to this section is the color palette generator for alternative background colors that may go well with the given foreground color, or for alternative foreground colors for a given background color.

```
For  R = 0 To 255                                                    (4.6)
        For G = 0 To 255
                For B = 0 To 255
                        AltColor = Create New Color(R,G,B)
                        Color Difference = (AltColor, UserColor)
                        If Brightness Difference > 125 And Color Difference > 350 Then
                                Add New Color To The Palette = AltColor (R, G, B)
                        End If
                Next B
        Next G
Next R
```

This algorithm iterates for every single possible colors that is possible in a RGB Color Model. During every iteration it checks to see if new color created (i.e. AltColor) using the R, G and B values passed as parameters has a color difference greater than 350. If it does then that color is added to the palette list.

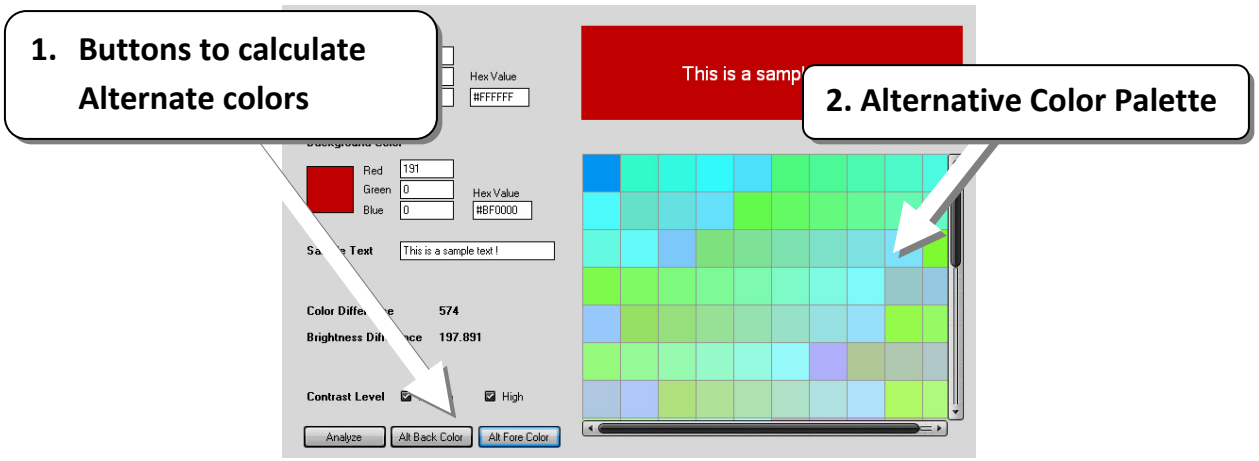

Figure 4.4 - Alternative color generation in Color Contrast Analyzer module

CHAPTER 5

COLOR SCHEME GENERATOR

Color schemes are created in this part of the application where user selects an input from the color wheel [4] and based on what type of scheme the user has selected a new color palette is created, listing the colors that appear harmonious to each other. Color harmony forms the basis of creating good color combinations.

5.1 Color Wheel

Color wheel is the heart of color theory [5] and the various schemes that can be generated from it. Johannes Ittens [6], a color theorist reinvented the color wheel and mapped out various schemes that can be made from it and other variations that can be derived from the basic scheme sets.

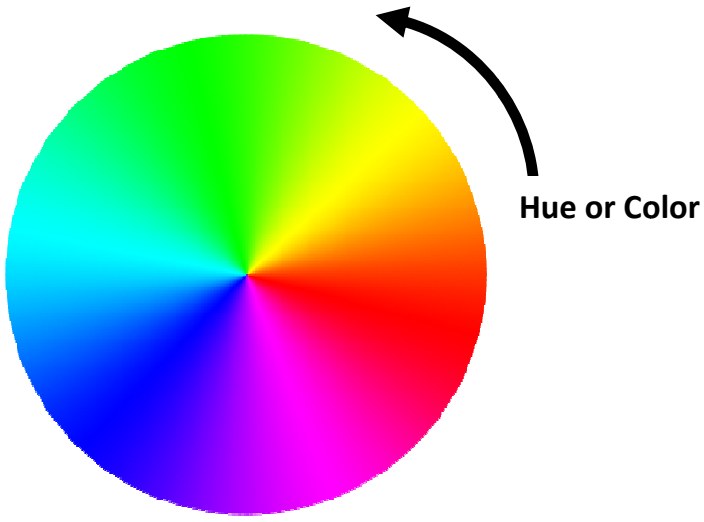

Figure 5.1 - The Color Wheel

Hue is nothing but the color we get when we move around the color wheel.

5.2 Color Theory

As mentioned earlier, color theory is used to find different types of color scheme sets. Some of the basic ones are Analogous, Monochromatic, Triad, Complimenary, Split Complimentary, etc.

5.2.1 Analogous Color Scheme

This scheme generates colors adjacent or colors that lie side by side to the base color (i.e. the color that the user picks).

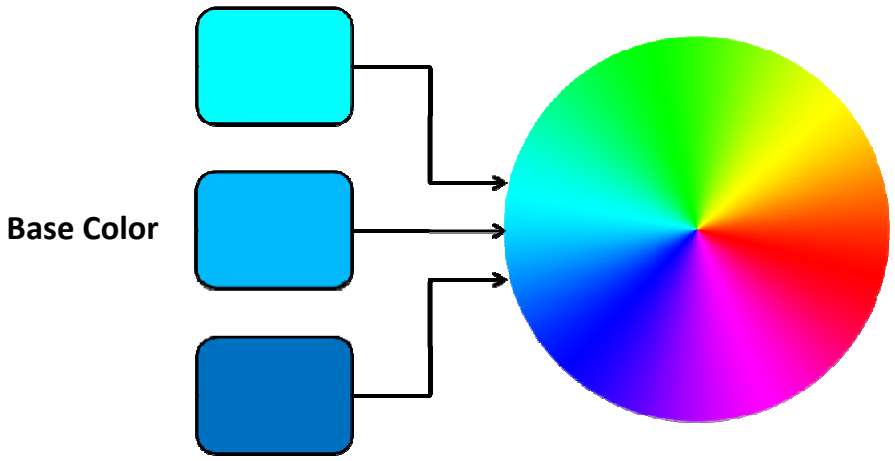

Figure 5.2 - Analogous Color Scheme

5.2.2 Monochromatic Color Scheme

This color scheme is generated based on the theory that given a base color, produces its lighter and darker variations.

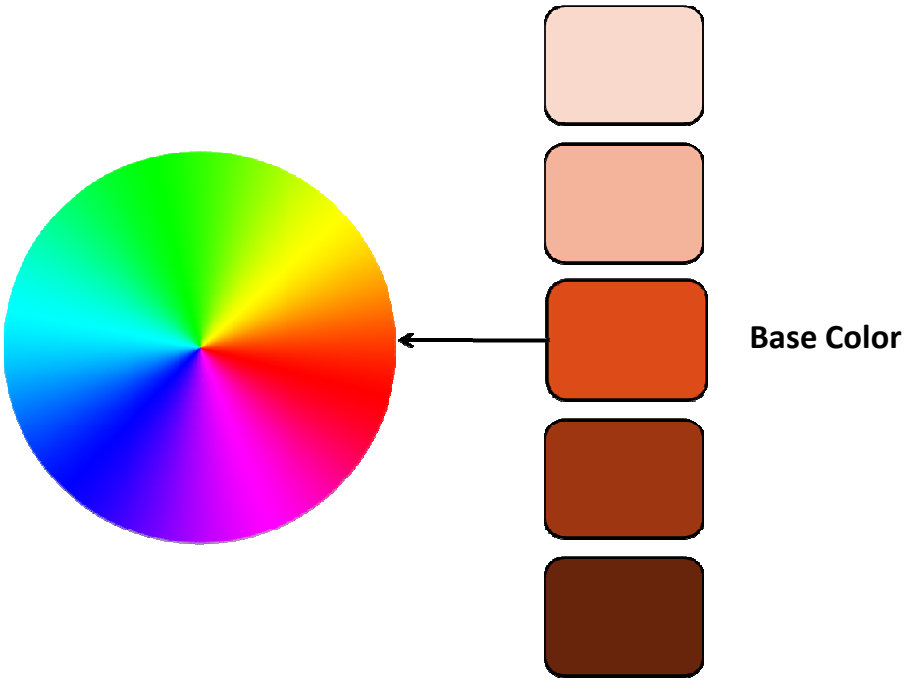

Figure 5.3 - Monochromatic Color Scheme

5.2.3 Complimentary

Complimentary scheme generates a color opposite to the base color on the color wheel.

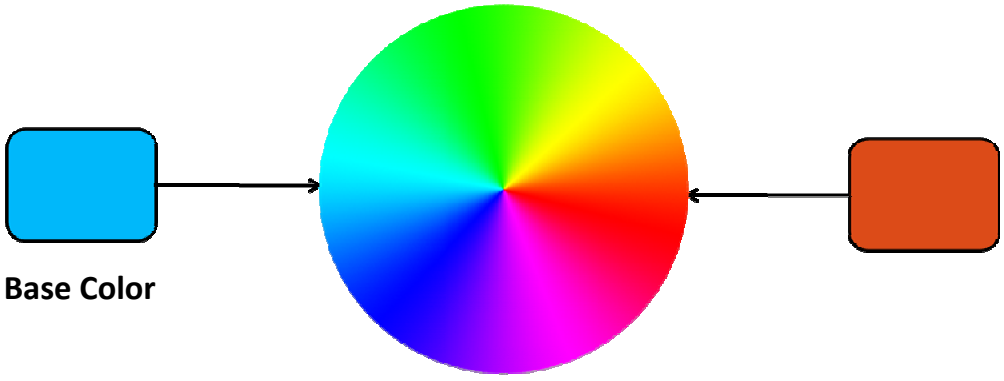

Figure 5.4 - Complimentary Color Scheme

5.2.4 Split Complimentary

This is a variation of the original complimentary, except that the colors we get opposite to the base color on the color wheel are the analogous of its actual complimentary color.

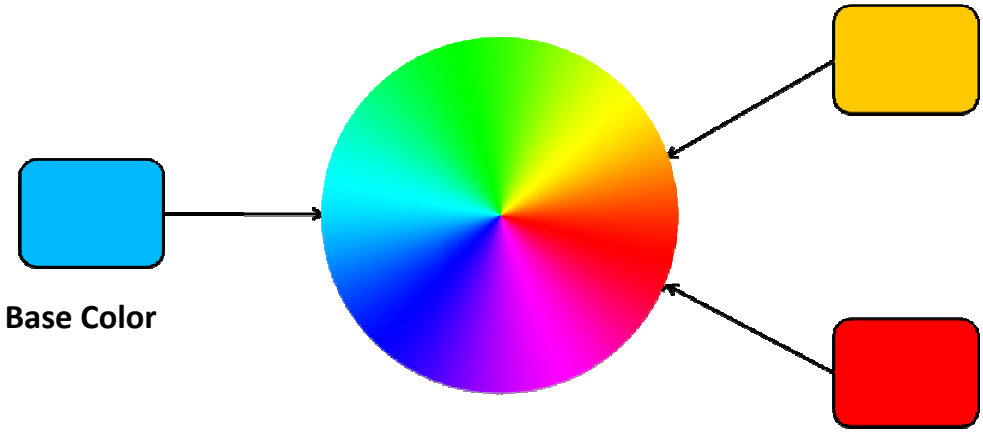

Figure 5.5 - Split Complimentary Color Scheme

5.2.5 Triad Color Scheme

This scheme produces a set of colors that appear 120^{o} away from the base color on both the sides.

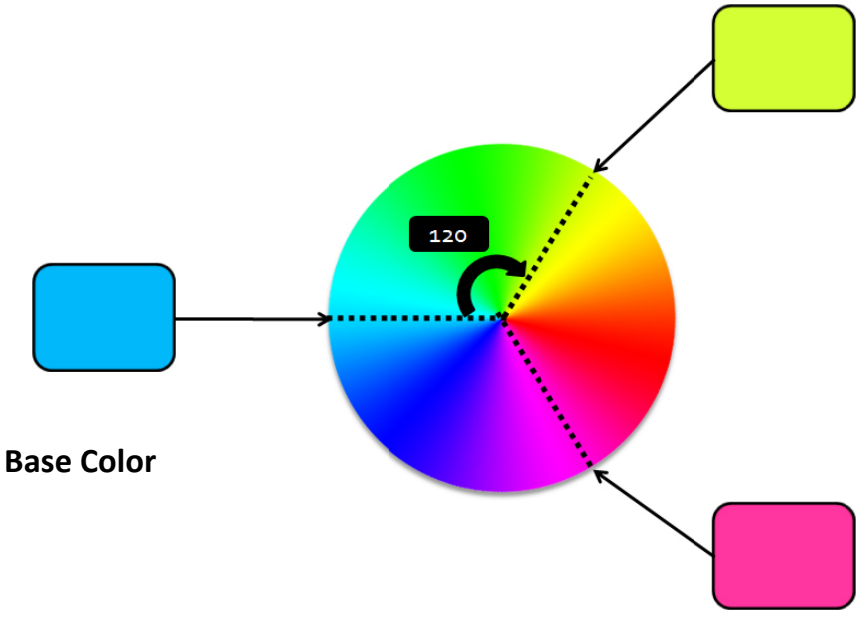

Figure 5.6 - Triad Color Scheme

Among the 5 basic types only 3 were implemented and they are:

1. Monochromatic Color Scheme
2. Complimentary Color Scheme
3. Triad Color Scheme

5.3 How It All Works in Color Scheme Generator

The following images are screen shots of the application ***Color Contrast Processor & Color Scheme Generator*** (also known as ***Color Doc***).

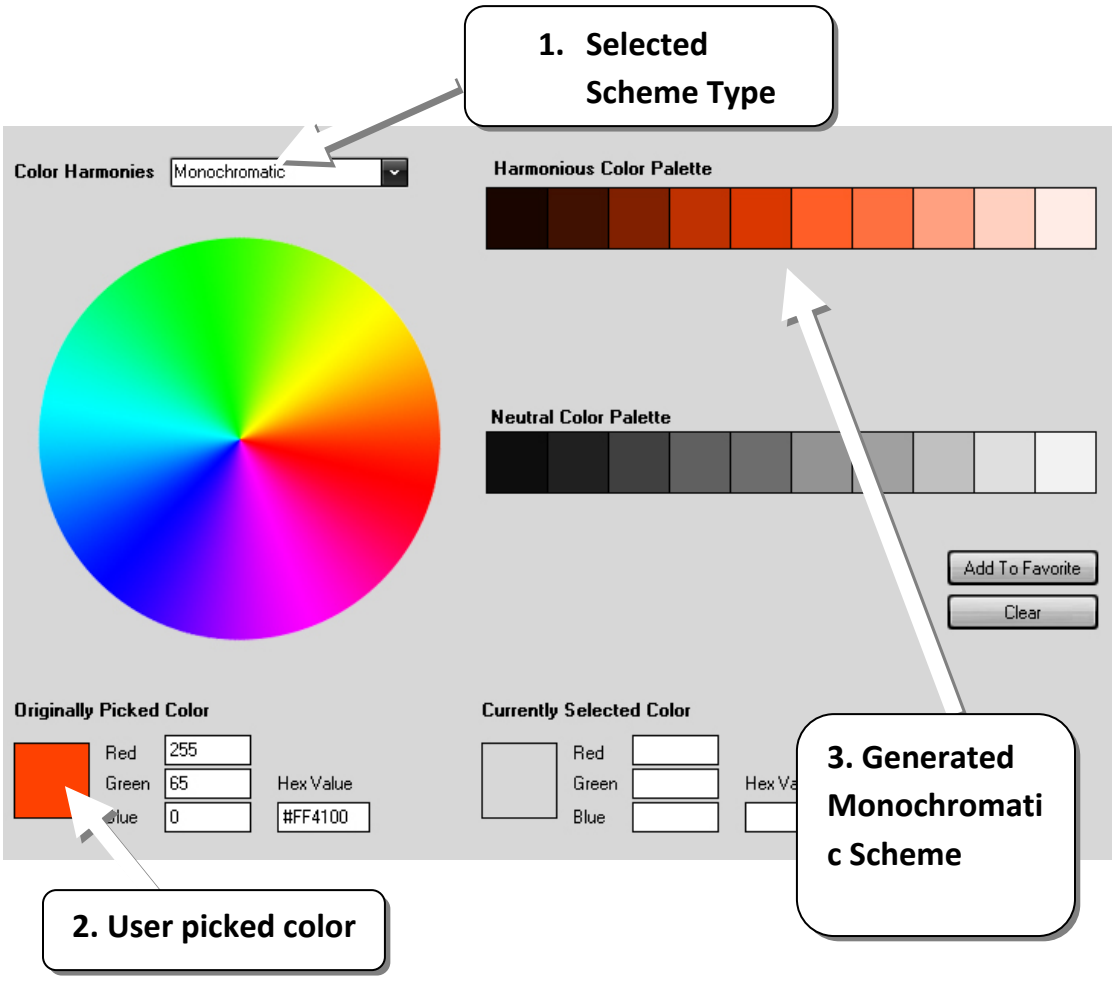

Figure 5.7 - Generating color scheme in Color Scheme Builder

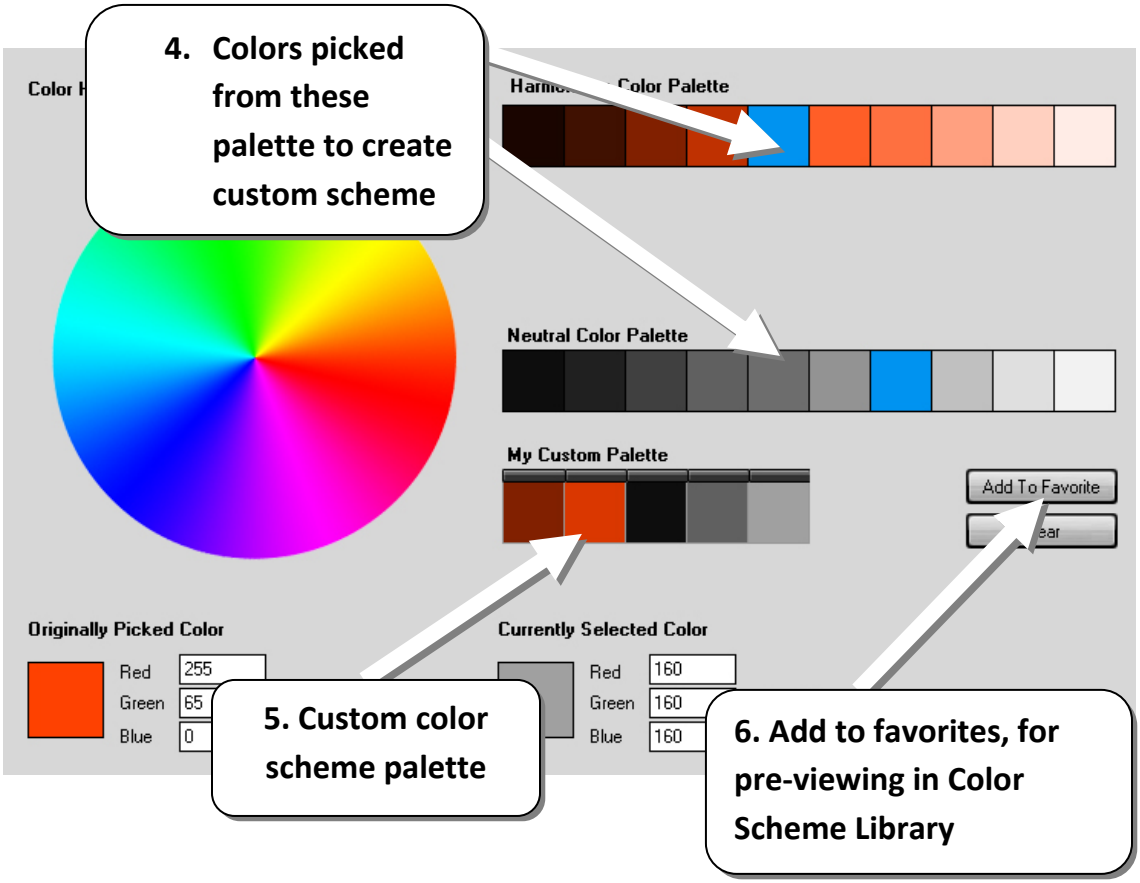

Figure 5.8 - Creating custom colors in Color Scheme Builder

CHAPTER 6

COLOR SCHEME LIBRARY

The last section is a basic inventory system for displaying all the pre-defined and saved color schemes from the database. Its main purpose is to assist users in locating a certain scheme from the list and retrieve the color information of each color in a selected scheme if required. It also has the functionality to delete any scheme from the list.

6.1 How it all works in Color Scheme Library

The following the screenshot represents a typical scenario of how things work in this module.

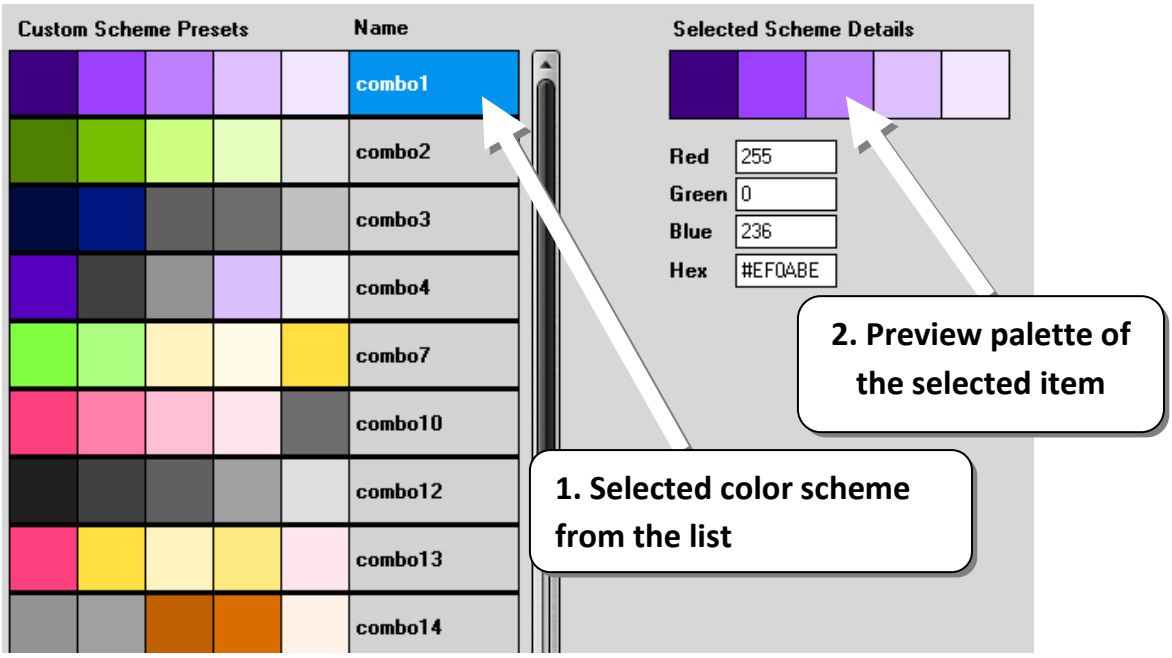

Figure 6.1 - Previewing color schemes in Color Scheme Library

CHAPTER 7

POSSIBLE EXTENSIONS

Throughout the development process new innovative ideas were being born, few parts of the application were left out due to its complexities and a need for sophisticated building tools. However, there is a lot of room for improvements and needs to be mentioned.

A summarized version of the possible extensions that can be integrated to the application are as follows.

1. Detection of websafe colors.
2. Detection of colors that will potentially affect people with color blindness.
3. Extraction of color information from various sources such as images, Flash Symbol Objects, Adobe Illustrator Vector Arts.
4. Add Analogous and Split Complimentary schemes to the ***Color Scheme Generator***.
5. Creation of hybrid color schemes from the basic scheme sets.
6. Add functionality in the ***Color Contrast Analyzer*** to let user compute alternative colors based on a given range, eg. Find all alternative colors which are near blue color.
7. Add functionality to the ***Color Contrast Analyzer*** that would allow user to also save favorite contrasting color sets for later reference.
8. Add a feature to search for color schemes in the ***Color Scheme Library*** based on a search criteria such as; Find schemes that have colors that are red or near red.

CHAPTER 8

CONCLUSION

The application Color Doc (based on the research and principles of Color Contrast Processor And Color Scheme Generator) that has been developed is a working model of the proposed design. Although the specifications declared in the objectives have been met and have been tested thouroughly, it still needs improvements in certain areas before it can be put to practical use.

The concept of Color Contrast Processor And Color Scheme Generator is intended for developing an application that is flexible, easy to use and provide satisfaction to its end-use (i.e. Software GUI and Web Developers/Designers) by automating a lot of manual process(such as finding color contrasts, color combinations, etc).

There are however a lot of places where it needs improvement and tweaking. For example, the algorithm used to find alternative contrasting colors adds colors that appear same, by eliminating such near similar colors could provide better decision making to the end-user. The algorithm used in making color schemes works great but a more sophisticated method of calculating color combinations could be used to produce better looking color combination palettes. These few issues are but stepping stones for making Color Doc application a better working product.

LIST OF REFERENCES

REFERENCES

[1] http://www.color-wheel-pro.com/ (Color Wheel Pro)

[2] http://www.colorschemer.com/ (Color Schemer Studio)

[3] http://www.colorcombos.com/ (Color Combos)

[4] http://www.worqx.com/ (WORQX)

[5] http://gmazzocato.altervista.org/colorwheel/wheel.php

[6] http://www.w3.org/ (World Wide Web Consortium)

[7] http://www.w3.org/TR/AERT/ (Techniques For Accessibility Evaluation And Repair Tools)

[8] http://www.w3.org/TR/WAI-WEBCONTENT/ (Web Content Accessibility Guidelines 1.0)

[9] http://en.wikipedia.org/wiki/Color_wheel/ (Color Wheel, Wikipedia)

[9] Jim Krause, Color Index, HOW Design Books, 2000

[10] http://en.wikipedia.org/wiki/Color_theory/ (Color Theory, Wikipedia)

[10] Color Harmony Workbook, Edition Olms Zurich, 3rd Edition, 2004

[11] http://en.wikipedia.org/wiki/Color_theory/ (Johannes Ittens, Wikipedia)