***Introduction:***

Applications such as Geographic Information Systems provide users with the ability to choose predetermined and/or custom color configurations to represent the data. The majority of users will be inclined to pick a color scheme similar to the one in Figure 1. The color scheme contains several different colors, but the color scheme does not enable users to successfully interpret data.

According to Dr. Ananya Mandal, anywhere from five to eight percent of men may be colorblind and anywhere from 0.5 to one percent of women may be colorblind (Dr. Mandal). There are three types of partial colorblindness, Tritanopia, Protanopia, and Deuteranopia. Tritanopia is the colorblindness that results from sensitivities in the short-wavelength cones; it is difficult to determine the difference between colors such as blue and yellow. Deuteranopia is the colorblindness that results from sensitivities in the medium-wavelength cones. Protanopia is the colorblindness that results from sensitivities in the long-wavelength cones. Interestingly, individuals that are sensitive to long or medium wavelength colors are unable to distinguish between green and red. Visualization programs often fail to offer explicit options for those that contain the colorblind genes. The goal of this project was to create software that would allow user(s) to dynamically alter the color patterns while adhering to, possible, colorblind restrictions entered by the user.

***Visualization Guidelines:***

*Guideline 4.14 – To create a set of symbol colors that can be distinguished by the most colorblind individuals, ensure variation in the yellow-blue direction.*

We provide both command line options and keyboard interrupt options for the user(s), to ensure that colors can be distinguished by the majority of colorblind individuals. The user(s) can execute the program passing the argument “-cb colorblind\_option”. The colorblind\_option refers to either “A”, “D”, “P”, or “T” which represent the completely colorblind, medium-wavelength, long-wavelength, or short-wavelength sensitivities. The user(s) can also enter these letters during the execution in order to change the current colorblind arguments.

*Guideline 4.15 – Do not use more than ten colors for coding symbols if reliable identification is required, especially if the symbols are to be used against a variety of backgrounds.*

In order to successfully code this visualization guideline, we ensured that the default number of symbols would be eight. We chose eight, because it is less than the suggested maximum and we noticed that eight bands would provide each of the colors with enough contrast to easily distinguish the color changes. We did allow the user(s) to alter the number of color bands using a command line argument “-b number\_of\_bands”.

*Guideline 4.19 – Use a spectrum approximation psuedocolor sequence for applications where its use is deeply embedded in the culture of users. This kind of color sequence can be used where the most important requirement is reading map values using a key. If this sequence is used, the spacing of the colors should be carefully chosen to provide discriminable steps.*

We have used and elevation data map to illustrate the use of a psuedocolor sequence in which the colors are important for determining elevation bands. We have created an algorithm that will pick two saturated colors that are at least a certain distance apart in order to create bands that are easily distinguishable.

*Guideline 4.20 – If it is important to see highs, lows, and other patterns at a glance, use a psuedocolor sequence that monotonically increases or decreases in luminance. If reading values from a key is also important, cycle through a variety of hues while trending upwards or downward in luminance.*

We have used and elevation data map to illustrate the use of a psuedocolor sequence in which the colors are important for determining elevation bands. We have created an algorithm that will pick two saturated colors that are at least a certain distance apart in order to create bands that are easily distinguishable. We provide VTK with two distinct colors and a midpoint. VTK chooses the remaining colors for the lookup table based on the luminance values for the colors between each endpoint and the midpoint.

NOTE: We can mention why we did not choose to use the guideline that discusses changing background colors. We can also mention why we did not choose to use the guideline that discusses the scaling rate (G10.14).

***Methodology:***

In this section we can talk about the algorithms for

1. Picking the colors – ensuring that they are saturated
2. Algorithm(s) for picking the mid color
3. Algorithm for picking colors a certain distance apart
4. Algorithm for making colorblind options
5. Dynamic aspects of the project

***Practical Applications:***

In this section we can talk about the practicality of the project.

***Results:***

In this section we will discuss the results of the project and what we accomplished.

***Conclusion:***

In this section we will go back through and note all of the stuff that was discussed.

***Acknowledgements/Resources:***

<http://www.news-medical.net/health/Color-Blindness-Prevalence.aspx>

Course Textbook.