Visualizing Complex Functions (VCF) Test Plan Document

Document Overview

This document describes a test plan for the development of the VCF application. The primary goals of our testing includes validating accuracy in modeling complex functions, simplicity and usability of the program with relation to the end user, and functionality in different environments (browser, OS).
The primary goals of our testing are:

- Determining function drawing errors.
- Evaluating problems in the user interface. These problems may be:
    - Simplicity errors: The interface makes the system appear too complex for the average user to understand.
    - Functionality errors: The interface does not correctly map to the system function it represents.
- Finding errors once porting the application to different environments.
- Determining the user's problems by testing a small group of potential users.

The average user of this program will be students studying complex functions.
Their purpose in using it will be to gain knowledge about complex functions through visualization. We will try to test a group of these students if possible to evaluate the interface and functionality of our program.

Test Plans & Impact Analysis

Frequency Scale

High - 30% or more of the participants experience the problem
Moderate - 11% - 29% of participants experience the problem
Low - 10% or fewer of the participants experience the problem

Impact Scale

High - prevents the user from completing the task (critical error)
Moderate - causes user difficulty but the task can still be completed (non-critical error)
Low - minor problems that do not significantly affect the task completion (non-critical error)

Name	Description	Freq. & Impact	Test Cases
Input Field	Area for users to input functions to graph	High Freq. High Impact	Buffer Overflow Divide by 0 Unrecognized Variable Uneven number of parentheses Parentheses with empty field No entry

			Negative Square Root Int/Double Overflow
String Tokens	Breaking down a string into readable components	Med Freq. High Impact	Evaluating tokens that are invalid on their own (e.g. “-”) Parsing capital and lowercase letters Tokens that can’t be evaluated together
Buttons	Buttons will add to the string	Med Freq. Low Impact	Placement within function (e.g. “ab|c” add () -> “ab()c”)
Sample Function List	Dropdown list of functions to choose from	Low Freq. Med Impact	Choose each function and check if it works Switch rapidly between functions Check how changing the function affects the graph, whether it stays consistent
Color Schemes and Mappings	Dropdown list of color schemes and mappings to choose from	Med Freq. Med Impact	Choose each color scheme and check if it works Switch rapidly between color schemes Change combinations of color schemes and functions
Tutorial	Helpful guide to using the program	Med Freq Med Impact	See if the user can operate the program with reasonable skill after seeing the tutorial Evaluate interface of tutorial
Interface Design	Conveys program features effectively and ease of use by our target user (Mathematics student)	High Freq Med Impact	See if the user can infer what each button does at first glance See if the user can operate the program with relative ease Get user feedback on the interface
Complex Function Methods	The process of evaluating the function input into something mathematical	Med Freq High Impact	Try different functions and check the program’s values at runtime Feed those functions into the graph Invalid function strings
Graphing Methods	The process of turning the function into a visible graph	Med Freq High Impact	Feed different functions to the graphing method Feed very large functions (large w and z values)

			Feed very small functions
Website Interaction with Java Applet	The effects of having the application embedded on a website	Low Freq Low Impact	Check the website's formatting and compare it to what it should be Embed the application on other websites
Web Browser Compatibility	The consistency of the application between the major browsers	Low Freq Low Impact	Try the application on Chrome, Firefox, IE, Safari Reload the page multiple times with many tabs
Window Placement	The windows display the different components	Low Freq Low Impact	Make sure the window doesn't go off the screen Make sure windows don't overlap
Window Size	The windows are the correct size for the data they represent	Low Freq Med Impact	Make sure no letters are cut off on drop down lists Try large graphs that might go off its window Try small graphs to see if the window scales to show more detail
Window Format	The windows display their data in an intuitive format	Med Freq Low Impact	Check users and see if their graph is readable with the data on the window Check users for all other windows that they understand the data presented
Window Color Scheme and Mapping	Make sure the selected color scheme/ shading always matches the visible graph	Med Freq. Low Impact	Choose each color scheme Switch color schemes for the same function Do the same for shading