Chroma Canvas Test Plan

Version 2.0

December 17,2023

Matt Engrav, Sean Clewis, Marley Brandon, Amad Saeed, James Stevens, Josh Williams

Lead: Matt Engrav mjemt9@umsystem.edu

Revision History:

| Version | Date | Author | Description of Changes |
| --- | --- | --- | --- |
| 1.0 | 11/17/2023 | Sean Clewis | Initial version |
| 1.1 | 11/20/2023 | Sean Clewis | Updated Section 1 Introduction |
|  |  |  | * Clarified the description of the Paint Canvas program and its functionalities |
|  |  |  | * Added details on the "Get Started" button and the subsequent user interactions. |
|  |  |  | * Revised the flow of the user experience. |
|  |  |  |  |
| 1.2 | 11/26/2023 | Sean Clewis | Updated Section 2 Items to be Tested |
|  |  |  | * Modified test cases in 2.1 Input for initial painting to enhance coverage and clarity. |
|  |  |  | * Added test cases for the animation (2.2) and input for experiment (2.3). |
|  |  |  | * Included additional checks for the loading screen and site continuity (2.3.29, 2.6). |
|  |  |  |  |
| 1.3 | 12/1/2023 | Sean Clewis | Updated Section 2 Items to be Tested |
|  |  |  | * Revised test cases in 2.3 Input for experiment for clarity and completeness. |
|  |  |  | * Modified test cases in 2.4 Table of Results to ensure accurate display of results |
|  |  |  | * Expanded test cases in 2.5 Results Chart for comprehensive coverage. |
|  |  |  |  |
| 1.4 | 12/6/2023 | Sean Clewis | Updated Section 2 Items to be Tested |
|  |  |  | * Enhanced test cases in 2.2 Animation for better coverage of animation features. |
|  |  |  | * Revised and added checks for input validation in 2.3 Input for experiment. |
|  |  |  |  |
| 1.5 | 12/10/2023 | Sean Clewis | Updated Section 2 Items to be Tested |
|  |  |  | * Ensured comprehensive testing of 2.6 Site Continuity, including page resets and variable clearing. |
|  |  |  |  |
| 2.0 | 12/17/2023 | Sean Clewis | Updated document to adhere to new name Chroma Canvas |
|  |  |  | * Updated layout and file format |
| 2.1 | 12/17/2023 | Marley Brandon | Updated Section 2 Items to be Tested |
|  |  |  | * Added tests for section 2.4-2.6 and tested. |

# Section 1: Introduction

The Chroma Canvas program is hosted on a website and will begin with an introduction animation that explains the website then and a “Get Started” button. Once the user clicks the button it will navigate to a page with a form for variables that go with an initial “experiment”. The user will enter the size of the grid, the three colors they wish to use, and a stopping criteria for the experiment. After those inputs are submitted and verified a single animation will play of the experiment with drops of paint falling onto a grid until the stopping criteria is met.

Once this has been completed the user will have the option to continue and

after they click that button they will start an experiment with a series of the previous painting. They will need to specify the independent and dependent variables, the grid size, the colors used, the stopping criteria, and the number of repetitions. After those have been chosen the website will run through the inputs and once the experiment is complete it will display the results. Firstly it will display a table with the results, then it will ask users to select data points to create a graph of the data. At this point the user will be prompted to either quit or start over.

# Section 2: Items to be Tested

The testing environment should be Google Chrome on Version 119.0.6045.159 (Official Build) (64-bit).

**2.1 Input for initial painting:** The form for inputs should be tested by first entering invalid data, numbers too low, too high, and characters other than digits. Check that appropriate error messages are given. Then enter data that should be valid and submit, check console log to see that variables were assigned properly.

**2.**1.1 Enter -5 for X. Should give error message “Your X dimension value must be between 1 and 50”

**2.**1.2 Enter 33.7 for X. Should give error message “Please enter a valid value”

**2.**1.3 Enter 0 for X. Should give error message “Your X dimension value must be between 1 and 50”

**2.**1.4 Enter 51 for X. Should give error message “Your X dimension value must be between 1 and 50”

**2.**1.5 Enter 33 for X. Should accept value and move onto Y dimension.

**2.**1.6 Enter -5 for Y. Should give error message “Your Y dimension value must be between 1 and 50”

**2.**1.7 Enter 33.7 for Y. Should give error message “Please enter a valid value.”

**2.**1.8 Enter 0 for Y. Should give error message “Your Y dimension value must be between 1 and 50”

**2.**1.9 Enter 51 for Y. Should give error message “Your Y dimension value must be between 1 and 50”

**2.**1.10 Enter 33 for Y. Should accept value and move onto color choice.

**2.**1.11 Enter Red, Grey, Blue from dropdown menus. Should accept value and move on.

**2.**1.12 Enter Pink, Red, Purple from dropdown menus. Should accept value and move on.

**2.**1.13 Enter Orange, Blue, Pink from dropdown menus. Should accept value and move on.

**2.**1.14 Enter Every Square is full for stopping criteria. Should accept value and move on.

**2.**1.15 Enter A single square was double dropped on for stopping criteria. Should accept value and move on.

**2.**1.16 Enter (2 \* Grid Size) color drops have fallen for stopping criteria. Should accept value and move on.

**2.**1.17 Attempt to submit form without filling out X dimension. Should give error message “please fill out this field.”

**2.**1.18 Attempt to submit form without filling out Y dimension. Should give error message “please fill out this field.”

**2.2 Animation:** Visually inspect after the form is submitted that the animation runs smoothly, the grid is properly sized, the correct colors are used, and the appropriate stopping criteria is used.

**2.**2.1 Enter X and Y dimensions of 5 and 5 to ensure that grid is 5x5 square grid.

**2.**2.2 Enter X = 10 and Y = 2 to ensure that grid is 10x2 sized grid.

**2.**2.3 Enter X = 1 and Y = 20 to ensure that grid is 1x20 sized grid.

**2.**2.4 Enter colors red, green, and blue and ensure that when painting only those colors are dropped.

**2.**2.5 Enter colors pink, grey, and orange and ensure that when painting only those colors are dropped.

**2.**2.6 Enter stopping criteria Every Square is full and ensure that animation stops once every square is full.

2.2.7 Enter stopping criteria A single square was double dropped and ensure that animation stops once a square is double dropped on.

**2.**2.8 Enter stopping criteria (2 \* Grid Size) color drops have fallen and ensure that animation stops once a square is double dropped on.

**2.**2.9 Ensure that clicking slow down button slows down animation.

**2.**2.10 Ensure that by clicking the slow down button repeatedly the max it will slow down by is 0.2x speed.

**2.**2.11 Ensure that clicking speed up button speeds up animation.

**2.**2.12 Ensure that by clicking speed up button repeatedly the max it will speed up by is 3.0x speed.

**2.**2.13 Ensure that dripping noise plays each time a paint drop animation is played.

**2.3 Input for experiment:** The form for inputs should be tested by first entering invalid data, numbers too low, too high, and characters other than digits. Check that appropriate error messages are given. Then enter data that should be valid and submit, check console log to see that variables were assigned properly.

**2.**3.1 Enter Single dimension for both X and Y axis and ensure that input is accepted.

**2.**3.2 Ensure that by choosing Single dimension for both X and Y will go on to prompt user for repetitions.

**2.**3.3 Ensure that by choosing Single dimension for both X and Y, these values will be accurately represented in the experiments.

**2.**3.4 Enter X dimension and ensure that input is accepted.

**2.**3.5 Ensure that by choosing X dimension will go on to prompt user for both repetitions and Y dimension.

**2.**3.6 Ensure that by choosing X dimension this value will be accurately represented in the experiments.

**2.**3.7 Attempt to submit form without having chosen a value for # of independent values, should throw error and ask user to “please select a valid number between 2 and 10.”

**2.**3.8 Enter -1 for # of independent values. Should throw error and ask user to “please select a valid number between 2 and 10.”

**2.**3.9 Enter 11 for # of independent values. Should throw error and ask user to “Please select a valid number between 2 and 10.”

**2.**3.10 Enter 3.5 for # of independent values. Should throw error and ask user to “Please enter a valid value”

**2.**3.11 Enter 7 for # of independent values. Should accept values.

**2.**3.12 Attempt to submit independent variable values with nothing. Should throw error “You must provide # independent values that you indicated in the previous step.”

**2.**3.13 Enter 1, 2, 3 when 7 values are required. Should throw error “You must provide 7 independent values that you indicated in the previous step.”

**2.**3.14 Enter 1, 3, 3, 4 when 2 values are required. Should throw error ““You must provide 2 independent values that you indicated in the previous step.”

**2.**3.15 Enter 0, 1, 2 when 3 values are required. Should throw error “Your independent values cannot contain a 0.”

**2.**3.16 Enter -2, 1, 3 when 3 values are required. Should throw error “Your independent values must be in increasing order.”

**2.**3.17 Enter A, 1, 2 when 3 values are required. Should throw error “Your independent values contain something that is not a number. Or, you did not use commas to separate your values.”

**2.**3.17 Enter 1 2 3 4 5 when 5 values are required. Should throw error “Your independent values contain something that is not a number. Or, you did not use commas to separate your values.”

**2.**3.18 Enter 1, 2, 3, 4 when 4 values are required. Should accept values and allow submission of form.

**2.**3.19 Enter -1 for # of repetitions, Y dimension, and X dimension. Should throw error “Please select a valid number between 1 and [value cap].”

**2.**3.20 Enter 1000000000 for # of repetitions, Y dimension, and X dimension. Should throw error “Please select a valid number between 1 and [value cap].”\

**2.**3.21 Enter 0 for # of repetitions, Y dimension, and X dimension. Should throw error “Please select a valid number between 1 and [value cap].”

**2.**3.22 Enter 5 for # of repetitions, Y dimension, and X dimension. Should accept value and continue.

**2.**3.23 Enter Red, Grey, Blue from dropdown menus. Should accept value and move on.

**2.**3.24 Enter Pink, Red, Purple from dropdown menus. Should accept value and move on.

**2.**3.25 Enter Orange, Blue, Pink from dropdown menus. Should accept value and move on.

**2.**3.26 Enter Every Square is full for stopping criteria. Should accept value and move on.

**2.**3.27 Enter A single square was double dropped on for stopping criteria. Should accept value and move on.

**2.**3.28 Enter (2 \* Grid Size) color drops have fallen for stopping criteria. Should accept value and move on.

**2.**3.29 Upon submitting experiment forms, should show a loading screen that accurately represents how far in calculating data the program is.

**2.4 Table of Results:** Check that the results are displayed correctly with the appropriate headers and data points.

**2.**4.1 Enter any values and ensure that table displays, independent value, X Dimension, Y dimension, Color1, Color2, Color3, StoppingCriteria, TotalDrops, TotalColor1, TotalColor2, TotalColor3, MaxDrops1Square, and Average Drops.

**2.**4.2 Enter 3 independent variables with 1 repetition each and ensure that only 3 entries in table displays.

**2.**4.2 Enter X and Y coordinates of 1 and ensure that table displays only one drop for each experiment when every square is full is chosen as stopping criteria.

**2.**4.3 Enter repetitions as 5 and ensure that 5 experiments are shown for each independent variable value.

**2.**4.4 Enter X and Y coordinates of 3, 5, 10, 25, 30 with stopping criteria every square is full and ensure that the dependent variables all scale in size with these coordinates with chosen independent variables.

**2.**4.5 Enter stopping criteria of (2 \* Grid Size) and ensure that total drops is equal to 2 \* grid size in table.

**2.**4.6 Enter stopping criteria double drop and ensure that the max drops on each square is 2 on table.

**2.5 Results Chart:** Verify that all applicable options for dependent and independent variables are available. Then verify that after that form is submitted it properly draws the graph with correct labels, data points, scaling, and visual features.

**2.**5.1 Enter 3 independent variables with 1 repetition each and ensure that if min and max are chosen for just one dependent variable the chart and graph display equal values between the min and MaxDrops1Square, AverageDrops

**2.**5.2 Enter stopping criteria of (2 \* Grid Size) and ensure that total drops min and max are equal on graph.

**2.**5.3 Choose Amax and Amin and ensure that table and chart shows those values.

**2.**5.4 Choose Aavg and A1min and ensure that table and chart show those values.

**2.**5.5 Choose A1max and A2avg and ensure that table and chart show those values.

**2.**5.6 Choose A2min and A3min and ensure that table and chart show those values.

**2.**5.7 Choose A3Max and Amin and ensure that table and chart show those values.

**2.**5.8 Choose Bmax and Bmin and ensure that table and chart show those values.

**2.**5.9 Choose Bavg and Cmax and ensure that table and chart show those values.

**2.**5.10 Choose Cmin and Cmax and ensure that table and chart show those values.

**2.**5.11 Choose A1min and Cavg and ensure that table and chart show those values.

**2.6 Site Continuity:** Do a complete run through of the site ensuring that each page has the background loaded correctly, links and buttons for continuing with the experiment work, and the page properly resets if the user decides to do the experiment again, including that variables from the previous experiment have been cleared out.

**2.**6.1 Choose Make new table/graph at end of program and ensure that the same data is utilized.

**2.**6.2 Choose abandon this experiment and ensure that running experiment with new data returns expected results.

**2.**6.3 Choose quit the program and ensure that it does take user back to introduction of website and allows user to use program again without issues.

**2.**6.4 Upon website loading ensure that background loads properly.

**2.**6.5 Ensure that all buttons result in expected changes.

**2.**6.6 Ensure that variables are cleared out between different experiments.

**Section 3: Testing Record**

|  |  |  |  |
| --- | --- | --- | --- |
| Tester | Sections Tested | Pass/fail | Signature |
| Sean Clewis | 2.1-2.4 | FAIL | Sean |
| Sean Clewis | 2.2 - 2.2.13 | PASS | Sean |
| Sean Clewis | 2.3 – 2.3.23 | PASS | Sean |
| Sean Clewis | 2.2.6, 2.2.7 | FAIL | Sean |
| Sean Clewis | 2.1 - 2.1.15 | PASS | Sean |
| Sean Clewis | 2.4, 2.5 | FAIL | Sean |
| Sean Clewis | 2.4 - 2.5 | FAIL | Sean |
| Sean Clewis | 2.3.1 - 2.3.29 | PASS | Sean |
| Sean Clewis | 2.4, 2.5 | PASS | Sean |
| Sean Clewis | 2.6 | PASS | Sean |
| Sean Clewis | 2.2.6, 2.2.7 | PASS | Sean |
| Marley Brandon | 2.4-2.6 | PASS | Marley |