**Test plan for**

**<< Virtual Interactive Board >>**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| 001 | 30.10.23 | Vaibhav Singh | Initial Draft |
|  |  |  |  |
|  |  |  |  |

1 Introduction 2

1.1 Scope 2

1.1.1 In Scope 2

1.1.2 Out of Scope 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 2

2 Test Methodology 3

2.1 Overview 3

2.2 Test Level……………………………………………………………………………………………………………………………………..3

2.3 Test Completeness 4

3 Test Deliverables 4

4 Resource & Environment Needs 4

4.1 Testing Tools 4

4.2 Test Environment 4

5 Terms/Acronyms 5

# Introduction

The Virtual Interactive Board project is a computer vision-based system that allows users to draw and interact with a virtual canvas in real-time. The system uses a webcam and the Mediapipe library to detect hand landmarks and track the movement of the user's hand. Different colored pens can be selected by the user using the colored rectangles displayed on the screen. The system uses deques to keep track of the points of each color drawn on the canvas. The canvas can be cleared by selecting the "CLEAR" option. This project has applications in education, virtual meetings, and artistic expression.

## Scope

### In Scope

**Functional Requirement:-**

Webcam Feed: The program must accept input video stream from the webcam.

Finger Detection: The program shall utilize a hand landmark model to track various fingers co-ordinates.

User Interface: The program should provide a user-friendly interface for users to draw and edit their work.

Drawing Functionality: Users can use their index fingers to draw or select different formatting styles.

Clearing Canvas: Users are capable of clearing the whole canvas by selecting the ‘CLEAR’ button with their index finger.

**Non-Functional requirement:-**

Performance: The program should have minimal lag or delay while updating the canvas.

Speed: The program should perform efficiently, with minimal processing time.

Precision: The program should be able to detect the main working hand precisely.

Reliability: The user should be abe to use the program and perform basic functions reliably.

Compatibility: The program must be compatible with different builds of computer hardware.

### Out of Scope

Out of scope for the Virtual Interactive Board project are Multiple user collaboration, 3D Drawing, Advanced Filters and Effects, and Ambidextrous drawing.

## Quality Objective

* To create a hands-free digital drawing canvas that uses camera, opencv and mediapipe to recognize and map the hand gestures.
* To reduce the use of hardware components like mouse, touch screen etc. and improve the efficiency and creativity of drawing or annotating.
* To implement various functionalities such as choosing different shapes, colors, size, clearing, erasing, and other functionalities using hand gestures.
* To ensure the system is user-friendly and requires minimal training for new users.
* To make the system robust against different lighting conditions.
* To design the system in a way that it can be easily integrated with other applications or platforms

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

* **QA Analyst**- Vaibhav Mittal
* **Test Manager**- Mrs. Shreela Pareek
* **Configuration Manager**- Mrs. Neha Shukla
* **Developers**- Sarthak Srivastava, Vaibhav Singh
* **Installation Team**- Sarthak Srivastava, Vaibhav Singh

# Test Methodology

## Overview

An Agile methodology is the most suitable for our project. It allows for flexibility, ongoing testing, and adaptation, which are essential for projects that involve machine learning, image processing, and AI. Agile enables you to respond to changing requirements and refine the hand detection algorithm as you gain insights from testing and user feedback.

## Test Levels

The testing to be performed is white box testing.

The testing is performed by the developers along with QA and Configuration Manager.

**Unit Testing:**

Scope: Individual components and functions of the hand detection algorithm.

Objective: To verify that each component works as intended, including layers, loss functions, and optimization steps.

Testing Approach: Developers and machine learning engineers conduct unit tests to validate the correctness of the algorithm.

**Integration Testing:**

Scope: The interactions and interfaces between various components, libraries, and frameworks used in the project.

Objective: To ensure that the integration of different components does not introduce errors or inconsistencies in the style transfer process.

Testing Approach: Developers and testers assess the data flow and interactions between components and detect any integration issues.

**Functional Testing:**

Scope: The complete finger tracking system.

Objective: To validate that the system functions according to specified requirements and that it accurately detects finger manuevers.

Testing Approach: Testers execute functional tests by using the webcam to track their index finger and verifying that the canvas drawing and formatting fucntionality meet the desired criteria.

**Performance Testing:**

Scope: Assessing the system's speed and efficiency in handling finger tracking.

Objective: To measure how well the system performs in terms of processing time, memory utilization, and resource consumption.

Testing Approach: Performance tests evaluate the system's response time and resource usage under various loads and conditions.

**Usability Testing:**

Scope: The user interface and user experience.

Objective: To assess how user-friendly and intuitive the interface is for users to draw and edit their work.

Testing Approach: Usability tests involve users interacting with the system to evaluate the ease of use, clarity, and navigation of the interface.

**Security Testing:**

Scope: The system's security mechanisms, especially for handling user webcam data.

Objective: To identify and mitigate potential security vulnerabilities, including data breaches and unauthorized access.

Testing Approach: Security testing includes penetration testing, data encryption checks, and access control assessments.

**Compatibility Testing:**

Scope: The system's compatibility with various platforms and devices.

Objective: To ensure that the system functions correctly on different operating systems, and devices.

Testing Approach: Testers verify that the system is compatible with a range of devices and configurations.

**Regression Testing:**

Scope: The entire system after updates or changes.

Objective: To confirm that new changes or enhancements do not introduce defects or negatively impact existing functionality.

Testing Approach: Automated regression tests are executed to validate that previously tested features still work as expected.

## Test Completeness

Here you define the criterias that will deem your testing complete.

For instance, a few criteria to check Test Completeness would be

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release
* All finger tracking and canvas drawing tests have been executed, ensuring that various formatting methods have been tested successfully and meet the defined criteria.
* Automated regression tests have been executed, and previously tested features still work as expected after updates or changes.

# Test Deliverables

# Test cases:-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Objective** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | System Initialization | Internal Program Code | Webcam turns on and displays the OpenCV frame with buttons for colors and clear. | Webcam turns on and displays the OpenCV frame with buttons for colors and clear. | Pass |
| 2 | Hand landmarks recognition | Webcam feed containing hand of the user | Hand landmarks are detected by the mediapipe and the positions of the fingers are obtained | Hand landmarks are detected by the mediapipe and the positions of the fingers are obtained | Pass |
| 3 | Ability to Draw | Index finger tracking from webcam feed | The Index finger is tracked accurately and the drawing is visible | The Index finger is tracked accurately and the drawing is visible | Pass |
| 4 | Ability to change colors | Index finger position on the color button | Different colors can be selected and the drawing with selected color is visible on the canvas | Different colors can be selected with index finger and the drawing with selected color is visible on the canvas | Pass |
| 5 | Ability to clear the canvas | Index finger position on the clear button | The user is able to clear the canvas by selecting ‘CLEAR’ button | The user is able to clear the canvas by selecting ‘CLEAR’ button with the index finger | Pass |
| 6 | Ability to move hand freely | Presence of individual index finger | The user is able to move hand freely by performing a pinching gesture and not being able to draw on the canvas while in this gesture | The user is able to move hand freely by performing a pinching gesture and not being able to draw on the canvas while in this gesture | Pass |
| 7 | Closing the program | Internal program code | The user being able to close the program by pressing the ‘q’ button on the keyboard ensuring the program is not running in the background | The user being able to close the program by pressing the ‘q’ button on the keyboard ensuring the program is not running in the background | Pass |
| 8 | Detecting multiple hands | Multiple hands on webcam feed | The program detects multiple hands and assigns a 3D landmark model to each of them | The program was not able to detect any hand after the initial hand | Fail |
| 9 | Visibility of hand | Position of the hand on webcam feed | The program detects the hand accurately no matter the position on the screen | The program fails to detect the hand if the angle of the hand is not exclusively facing the webcam in an upright position | Fail |

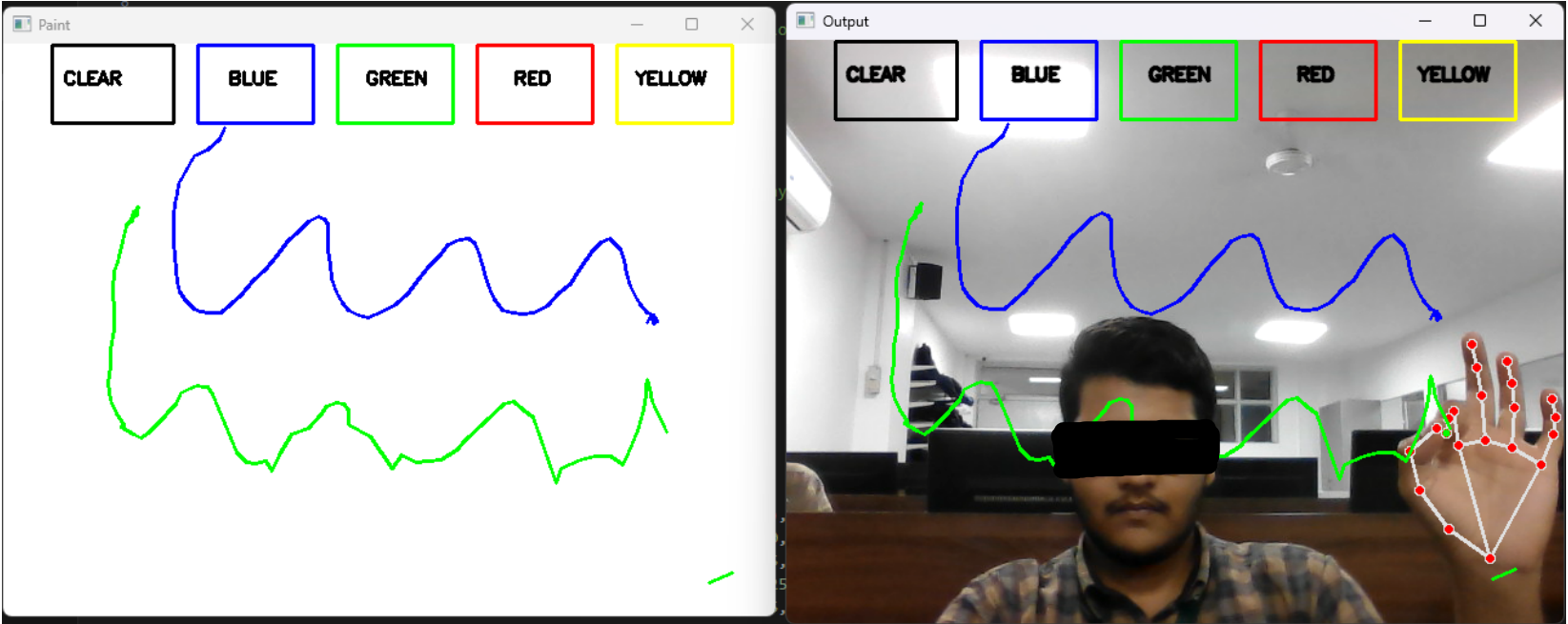
**Decision Table for Hand Gesture**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conditions** | **Rule 1** | **Rule 2** | **Rule 3** | **Rule 4** |
| Index Finger upright | False | True | False | True |
| Pinch Gesture | False | False | True | True |
| Output | Invalid gesture | Draw/Select | Standby | Invalid gesture |

**Decision Table for Assigning Landmark Model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conditions** | **Rule 1** | **Rule 2** | **Rule 3** | **Rule 4** |
| Sufficient Ambient Light | False | True | False | True |
| Presence of Hand | False | False | True | True |
| Output | Not assigned | Not assigned | Not assigned | assigned |

**Output Image**

****

# Resource & Environment Needs

## Testing Tools

No Testing tools are utilized, manual testing is done.

## Test Environment

The minimum **hardware** requirements that will be used to test the Application.

* Any decent entry level PC or Laptop with a discrete GPU

Following **software’s** are required:

1. Windows 8 and above
2. Office 2013 and above

# Terms/Acronyms

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| QA | Quality Assurance |
| PC | Personal Computer |
| GPU | Graphics Processing Unit |

**Testing Faculty**: Mrs. Shreela Pareek

**Project Guide**: Prof. Pradeep Tyagi

**Submitted to**: Mrs. Neha Shukla