
SOFTWARE REQUIREMENTS SPECIFICATION

for

Hand Gesture GUI Control System

Version 1.0 approved

Prepared by
MDL15CS040 19 George Thomas Shanti
MDL15CS082 46 Prince Mathew
MDL15CS089 51 Riza Salahuddin

May 6, 2019

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Document Conventions	4
1.3	Intended Audience and Reading Suggestions	4
1.4	Project Scope	4
1.5	Overview of Developer's Responsibilities	4
2	Overall Description	5
2.1	Product Perspective	5
2.2	Product Functions	5
2.3	User Classes and Characteristics	5
2.4	Operating Environment	6
2.5	Design and Implementation Constraints	6
2.6	User Documentation	6
2.7	General Constraints	6
2.8	Assumptions and Dependencies	6
3	External Interface Requirements	7
3.1	User Interfaces	7
3.2	Hardware Interfaces	8
3.3	Software Interfaces	8
3.4	Communications Interfaces	8
4	Hardware and Software Requirements	9
4.1	Hardware Requirements	9
4.2	Software Requirements	9
5	Functional Requirements	12
5.1	Start/Stop the Hand Gesture Daemon	12
5.2	Add new gestures	13
5.3	Remove gestures	14
5.4	Modify Gestures	15
5.5	Choose webcam	16
6	Other Nonfunctional Requirements	17
6.1	Performance Requirements	17
6.2	Safety Requirements	17
6.3	Security Requirements	17

6.4 Software Quality Attributes	17
7 Other Requirements	18
References	19

1 Introduction

1.1 Purpose

Hand Gesture GUI Control System version 1 revision 1 - A Real time dynamic Hand Gesture Recognition GUI system to improve the interaction between user and computer. The user can use his webcam to record hand gestures which our system can use to perform Linux GUI functions in realtime.

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers of the system. Developers may use document to get a better understanding of the requirements of the system and the prerequisites for building the system.

1.4 Project Scope

The software specified has the purpose of implementing natural and easy to understand hand gestures to control the operating system. The operating system we use here is Linux. We are implementing fluid hand gestures rather than static hand gestures which is tracked by the standard webcam rather than using specialized equipment.

1.5 Overview of Developer's Responsibilities

The developers must design and implement the Hand Gesture Recognition System and the Control System for the GUI. Developers must determine valid gestures and mappings for each gesture to a appropriate function in the GUI of the computer.

2 Overall Description

2.1 Product Perspective

This system arose from the need for a more fluid and convenient way to interact with a computer. It allows you to interact with the computer with no physical contact with any peripheral. Similar systems have been developed but most required additional peripherals or did not offer a wide range of Linux GUI functions that could be accessed via gestures.

Our product serves as an add-on to the current user interface, the mouse and the keyboard

2.2 Product Functions

The major Linux GUI features to be implemented are:

- Switch Window to Next one.
- Drag operations.
- Minimize window
- Close window
- Open Apps Launcher
- Open Terminal
- Scroll down / Up
- Zoom in/ out
- Switch Tabs.

2.3 User Classes and Characteristics

The user base intended for this product are all the Linux users. Linux Developers are encouraged to add-on their own gestures and tinker with the product and contribute.

The normal Linux users can use the product freely without restrictions.

2.4 Operating Environment

The software will run on a Linux machine as a daemon and is not architecture dependant. The only hardware dependant is the camera which must be compatible with the linux operation system with appropriate drivers.

2.5 Design and Implementation Constraints

Implementation constraints include the speed of the computer being used so as to provide a smooth implementation. A good system is required to optimally recognize the gestures from the video input.

A webcam that is compatible with the Linux OS is required to capture the gestures performed by the user and appropriate drivers for the same.

The OpenCV library will be used to recognize the hand gestures from the input.

2.6 User Documentation

An online documentation will be provided consisting of the Hand Gestures that are included.

2.7 General Constraints

The software can be limited by several factors.

- Lack of light
- Complex backgrounds
- Poor camera quality
- Poor System Performance
- System Memory Constraints
- Camera is already is use by another application

2.8 Assumptions and Dependencies

Our assumption is that the product will always be used on a system with a good graphics card as well as a good processor. The operating system will be Linux and will have OpenCV libraries installed in it. A good webcam will be present and will be always searching for hand gestures.

3 External Interface Requirements

3.1 User Interfaces

There will be single user interface where the user can enable and disable the daemon. Add and remove gestures. Change mappings between gestures and commands and map gestures to custom macros and keyboard strokes.

Main UI

Enable/Disable Daemon

Gesture	Command	Enabled
G489	google-chrome	Y
G798	bash	Y
G809	Open App Drawer	N

Add NewDelete Gesture

Add New

Choose cameraChoose action

Camera Feed

CancelAdd Gesture

3.2 Hardware Interfaces

The hardware required for the application is the webcam. The images are then handled by our application and the webcam too is managed by our daemon running in the background.

3.3 Software Interfaces

The daemon must interface with the camera driver and the operating system to be able to control the GUI based on input gestures.

3.4 Communications Interfaces

The application will need internet connection to send bug reports in case of a crash or malfunction. Bug reports will be sent in the form of HTTP requests to a remote webserver.

4 Hardware and Software Requirements

4.1 Hardware Requirements

- A computer with at least 2GHz processor, 1GB RAM and USB ports.
- A USB camera.

4.2 Software Requirements

- Linux operating system.

The linux operating system was chosen due to the fact that it is easy to build software due to wide range of libraries and easy access to them and the compilers. Whereas in a OS like windows a special IDE like Visual Studio and proprietary libraries would be required to build software.

- C++ compliler eg:- gcc

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, Web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms.

Some of the interesting features of C++ are:

- **Object-oriented:** C++ is an object-oriented programming language. This means that the focus is on “objects” and manipulations around these objects. Information about how these manipulations work is abstracted out from the consumer of the object.
- **Rich library support:** Through C++ Standard Template Library (STL) many functions are available that help in quickly writing code. For instance,

there are standard libraries for various containers like sets, maps, hash tables, etc.

- **Speed:** C++ is the preferred choice when latency is a critical metric. The compilation, as well as the execution time of a C++ program, is much faster than most other general purpose programming languages.
- **Compiled:** A C++ code has to be first compiled into low-level code and then executed, unlike interpreted programming languages where no compilation is needed.
- **Pointer Support:** C++ also supports pointers which are widely used in programming and are often not available in several programming languages.

- OpenCV C++ library

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. OpenCV is written in C++ and its primary interface is in C++. All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

- **Core functionality (core)** - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
- **Image Processing (imgproc)** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- **Video Analysis (video)** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
- **Camera Calibration and 3D Reconstruction (calib3d)** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- **2D Features Framework (features2d)** - salient feature detectors, descriptors, and descriptor matchers.
- **Object Detection (objdetect)** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- **High-level GUI (highgui)** - an easy-to-use interface to simple UI capabilities.
- **Video I/O (videoio)** - an easy-to-use interface to video capturing and video codecs.

OpenCV has functions available for background subtraction. Several algorithms were introduced for this purpose. OpenCV has implemented three such algorithms which are very easy to use like BackgroundSubtractorMOG, BackgroundSubtractorMOG2 and BackgroundSubtractorGMG.

OpenCV also has various feature extractors that can be used for detect the fingers of the user like CvFeatureEvaluator, CvFeatureParams, CvHaarEvaluator and CvHaarFeatureParams.

- GTK+ Library for C++

GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off tools to complete application suites.

The GTK+ Library is needed to build an interactive GUI where users can configure the Gesture Control System. Functions to create windows and other GUI elements such as buttons, lists, pop-ups, drop-downs, frames to display video, etc are available in the library. The library also allows to bind functions to events in the GUI such as clicks, key-presses and so on.

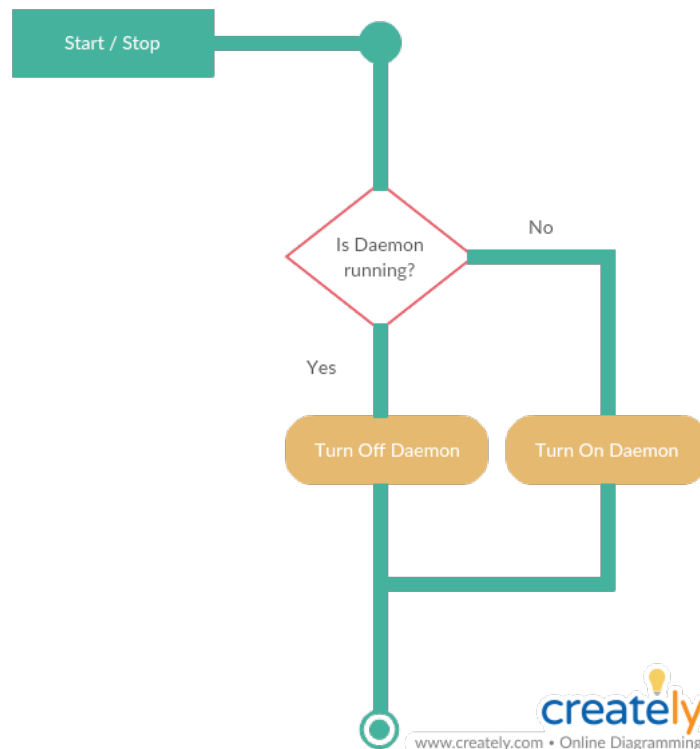
Features of GTK+:

- **Stability:** GTK+ has been developed for over a decade to be able to deliver the enticing features and superb performance that it brings to your application development. GTK+ is supported by a large community of developers and has core maintainers from companies such as Red Hat, Novell, Lanedo, Codethink, Endless Mobile and Intel.
- **Interfaces:** GTK+ has a comprehensive collection of core widgets and interfaces for use in your application.
- **Cross Platform:** Originally GTK+ was developed for the X Window System but it has grown over the years to include backend support for other well known windowing systems. Today you can use GTK+ on Windows, Mac OS X and Linux.
- **Accommodating:** GTK+ caters for a number features that today's developers are looking for in a toolkit including Native look and feel, Theme support, Thread safety and Object oriented approach.
- **Foundation:** GTK+ is built on top of GLib. GLib provides the fundamental algorithmic language constructs commonly duplicated in applications.

5 Functional Requirements

5.1 Start/Stop the Hand Gesture Daemon

A toggle button that starts and stops the daemon which tracks hand gestures.



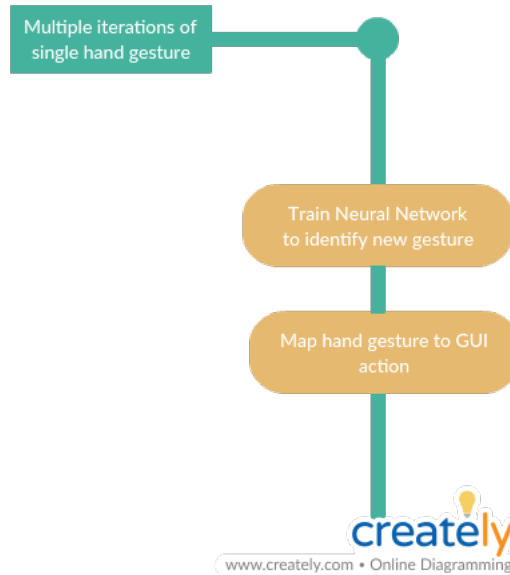
Input: Toggle button

Process: Will stop/start the daemon in the background

Output: The daemon shall be stopped/started.

5.2 Add new gestures

The user should be able to add new gestures by recording the hand movement and mapping the GUI feature to the current Hand gesture.



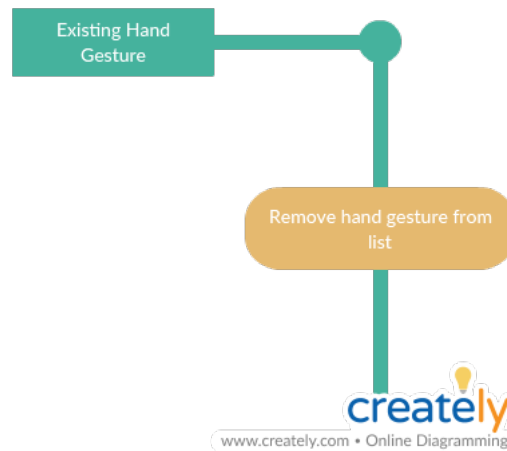
Input: Multiple Iterations of a gesture via Camera; Action to carry out

Process: The software will train the Neural Network to identify the new hand gesture and map the hand gesture to the Command given as input

Output: A new mapping from the hand gesture to the command will be added

5.3 Remove gestures

The user can remove the existing gestures

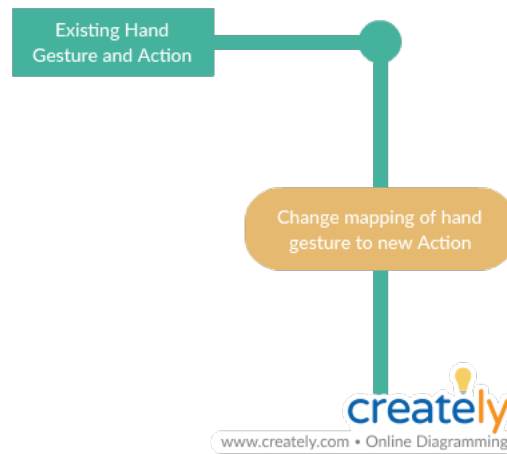


Input: A hand gesture

Process: The software will remove the mapping of the input hand gesture from the list of hand gestures

Output: The hand gesture is removed

5.4 Modify Gestures



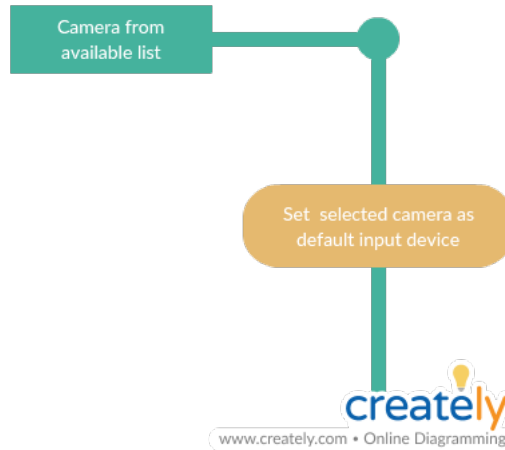
Input: Existing hand gesture; Action to execute

Process: The software will change the mapping of the input hand gesture to the input Action

Output: The hand gesture is modified

5.5 Choose webcam

For systems with multiple webcams connected the user must be able to choose which webcam the application must use.



Input: A camera detected by the daemon

Process: The software will set the selected camera as the default input for images

Output: The daemon will now watch for gestures through the selected camera when activated

6 Other Nonfunctional Requirements

6.1 Performance Requirements

The product must be fast enough to recognize and execute the hand gesture such that the action becomes fluid. The product must be responsive all the time and multiple hand gestures shown rapidly must be processed effectively.

6.2 Safety Requirements

No other application or user should access the camera feed.

6.3 Security Requirements

The camera feed must not be stored in the system. No one should be able to tap into the camera feed or be able to extract information from the camera.

6.4 Software Quality Attributes

Our product must be adaptable to all lighting conditions, poor webcams, and poor system capabilities. The product will be easy to maintain and review for other people. The product will run on newer updates of the operating system without any glitches.

7 Other Requirements

The product can be reused by other developers to create and maintain their own products. The GUI of the application must work with all languages so that people from all backgrounds can work seamlessly.

Bibliography

- [1] REAL TIME HAND GESTURE RECOGNITION SYSTEM FOR DYNAMIC APPLICATIONS , Siddharth S. Rautaray , Anupam Agrawal, 2012