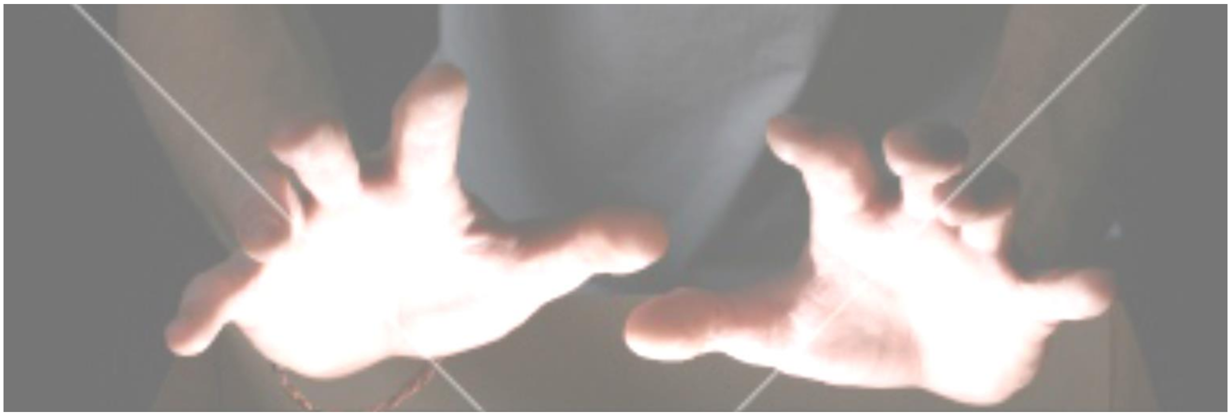
Software Requirements Specification

for

Magic Hands

`



Version 1.0
Magix Lab
October 14, 2013

Prepared by: Nathan Tung

Table of Contents

1. Introduction

1.1 Purpose

This **Software Requirements Specification (SRS)** document provides details and information on the Magic Hands system, version 1.0. This document will describe the system's important features, how they function, and explore the design behind the **User Interface (UI)**. The SRS will refer to **hardware**, **software**, and external factors involved in the Magic Hands system and explore the constraints and requirements around it.

1.2 Document Conventions

Some technical terminology that the reader might not recognize will be listed in bold and defined in the Appendix (Glossary). Words or phrases that are crucial to the operation of Magic Hands (such as the objective) may be italicized for emphasis.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for all people who will use, design, or manage the Magic Hands system. An emphasis will be placed on the user and the developer, but this document is meant to interest and inform all individuals involved with the Magic Hands project.

Readers interested in the technical development of Magic Hands should read the entire document. It is important for developers to understand the purpose of the Magic Hands system (starting in Section 1.4), all of its required features, and the assumptions and constraints surrounding its design.

Readers who are more concerned with the practical use of Magic Hands can skip to Section 3 for a very general idea of what the application is supposed to do. For more background information on the purpose and layout of this project, read Sections 1.4 and 2.1.

1.4 Project Scope

The goal of Magic Hands is to allow consumers and professionals alike to *interact with their computers using only body gestures*. The project will be built using a **Microsoft Kinect Sensor**, which functions as a sensor to detect gestures. The Magic Hands software will then collaborate with the Kinect Sensor in order to read in and distinguish between various gesture inputs.

The system will *recognize a large variety of default gestures that link to basic computer commands* while *allowing the users to link their own recorded gestures to specific computer commands*. All meaningful gestures should be detected and trigger an event instantly, without a delay for processing the computer command. At the same time, the system automatically separates different gestures performed consecutively and ignores gestures that are unrecognized or done unintentionally.

The project's initial short-term objective is to *be able to support presentation applications* like Microsoft PowerPoint; its second short-term objective is to *have browsing support for surfing the web using Mozilla Firefox and Google Chrome*. Its long-term objective is to *enable users to interact with computers naturally and efficiently without risking wrist strain or injury*.

1.5 References

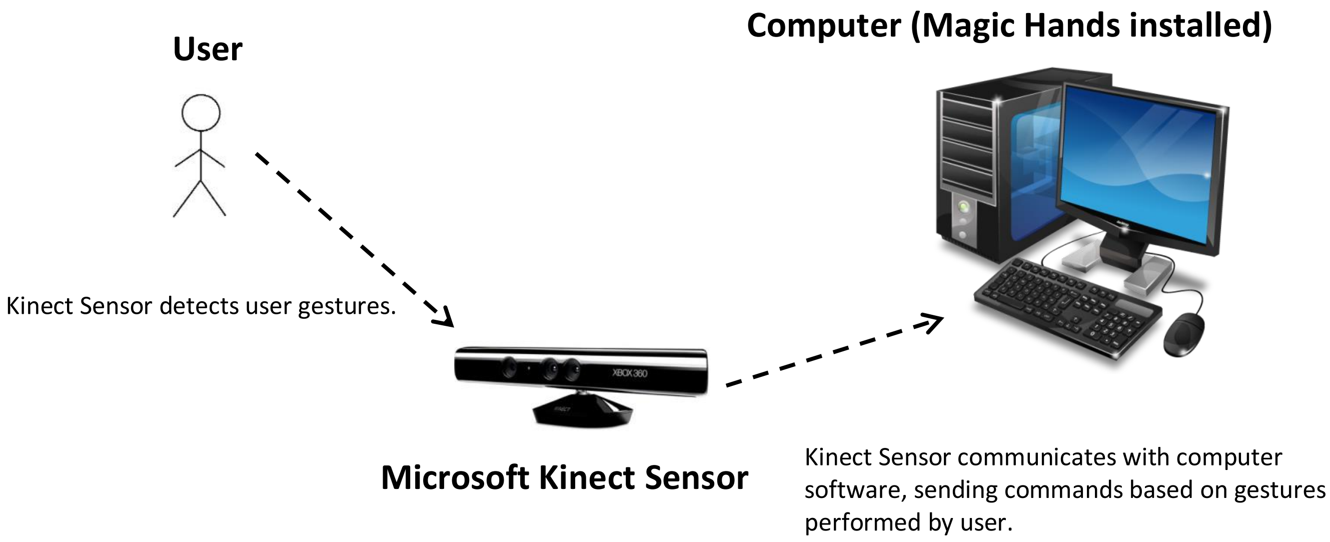
IEEE Guide to Software Requirements Specifications, IEEE Standard 830, 1984.

R. Aasen *et al*., "Software Requirements Specification for SplitPay," ZILDOR, Inc., Feb. 2011.

2. Overall Description

2.1 Product Perspective

Magic Hands is a new, self-contained (standalone) product designed under Magix Lab at UCLA. Since Magic Hands depends on a local sensor that draws information from a local database, there is not necessarily a need for internet functionality or connectivity. Instead, the project comprises the Magic Hands software on a computer and a Kinect Sensor to communicate with the software. This setup is shown in the diagram below.



Kinect Sensor detects user gestures.

Kinect Sensor communicates with computer software, sending commands based on gestures performed by user.

2.2 Product Features

Essential Features

1. User Account Creation
   - At software installation, user specifies an administrator account/password
   - At login screen, user can create an account/password with limited access
   - Users customize their own settings/preferences, while administrators can modify any user's settings/preferences

2. Administrator Control Panel
   - Allows control over all user accounts
   - Allows system settings to be modified

3. Sensor On/Off Function
   - Kinect Sensor automatically turns on when user performs meaningful gestures
   - Kinect Sensor automatically turns off after a set amount of time without input

4. Gesture Recognition Algorithm
   - Software matches user's gestures with those stored in local database, where the closest computer action is performed
   - Gestures that are messy, unintentional, or are not in the database are ignored
   - An amount of time (1 second) is specified for user to make a single gesture
   - Works with high accuracy
     - 95% accuracy for preloaded gestures, 80% for custom gestures

5. Virtual Interface Keyboard
   - User can "type" into their computer by using a software keyboard that responds to specific gestures; no actual typing
   - Can be easily accessed by a specified default gesture

6. Presentation Menu
   - User selects PowerPoint or presentation file to open
   - By default, hand swipe gestures go back and forth between slides

7. Browse Internet Menu
   - User selects either Firefox or Chrome by signaling in a direction
   - User can visit a website via the virtual keyboard or by saving websites to another custom gesture
   - Moving arm up and down can scroll page in that direction, opening and closing both arms can zoom in and out of the page, and so on

8. Settings Menu
    - Users can customize their own settings/preferences
    - Users can change their password (and optionally delete their account)
    - Users can add or modify their own existing custom gestures
    - Can be easily accessed by a specified default gesture

9. Help Menu & Online Help Forum
    - Users can sort through topics that provide help or information on specific aspects of the software (also found in text-format in readme file)
    - Users can discuss problems or ideas with other users or project employees
    - Can be easily accessed by a specified default gesture

10. Gesture History
    - Show user a certain number of their last recorded gestures for security reasons

11. Restore Factory Settings
    - Remove all custom gestures in case of emergency

<u>Extra Features</u>

1. Voice/Face Recognition
    - In addition to using the password for logging in, the software can match the user's face or voice for extra security (administrator can turn on and off)

2. Interactive Tutorial
    - Provides first-time users with a walkthrough on using Magic Hands

For more detailed information on features, skip to Section 3 (System Features) in the SRS.

2.3 User Classes and Characteristics

Magic Hands is a very broad type of software, so there is essentially only one class of users. With a few exceptions in how the software can be utilized, users are generally expected to use this system in the same way. Some characteristics defining the different types of usage will be summarized here.

1. Light/Heavy Custom Gesture Recording
    - Few custom gestures recorded: default gestures are detected quickly/accurately
    - More custom gestures recorded: more time spent distinguishing between gestures

2. Basic Office Use
    - Used mostly for Microsoft Office products, such as making presentations, or browsing the internet with Firefox or Chrome
        - Relatively easy to streamline

3. Complex Entertainment Use
    - Used for movies or games that use high resources
        - Change gesture recognition algorithm to detect movements more quickly
        - Enable Magic Hands to run on low-resource mode (strip down UI) so games will run more smoothly

2.4 Operating Environment

Magic Hands will be restricted to the Microsoft Windows operating system only, or specifically Windows 2000/XP/Vista/7/8. The software is simple and resource-effective, so there are no specific hardware requirements except for 1 GB RAM and 5 GB hard disk space (gestures will need to be stored in some local database). The project will borrow from the **Application Programming Interface (API)** used by Microsoft Kinect.

In order to use the PowerPoint presentation with gesture support, Microsoft Office 2000 or higher is required. In order to browse the internet, Mozilla Firefox (for example, version 20 or higher) or Google Chrome (for example, version 30 or higher) must be installed. In addition, **.NET Framework 2.0** or higher is required run Magic Hands.

2.5 Design and Implementation Constraints

The biggest constraint for Magic Hands is the Microsoft Kinect Sensor and Windows OS requirement. Magic Hands should work for many older versions of Windows, so it is important that the design is **backward compatible**.

Another constraint is the necessity of Magic Hands to respond to gestures "on the fly." That means the algorithm must process quickly and efficiently without hogging system resources.

Most importantly, the purpose of Magic Hands is to make interacting with computers natural and efficient without chance of injury. Therefore, the software behind the Kinect Sensor must be able to detect gestures without forcing the user to strain him or herself in order to elicit a computer response. The algorithm must be sensitive and able to distinguish nuances between different gestures.

2.6 User Documentation

Since Magic Hands attempts to make interacting with computers easier, this system will be beginner-friendly and easy to master (say, less than 2 hours of training). To aid in that process, the application will deliver several different methods of training: Help Menu, Online Help Forum, and the Interactive Tutorial. A **readme** text file will also be included in case the user wants to view a physical copy of the Help Menu.

The Help Menu is a simple help dictionary for the different aspects of Magic Hands. The user can call up the Help Menu at any time by presenting a default gesture. From there, the question at hand can be selected from a list of topics or searched using a search bar.

Users can also use the Online Help Forum to talk with either project employees or other users. Discussion may include (but is not limited to) questions, suggestions, or comments.

Finally, the Interactive Tutorial allows users to go through all the basic functions of Magic Hands within a timely manner. The user will become familiar with default and custom gestures before the tutorial is over.

2.7 Assumptions and Dependencies

The Essential Features discussed in Section 2.2 are vital to the basic function of Magic Hands. On the other hand, the Extra Features are beneficial but not necessary; they may or may not be completed depending on availability of time and resources.

In addition to the API and .NET Framework mentioned before, the development of Magic Hands will depend on the available of certain software tools like Kinect for Windows **Software Development Kit** (SDK) and Developer Toolkit, as well as **Visual Studio**. Kinect for Windows SDK contains drivers that are essential for using the sensor on a Windows computer. Kinect for Windows Developer Toolkit provides resources for the developer to implement face tracking and the User Interface. Visual Studio may be necessary as an **Integrated Development Environment** (IDE) for designing a majority of the code that makes the Magic Hands software function.

3. System Features

As mentioned before, we have Essential Features and Extra Features. The former are necessary for the Magic Hands system to work as expected, while the Extra Features are, as the name suggests, optional. They will be added to the system for a better overall experience only if time and resources allow.

3.1 User Account Creation

During software installation, the user is prompted to enter a username and password for creating an administrator account. The password must be entered twice for confirmation purposes. During the system boot-up/login screen, new accounts with limited access can be created in a similar fashion. The usernames and passwords must be **encrypted** or otherwise safely stored locally.

3.2 Administrator Control Panel

The control panel here is responsible for global settings and preferences across the entire system and all accounts. Administrators can modify the settings and preferences of all users. No limited account access permitted.

3.3 Sensor On/Off Function

It is vital for the Microsoft Kinect Sensor to know when to record and when to stay idle for power-saving purposes. The sensor should automatically turn on when the user inputs meaningful gestures. It should also turn off (or go idle, since the sensor is always “on”) when no motion is detected for a set amount of time, say 1 minute.

- If no motion is detected for 1 minute, go idle
- Otherwise, if a meaningful gesture is detected, sensor goes active and begins interpreting input gestures

3.4 Gesture Recognition Algorithm

When the user inputs a gesture, the system runs through the gesture database and finds the best match. If there is no good match, the motion is either meaningless (has not been yet been linked to a computer command) or unintentional and is disregarded. A set time (default 1 second) is allotted for each individual gesture. This algorithm runs with very high efficiency and accuracy, especially for default gestures.

- User inputs gesture; systems checks if match exists
- If close match exists, the system instantly calls that computer action
- Otherwise, action is considered meaningless or unintentional

3.5 Virtual Interface Keyboard

If the user needs to input some characters, he or she can toggle the virtual keyboard by indicating a specific preset gesture. This software keyboard allows the user to “type” using more motions. Since Magic Hands strives to make computer interaction less strenuous, no actual typing or wrist action will be used here.

3.6 Settings Menu

All users can customize their own settings and preferences here by first indicating a specific preset gesture. Users can also change their password here or delete their account, if the administrator so allows. Custom gestures and their computer commands can be recorded, modified, or deleted here.

3.7 Presentation Menu

After signaling for the Presentation Menu, the user indicates which PowerPoint presentation file for which to apply gesture-control. Hand swipe gestures are, for example, preset to scroll through slides in the direction applied.

3.8 Browse Internet Menu

After signaling for the Browse Internet Menu, the user indicates either Mozilla Firefox or Google Chrome to use. In order to visit a website, users can signal for the virtual keyboard and “type” the website address that way. Users can also save websites to automatically open based on another custom gesture. Gestures like moving their right arm up and down, for example, can scroll the page, whereas expanding both arms can zoom into the web page.

3.9 Help Menu & Online Help Forum

When a user needs help regarding Magic Hands, he or she can indicate a specific preset gesture to bring up the Help Menu and Online Help Forum together. Using the Help Menu, topics relating to the problem at hand can be searched and sorted in order to present a solution. If no solution is available for the intended question, the user can ask project employees or other users on the forum for help or suggestions.

3.10 Gesture History

The system presents the last gestures inputted by the users as well as their associated actions. This is here both for security reasons and for the user to see how well the gesture detection sensor has been calibrated.

3.11 Restore Factory Settings

In case of emergency, users can restore their account to factory settings, meaning all their custom gestures will be removed. The administrator can restore the entire system to factory settings, which works like a reformat or a clean install.

3.12 Voice/Face Recognition (Extra)

Since one possible constraint is password security (the password might not be complex enough or another user might imitate the gesture), the Magic Hands system may utilize voice and/or face recognition in order to make user accounts more secure. Microsoft Kinect Sensor is able to take advantage of both sound and vision.

3.13 Interactive Tutorial (Extra)

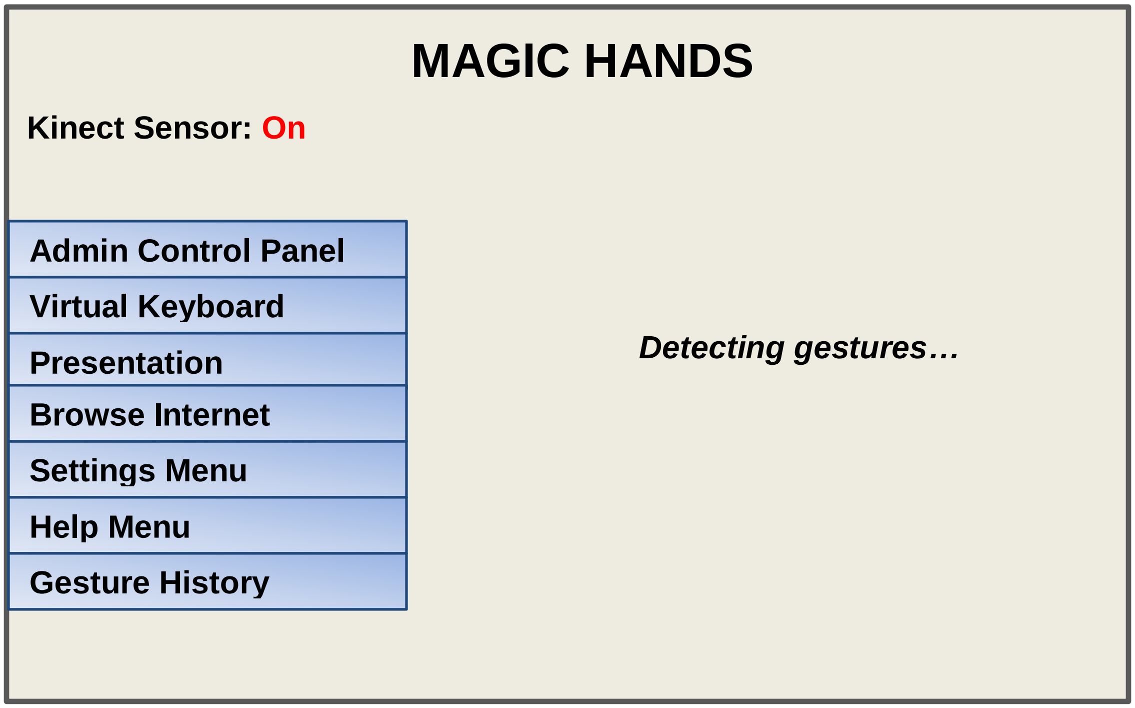
When a user logs in for the first time, he or she is prompted with an opportunity to take a tutorial that demonstrates the application’s features and tools. The tutorial is interactive, as it helps the user get used to the default gestures and offers an opportunity to start recording custom gestures.

4. External Interface Requirements

4.1 User Interfaces

Since Magic Hands is implementing a typing-and-mouse-free environment, the user interface is going to be less cluttered but more informational so that the user knows what options and commands he or she can call using gestures. Other than several settings and menus to display, the main purpose of this system is to allow gesture-control in other applications, so the UI is rather simple.

The main menu might look something like the following:



MAGIC HANDS

Kinect Sensor: On

Detecting gestures…

All other menus behave the same way and therefore will have a similar UI.

4.2 Hardware Interfaces

Magic Hands is an application meant to read gestures from the Microsoft Kinect Sensor. Apart from the Kinect USB Adapter, the computer, and the Kinect Sensor itself, no other hardware is specifically necessary.

The Kinect Sensor is supported by Microsoft, where its hardware interface can be found as well.

4.3 Software Interfaces

Gesture input is read through the Kinect Sensor. The sensor then relays that information to the Magic Hands software via USB interface. This occurs every time the Kinect Sensor detects some kind of movement. Magic Hands receives the gesture data then matches the gesture to some computer command.

That computer command data is directly transferred on the software level between different applications and translated into a computer action. These applications include Microsoft PowerPoint as well as Mozilla Firefox and Google Chrome. This is done through either Microsoft controls or through **macros**. The translation of gesture data to action occurs every time Magic Hands recognizes a meaningful gesture.

4.4 Communications Interfaces

Since Magic Hands itself does not utilize internet or network communication, there is no special communications interface. For browsing the internet, this system operates using the communications interface of Firefox and Chrome. HTTP standards will be used when accessing web pages.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Magic Hands must be able to read in a gesture from the user, match it with some command, and translate that command into action in a very streamlined manner and almost instantaneously (in a fraction of a second). This may become a problem when the computer's resources are spent on running graphics-intensive software. The system must therefore be prepared to use low-cost algorithms and a simple UI to enhance the application's efficiency.

5.2 Safety Requirements

This application does not interact directly with the internet, so safety concerns are largely mitigated. Even so, safety checks should be put in place on the target applications (such as Microsoft Office) so the user does not use it to manipulate or break other software, whether for accidental or malicious reasons.

5.3 Security Requirements

The complex gesture password should be encrypted and be accompanied with other security checks to maintain user integrity. Encrypted passwords should be stored in a local database that is inaccessible from the outside. But unless the face or voice recognition feature is included, any third-party that obtains the gesture password by watching the user may be able to gain access to the user’s account. Fortunately, Magic Hands does not hold any sensitive information, so there is little to be gained by accessing another limited account.

5.4 Software Quality Attributes

The algorithm in Magic Hands that recognizes gestures should most importantly be accurate and reliable. The user should be able to depend on Magic Hands consistently for interpreting their motions and getting the same expected results time and again. The history log of gestures may be useful in determining how the system should respond.

Efficiency is another important aspect of Magic Hands, as the users should not have to wait for the software to load and thereby slow down their work. Magic Hands should be just as efficient as using a keyboard and mouse, but simply with less strain. To achieve this, the recognition algorithm must be optimized.

Flexibility and usability can be beneficial to the system users as well. By designing Magic Hands to be compatible with many important applications, the user will be able to use gestures to control an even larger portion of his or her computer. Even if the algorithm is efficient and accurate, the application itself (the UI) must be practical and easily learned by the user, whether for home use or for professional use.

6. Appendix

6.1 Glossary

.NET Framework – software framework or platform developed by and for Microsoft Windows

Application Programming Interface – library that specifies the relationships between different software components

Backward Compatibility – ability of a product to work with inputs from older technology

Encrypt – to encode passwords such that only unauthorized third-parties cannot read it

Hardware – physical components that make up an electronic system

Integrated Development Environment – software application that includes tools (like compilers, debuggers, etc.) valuable for computer programming

Macro – rules that specifies how certain inputs should be replaced with other inputs

Microsoft Kinect Sensor – motion-sensing device with a webcam meant for Xbox 360 and PCs

Readme – file containing important information about files in a computer software directory

Software – instructions written for the computer to translate into computer operations

Software Development Kit – tools that allow software developers to create applications for some software system

Software Requirements Specification – document specifying comprehensive requirements for a software system

User Interface – field where users can interact with the computer using inputs and outputs

Visual Studio – Microsoft environment (IDE) used to create applications and other programs