Thesis

Picture Passwords in Mixed Reality: Implementation and Evaluation

GEORGE HADJIDEMETRIOU

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

May 2019

UNIVERSITY OF CYPRUS
DEPARTMENT OF COMPUTER SCIENCE

Picture Passwords in Mixed Reality: Implementation and Evaluation

George Hadjidemetriou

Supervisor:

Prof. Andreas Pitsillides

Co-Supervisor:

Dr. Marios Belk

The individual thesis submitted for partial fulfillment of the requirements for obtaining the degree of Computer Science, Department of Computer Science, University of Cyprus

May 2019

Acknowledgments

Firstly, I would like to thank my supervisor Prof. Andreas Pitsillides for giving me the opportunity to undertake this project, his continuous support, encouragement and for allowing me to conduct in-depth research in fields that I am interested in and build the foundations on my academic and professional career.

I would like to sincerely thank Dr. Mario Belk for his eagerness, great assistance, guidance and support in order to accomplish the development of this project and the writing of this report. This project would never be possible without his help, support and incredible sense of humor. Our cooperation was truly an inspiring experience that helped build the initial steps of my career and future and I hope that our cooperation keeps unfolding new paths towards great research results.

I am also grateful to Argyris Constantinides for his assistance, support and motivating attitude during this journey.

Moreover, I would like to thank my friend and classmate Antrea Chrysanthou, for her constant support and for always helping me overcome many difficulties that came up across the four years of my studies.

Finally, I would like to dedicate this work to my beloved family and friends who with their support, belief, encouragement, support and patience made this four-year journey an unforgettable experience. For that, I will always be grateful to them.

Recognitions

The following papers were authored based on the expertise and the findings that were obtained through this bachelor thesis:

1. Hadjidemetriou, G., Belk, M., Fidas, C., & Pitsillides, A. (2019). Picture Passwords in Mixed Reality. *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. New York, New York, USA: ACM Press. (published)
2. Hadjidemetriou, G., Fidas, C., Belk, M., & Pitsillides, A. (2019). On Scaffolding the Security of Graphical Passwords through Playful Interaction Experiences in Mixed Reality. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play - CHI PLAY '19*. New York, New York, USA: ACM Press. (submitted)
3. Fidas, C., Belk, M., Hadjidemetriou, G., & Pitsillides, A. (2019). Influences of Mixed Reality and Human Cognition on Picture Passwords: An Eye Tracking Study. *FIP TC13 Human-Computer Interaction (INTERACT 2019),* Springer-Verlag. (submitted)

Abstract

The past few years, saw a great increment in the use of digital devices, such as desktops and mobile devices in order to offer much needed help to everyday life tasks so that their completion can be made easier. Along with that, came the recent and sudden shift towards immersive technologies that enhance or replace a user's reality using computer generated objects. Such technologies are Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR).

This thesis targets methods of accessing services on Mixed Reality systems, that embrace hand gesture-based interaction with virtual keyboards which allow users to login using a textual password. This task is considered difficult and time demanding since users are forced to type complex and long passwords in a virtual keyboard using specific hand gestures and movements.

The purpose of this thesis was to create a new gesture-based authentication system which is based on Microsoft Windows 10 Picture Gesture Authentication (PGA) that can be used as an alternative authentication method in Mixed Reality. The system allows the users to draw three gestures (dot, line, circle) on three points of interest (POIs) of their choosing, that serve the purpose of each user's password, on an image that works as a cue for remembering the POIs and the gestures.

In addition, eye tracking mechanisms were implemented in order to capture the users' fixations so that we could analyze their visual behavior while using our system so that we could identify whether this new feature will have a positive impact on the users of Mixed Reality systems.

Finally, an alternative graphical password authentication system has been researched and developed in order to implement more methods for allowing such interactions. This alternative system allows a selection of five items from an array of items that serves as a Recognition-based authentication system.

Table of Contents

Chapter 1

Introduction

1.1 Thesis Overview

Modern technologies, which tend to lean towards using immersive systems in order to enhance our everyday productivity and entertainment, have introduced many new concepts of interacting with computers. Technologies like VR, AR and MR, are targeted to not only entertain users, but also help them overcome many physical as well as emotional problems. So, as a result, interaction with said systems, is of utmost importance because easy interaction modules should help even the most novice of users achieve everyday tasks.

Having said that, one of the most common tasks a user faces with any sort of electronic device is the one about authenticating themselves as the rightful owners of an account. This task has become trivial on desktop and mobile devices, but still has a long way to go before achieving the same state on immersive head mounted displays. Researches have to tackle this problem in order to introduce alternative authentication methods that make the experience more user friendly but also maintain the security of a system.

Our aim was to research and implement such new ways in Mixed Reality headsets and more specifically Microsoft's HoloLens which introduced a whole new spectrum of visualizing computer-generated holograms. As the current authentication module consist of a tedious and time-consuming task, we implemented two new authentication methods for this device, which are inspired from already-researched models that exist on conventional desktop environments.

1.2 Problem Statement

As mentioned before, immersive headsets are rapidly being introduced to the public but still lack a usable authentication system. Since these headsets offer a simpler interaction method by either utilizing remote controls or hand gestures, we tried to tackle this problem by simplifying user authentication without compromising security. As more research has been directed towards Virtual Reality headsets and almost none towards Mixed Reality headsets, we targeted the latter.

For that reason, we attempted to implement two new authentication modules for Microsoft's HoloLens. The first one resembles a Recall-based authentication system, the Windows 10 Picture Password, mapped for Mixed Reality headsets. The second system implements a Recognition-based authentication system similar to ImagePass[15], which requests from the user to recognize their five-image selection from an array of many images. Our expectations from these systems, is to statistically improve the usability of the authentication system in HoloLens without affecting negatively the security of the device. Also, we attempt to set the foundations for further research of such systems and finally introduce the best authentication system or immersive headsets.

1.3 Mixed Reality

Mixed Reality is term that was originally introduced by Paul Milgram and Fumio Kishino in a paper they wrote in 1994 [16]. As they stated, "it is a particular subset of Virtual Reality

related technologies that involve the merging of real and virtual worlds somewhere along the "virtual continuum" which connects complete real environments to completely virtual ones." By extension to that, MR is a new environment, where physical and virtual objects co-exist and interact in real time, by anchoring the virtual objects to the real-world objects and allowing the user to interact with the combination of both.

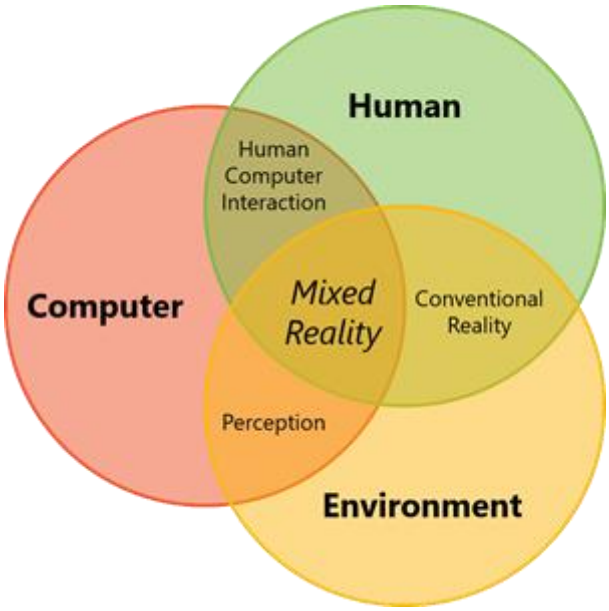

Figure 1.1 – The aspects of Mixed Reality

Through the years, the area of Human and Computer Interaction has been thoroughly studied by researchers, with the main computer input methods being keyboard, mouse touch, voice, etc. However, technological advancements in recent years, have allowed developers and engineers to escape the conventional computer input methods and begin developing and perfecting perception input methods. Such methods include head tracking, surface and boundaries tracking, environmental sounds, location, etc.

MR has found use into several applications such as:

- IPCM (Interactive Product Content Management) – the transition from classic product catalogs to more interactive 3D digital ones which show a more realistic representation of products
- SBL (Simulation Based Learning) – the transition from traditional electronic-learning to simulation-learning, which is a more interactive learning method
- Military Training – where realistic simulated environments are possible through HMDs (Head Mounted Displays) for better adjustment of the military personnel to realistic scenarios in the battlefield.
- Remote Working – where real-time remote communication between co-workers is possible no matter their physical location, with sharable workspaces and environments so that team synchronization and coherence can be achieved with ease
- Functional mockup – mockup building applications that can be utilized in industries so that virtual models can interact with physical objects and industries can see live preview of the product's behavior
- Consciousness – the hypothesis that a hybrid of MR and VR can build the foundations of the transfer of human consciousness to a digital form

Finally, Mixed Reality can be deployed on many kinds of technology equipment. The most famous of them are Mobile Phones and Tablets (though spatial capture mechanism have not yet been developed on mobile devices), Head-Up Displays (plane cockpits), Head-Mounted Displays (Microsoft HoloLens) and Computers.

1.4 Motivation

Following all of the above, our goal was to improve the authentication experience of all the users in Mixed Reality headsets. To achieve that, we decided it was best to implement graphical passwords.

Currently, Microsoft's HoloLens is mainly used in industries and business in order to increase the productivity, team work and coordination of employees. Personal use is not favorable even though it is still available for commercial use. In our personal experience, while operating the HoloLens, we found it extremely hard to authentication ourselves into the system and the available applications. That happens because the current authentication modules imply using a virtual keyboard and by executing air-taps on the keyboard buttons that we gaze, in order to enter passwords. Moreover, no previous research could be found on alternative authentication modules, so this was a chance to research this field.

1.5 Scope of the Thesis

The scope of this thesis is to take into consideration all the prementioned factors and based on research that was conducted for graphical password authentication on desktop devices, develop new mechanisms that will improve usability of these systems. By developing these new methods, we hope to help the research community to further attempt and tackle this problem but also help ourselves to further research this field and, in the end introduce innovative ideas for new authentication methods in such systems.

Chapter 2

Background Theory

2.1 Introduction

This chapter will discuss about conventional and graphical user authentication and it will explain, in depth, what user authentication in mixed/augmented/virtual reality is and what methods are already available. Furthermore, it will talk about Mixed Reality (MR), what it is and how it can be used, since it is usually confused with Augmented Reality (AR).

The main authentication method used in modern systems are text passwords where a user simply has to type in their registered combination of username and password in order to authenticate themselves and gain access to the required service/system. The most common type of textual password is a sequence of numbers, characters and special characters and also have a length between 8 and 24 characters. Even though they are highly accepted by society and researchers as a safe method of authentication [1, 22], they still are vulnerable to different attacks. Along with text passwords, other authentication methods are available. An example are graphical passwords. They are authentication systems where the user makes a selection from an array of images, from a Graphical User Interface, in a specific order that they determine. In general, graphical passwords are easier to remember that

conventional textual passwords, since recalling a complex string of characters is much harder than selecting images that each user has set as their password. Each image can be analyzed differently by each user, so that they can remember it by the specific characteristics that they have extracted from them. Moreover, graphical passwords may be more secure than text passwords, since hacker by utilizing a dictionary attack, which is using a very large list with possible password combinations in order to gain illicit access to an account, can gain access to an account. On the other hand, graphical passwords cannot be so easily cracked, because a hacker must try all the possible image combinations in order to find the correct one, and that can be very time consuming.

2.2 User Authentication

2.2.1 Conventional

User Authentication (UA) is a task performed in almost all human-to-computer interactions. Traditionally, it is a simple textual username, serving as the user's ID, and a password combination. Its purpose is to verify a user identity that it is legitimate and indeed has access to the requested resource. Through the years, these simplistic login systems have evolved in manner that offers more security and complexity in terms of password hacking. UA is composed of three main factors:

1. **Knowledge factors** which are all the required things the user must know in order to successfully login. For example: usernames, passwords, etc.
2. **Possession factors** which are the required elements the user must have in their possession in order to successfully login. For example: ID cards, one-time passwords (OTPs), etc.
3. **Inherence factors** which include all the required biological characteristics the user must have, inheritably, in order to successfully login. For example: biometrics such as fingerprints, facial and voice recognition, etc.

A user's authentication process consists of three tasks:

1. Accomplishing connection between them and the machine/service they want to access
2. Verifying their identity
3. Successfully approving their identity so that the machine/system can authorize access to the user

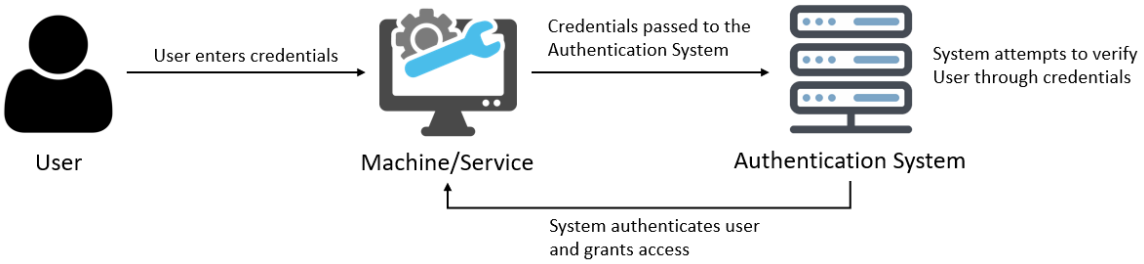

Figure 2.1 – General User Authentication Process

2.2.2 Graphical

Graphical User Authentication schemes have been around for some time, offering to replace the conventional text-based passwords in order to provide an easier login experience for the users and they can be grouped in *recall-based* and *recognition*-based. Recall-based authentication schemes require that the users can remember and recreate a drawing that they have entered during their account creation. Such password schemes are:

(i) Windows 10 Picture Gesture Authentication (PGA) [12, 24], where a user can set a background image and draw any combination of the three available gestures (Dot, Line, Circle) on any three points on the image that will serve as their password

(ii) Draw-a-Secret (DAS) [11] where the user draws a picture on an NxN grid and the password is the ordered sequence of cells where the user has drawn over

(iii) BDAS [8] which is the same as DAS with the use of a gridded background image

(iv) PassPoints [20] where the user selects a sequence of N positions on an image that will serve as their password

(v) Cued Click Points (CCP) [6] where the user selects one position on N images instead of the PassPoints system.

Recognition-based authentication shcemes require that the users can recognize and select pre-selected images from a set of images. Such password schemes are:

(i) PassFaces [5] where the user must select four faces from a grid of faces during registration and recognize them and select them during login

(ii) DejaVu [7] where the user must recognize and select the images from a portfolio they specify which are mixed up with other random images

(iii) ImagePass [15] where the user selects N images from an array of images, in a specific sequence, and they later have to recognize and repeat those images in the same sequence from an array of 12 images.

2.3 Graphical User Authentication in MR, AR and VR

In the past few years, Mixed Reality (MR), Augmented Reality (AR) and Virtual Reality (VR), have seen a major improvement technology-wise, as well as consumption- and development-wise. Physical keyboards have been replaced by virtual ones in such systems, therefore using textual-based passwords as an authentication method is a tiresome and time-consuming task [1, 9, 19]. For that reason, new authentication schemes have been studied that render text passwords unsuitable in terms of usability and user acceptance in MR/AR/VR systems. Such authentication schemes are:

(i) [23] which implemented a 3D password, pattern lock and PIN system authentication schemes in VR. By conducting two experiments to check the schemes in terms of usability and security against shoulder-surfing attacks, they found out that the 3D password offers the most security out of the three schemes, although it offers less usability. Moreover, the usability on both the pattern lock and the PIN system was approximately the same. In terms of security, the pattern lock is more secure against brute force attacks and the PIN system offers more security against shoulder surfing attacks.

(ii) [17] which attempted to leverage users' personal views by implemented a password manager application that works by utilizing both an AR Head Mounted Display (HMD) and a browser extension. By changing the browser's UI to show a unique QR code that represents the currently viewed website and then command the HMD to scan the code and find stored passwords that match that website. they achieved great protection against shoulder-surfing attacks since everything is displayed in the user's personal display.

(iii) [10] which also implemented PIN- and pattern-based authentication schemes in VR, that can be used in MR as well, and then conducting a user study with combinations of three sizes of grids and three different input modules. After conducting a user study, the results showed that the perceived security and usability was dependent on the authentication method and comparison between the different authentication methods in term of security, measured significant difference in some occasions rendering them more secure than others.

(iv) [21] which explored alternative options to improve the Google Glass authentication mechanism that suffers from shoulder-surfing attacks by presenting two PIN-based authentication schemes, one being voice-based, and the other being touch-based. After conducting a user study, the results showed the significant improvement of the login success rates as well as better perceived security and usability by the users in comparison with the build-in authentication system.

(v) [18] which presented a Google Glass authentication system that uses biometric data, such as the sound's conduction through a user's skull and a microphone in order to analyze the characteristic frequency of a user while talking in order to authenticate them. With the conduction of a 10-participant user study, the mechanism was proven stable even after putting off and on the HMD many times. Moreover, it had a very high accuracy on correctly identifying registered users and rejecting unregistered ones.

Concluding, many attempts have been made in order to improve user authentication in MR/AR/VR systems, although the ones utilizing a PIN-based solution and in general the usage of virtual keyboards deprecate usability, since research has shown that using virtual keyboards for text input are hard to work with.

Chapter 3

Tools, Technologies and Architecture

3.1 Introduction

This chapter will discuss the different technologies and tools that were used in the development of the two systems. Moreover, the selected system architecture will be explained

3.2 Technologies and Tools Used

In order to apply the selected architecture model which will be explained later, we used a selection of tools that gave us the required capabilities. These tools are:

(i) ***Microsoft's HoloLens*** is Microsoft's endeavor to implement the Mixed Reality aspect into our lives. It is a fully untethered, see-through holographic computer in the form of a wearable headset; one that the user wears and instantly is experiencing MR. HoloLens comes with semitransparent holographic lenses [Fig. 3.1] which generate multi-dimensional holograms that blend in with the user's surrounding environment and will be seamlessly attached on surfaces and objects. A high-level explanation of how holograms are projected is that light travels from the top of the lenses down, and at some point, the light rays escape from the lenses into the user's

eyes. It features a full spatial surroundings mapping which detects the surfaces in your environment and creates collider surfaces with them in order to allow the prementioned attachment of holographic objects onto those surfaces. Moreover, HoloLens comes packed with a series of sensors [Fig 3.2] such as an inertial measurement unit, an ambient light sensor, one depth measuring camera and four environment understanding cameras which allow for the spatial mapping. It also comes equipped with a 2-megapixel camera for capturing photos and videos. HoloLens is, without a doubt, a unique device that can be used to simplify day-to-day tasks and make dull experiences more enjoyable but also generate brand new experiences. Such applications are:

- Remote Instructions – The user can have internet video calls with others with shared screen in order to receive assistance in different complex tasks such as repairing an electric appliance
- 3D Computer-Aided Design – The user can construct and design new products in the virtual world and see them in the physical world in real-size to get a grasp on how they look and feel in the environment
- Gamification of tasks – Monotonous tasks can be gamified and be turned into more interactive and interesting ones so that fatigue and dullness can be eliminated, and productivity and interest can be boosted
- Gaming – The world of virtual gaming is further expanded by introducing mixed reality gaming where the user can align game elements with their surrounding physical environment and make the experience more immersive and enjoyable
- Holographic Attractions and Entertainment – Users can visit secluded locations in a risk-free and travel-free experience by simply viewing a projection of the location around them through the HoloLens and be able to traverse the environment through Mixed Reality

Figure 3.1 – HoloLens see-through lenses

Figure 3.2 – HoloLens Sensors and Camera

(ii) ***Microsoft's HoloLens Clicker*** [Fig. 3.3, Fig. 3.4]which is a peripheral device used only paired with the HoloLens as an input method instead of using hand gestures. It allows an alternative way of interacting and controlling the holograms projected by the device. The basic functionalities that someone can do with the clicker are: click, click-and-hold, scroll and zoom.

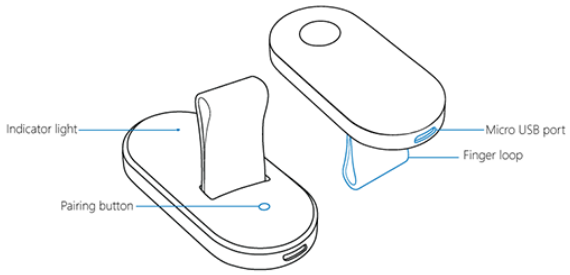

Figure 3.3 – Clicker

Figure 3.4 – How the clicker is held

(iii) ***Pupil Labs*** is a company that developed and offers a plug and play, open-source eye-tracking hardware and software suite. The hardware is a wearable headset [Fig. 3.5], like a pair of glasses, with mounted cameras underneath each eye for recording the eye's movements. The recording can be done using one of the available methods:

- monocular, which records and streams the user's gaze
- binocular, which estimates where the user is looking in 3D and their eye movement data

- Egocentric vision research, which records the user's field of view (FOV)
- Eye movement research, which records the user's eye movement data.

Pupil Labs' hardware, can be also attached to VR/AR headsets, including Microsoft's HoloLens [Fig. 3.6], so that research can be further enhanced by using their equipment and accompanying software for recording and analyzing eye-movement and gazing data. The available software are: i) Pupil Capture, which receives video and audio streams, detects the user's pupils, tracks their gaze, tracks markers in the environment that the user can set, streams data over network in real-time and records data; ii) Pupil Play, which is a media and data visualizer that works with Pupil Capture's recordings and allows for data visualization and export.

Figure 3.5 – Pupil Labs' wearable headset

Figure 3.6 – Pupil Labs' hardware attached on Microsoft's HoloLens

---

(iv) ***Unity 2018.2.8f1*** is a cross-platform game engine that gives the ability to developers to create interactive 2D and 3D experiences and is scripted mainly in C# but also supports other programming languages as well. Unity is the only game engine supported by Microsoft's HoloLens for developing applications and Microsoft offers for free a library, named HoloLens Tool-kit, with a vast variety of ready-to-use components for HoloLens. Such components are 3D objects, buttons and text-boxes that are used to create the user interface of an application.

(v) ***Unity HoloToolkit 2017.4.1*** is an open source collection of scripts and components that help developers create applications on HoloLens using Unity. It offers a huge variety of 3D objects which help you create interactive Mixed Reality interfaces and scenes and ready-to-use scripts that support the core functionality of HoloLens and can easily be expanded to include custom mechanics.

(vi) ***C#*** is a general-purpose programming language developed by Microsoft and is the primary scripting language for Unity and HoloLens and it can easily be configured for contacting APIs through the .NET framework that is supported only by C#.

(vii) ***PHP***, or also known as Hypertext Preprocessor, is a cross-platform programming language originally designed for web development but is also suited for server-side scripting such as Apache and IIS. In our case this was a very well-suited option since we wanted to create a simple and easily-configured API for storing and retrieving data from a database.

(viii) ***Python*** is a high-level and general-purpose programming language and it is the primary scripting language for the Pupil Labs' software and HoloLens plugin that was used for the eye-tracking data retrieval process. Its popularity, easy readability, huge availability of libraries and the community available, make the development process easier and smoother for even larger scale software.

(ix) ***MySQL*** is an open-source RDBMS (relational database management system) tha allows for development, administration, creation, maintenance and design of SQL databases. It is working on an Apache web server software and can be used for data storing, retrieval and management. This was a perfect option since we are using PHP as our web server scripting language and it could be easily configured to communicate with MySQL for all the purposes we needed it.

(x) ***Microsoft Visual Studio*** is an IDE (integrated development environment) and it is widely used for program development. It supports a plethora of programming languages and has C# as a built-in language. Its code editor supports IntelliSense which is a code completion component and a code refactoring tool.

(xi) ***Sublime Text*** is a source code editor that supports many programming and markup languages with added functionalities by using plugins. It is a simple code editor that provides similar functionalities to bigger IDEs such as auto-completion, syntax highlight and in-editor code building. In our development process we used it for PHP and Python scripting due to its simplicity and code highlight which was really helpful syntax-wise.

(xii) ***XAMPP Control Panel*** is a free and open-source cross-platform web server solution package which mainly consists of the Apache HTTP server and MySQL and allows for easy administration of MySQL through the phpMyAdmin administration tool.

3.3 System Architecture

The system we chose to implement is a replica of the already-existing Microsoft Windows 10 Picture Password Authentication system. For this reason, the architecture we chose to use is the three-tier model.

Three-Tier Architecture

A three-tier architecture model is mainly a client-server architecture where functional process login, data access, computer data storage and user interface are all developed and maintained as independent modules. The three tiers are:

1. Presentation Tier –This is the layer responsible for the front-end view of the system and consists of the user interface. The main function of this tier is to translate tasks and results in a form and view that the user can understand.
2. Application Tier – This tier is responsible for handling all the functionalities and logic that drive the presentation layer, establishes the communication between the user's choices in the interface with the underlying database for data retrieval and processes commands given by the user.
3. Data Tier – This is the layer responsible for storing and retrieving information from a database based on the commands that derive from the application tier. The data are accessed through API calls executed by the application layer. When a user is making a selection in the presentation tier, the application layer executes an API call based on that selection to the data tier which, in return, answers back to the application layer with the requested data.

By using a three-tier architecture, we take advantage of this architecture's capabilities in speed of development, since each part is developed independently, scalability since each tier can be easily expanded and managed and finally performance and availability since each tier can be individually optimized and checked for bugs and errors.

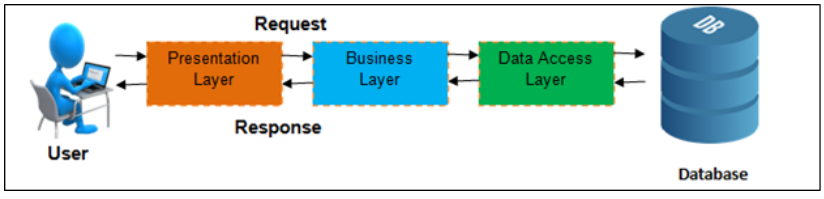

Figure 3.7 – Standard Three-tier Architecture model

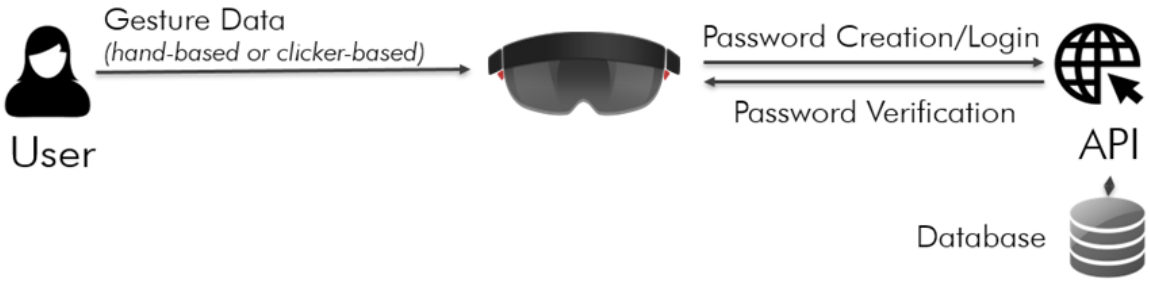

Figure 3.8 – Simple HoloLens Three-tier Architecture model

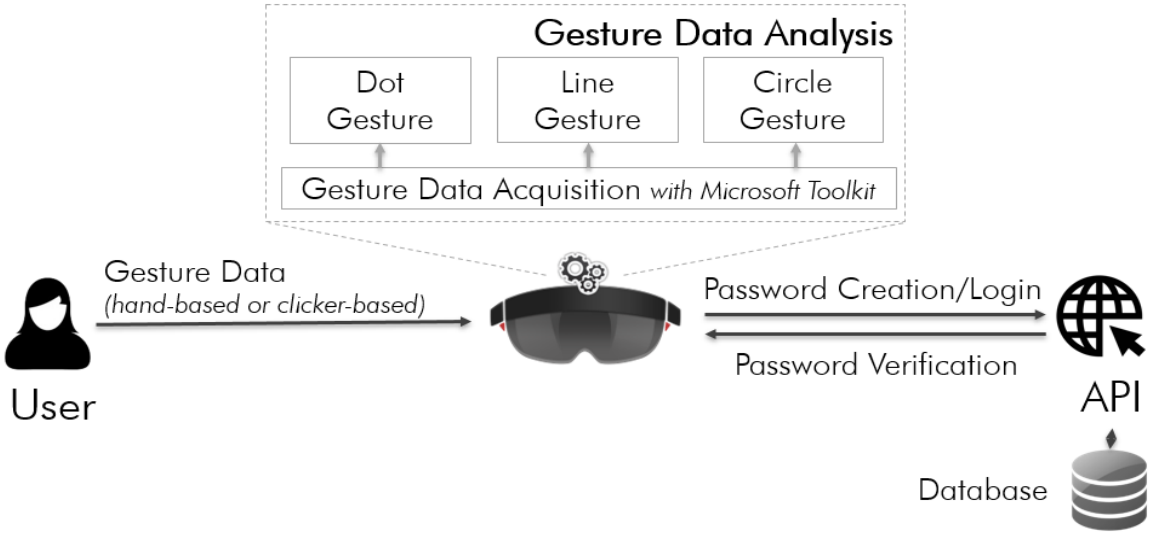

Figure 3.9 – HoloPass' (Recall-based) Three-tier Architecture model

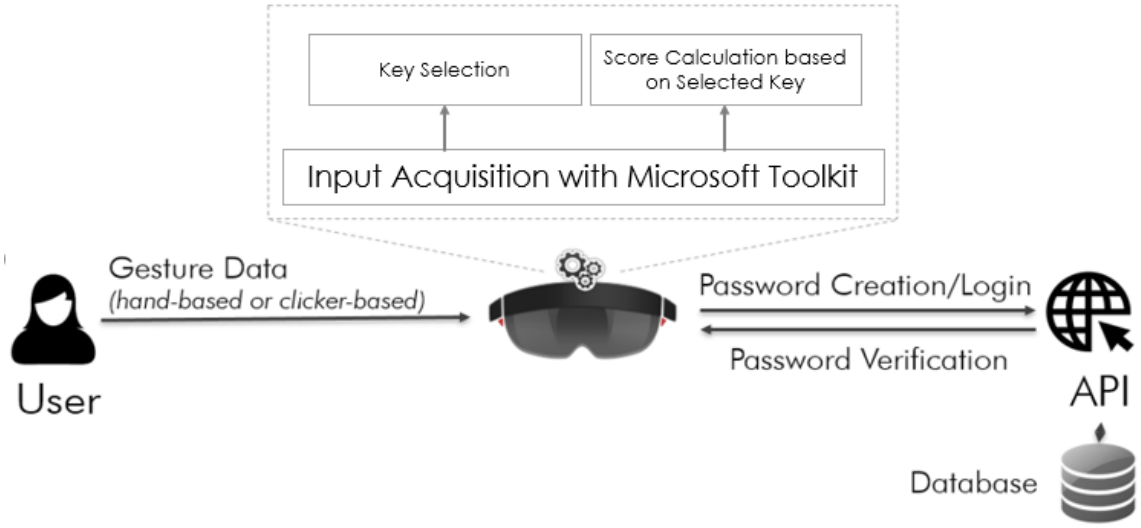

Figure 3.10 – Recognition-based Three-tier Architecture model

---

In our three-tier model adaptation, the first tier comprises of the interface the user sees in the application which will be demonstrated later. The second tier is built using C#. Unity3D components, Python and PHP as an API and finally the third tier is a MySQL database.

Chapter 4

Recall-based System – Design and Implementation

4.1 Introduction

This chapter will present the recall-based system that we developed, coined HoloPass, which is a HoloLens graphical user authentication system for registering an account and later logging in, in order to access certain functionalities and resources. Furthermore, the user interface and the functionality of the different modules that comprise the system will be discussed along with the architecture of the database as well as the process of implementing the gesture drawing functionality.

4.2 Database Architecture

Our database was structured and implemented using MySQL which is an open-source RDBMS (relational database management system) tha allows for development, administration, creation, maintenance and design of SQL databases. It is working on an Apache web server software and can be used for data storing, retrieval and management. The database consists of five tables. The primary table [Fig. 4.1] is responsible for holding the necessary data for a user's password. Every registered user has 3 records in this table, representing each one of their gestures. These data are: ***i) ID*** which is the record's ID in the database, ***ii) USERNAME*** which is the username each user typed as their own, ***iii) SELECTED_IMAGE*** which is the image the users selected from a pool of 6 images, ***iv) GESTURE*** which is the gestured the user performed (dot, line or circle), ***v) START_X*** which is the position on the X axis where the gesture was performed (this field was recorded only if a dot or line gesture was registered), ***vi) START_Y*** which is the position on the Y axis where the gesture was performed (this field was recorded only if a dot or line gesture was registered), ***vii) END_X*** which is the position on the X axis where the gesture was terminated (this field was recorded only if a line gesture was registered), ***viii) END_Y*** which is the position on the Y axis where the gesture was terminated (this field was recorded only if a line gesture was registered), **ix) CENTER_X** which is the position on the X axis where a circle was registered, ***x) CENTER_Y*** which is the position on the Y axis where a circle was registered and ***xi) RADIUS*** which is the radius of the registered circle gesture. The table ***register_fixation_data*** [Fig. 4.2] is responsible for holding the necessary data for every tracked fixation by the Pupil Labs' HoloLens plugin while the user is registering a password. These data are: ***i) ID*** which is the record's ID in the database, ***ii) REGISTER_DATA_ID*** which is the registered user's ID in the database, ***iii) GESTURE*** which is the gesture number (1,2 or 3) that he user was executing while the plugin was recording fixations, ***iv) SEGMENT_X*** which is the X axis of the segment the fixation belongs to, ***v) SEGMENT_Y*** which is the Y axis of the segment the fixation belongs to, ***vi) DURATION*** which is the duration of the registered fixation and ***vii) AOI*** which is the area of interest (1,2 or 3) the user's fixation was registered at. This table has

a M-1 relation with the ***register_data*** table between the ***REGISTER_DATA_ID*** and ID columns. The table ***register_data*** [Fig. 4.2] is responsible for holding the necessary data for every registered user. Such data are: ***i) ID*** which is the ID of the registered user, ***ii) USERNAME*** which is the username of the registered user, ***iii) TIME_TO_SELECT*** which is the time in milliseconds it took the user to select an image from the 6-image pool, ***iv) TIME_FOR_G1*** which is the time in milliseconds that it took the user to register their first gesture, ***v) TIME_FOR_G2*** which is the time in milliseconds that it took the user to register their second gesture, ***vi) TIME_FOR_G2*** which is the time in milliseconds that it took the user to register their third gesture, ***vii) TIME_TO_CONFIRM*** which is the time in milliseconds that it took the user to confirm their password by re-entering it, ***viii) RETRIES*** which is the total retries that it took the user in order to successfully register their password and ***viii) DATE*** which is the date and time of the registered password. The next two tables, ***login_fixation_data*** [Fig. 4.3] and ***login_data*** [Fig. 4.3] have the same structure as ***register_fixation_data*** and ***register_data*** respectively, with the only difference in the second table where there is no ***TIME_TO_CONFIRM*** column.

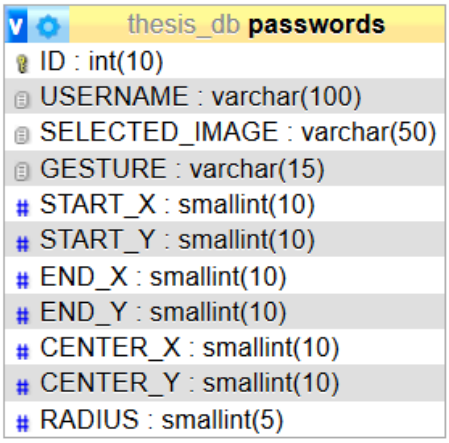

Figure 4.1 – passwords Table in the Database

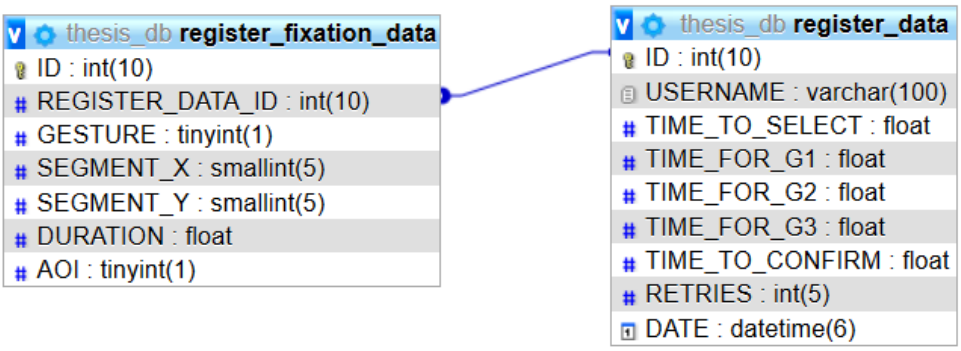

Figure 4.2 – register_fixation_data and register_data Tables in the Database

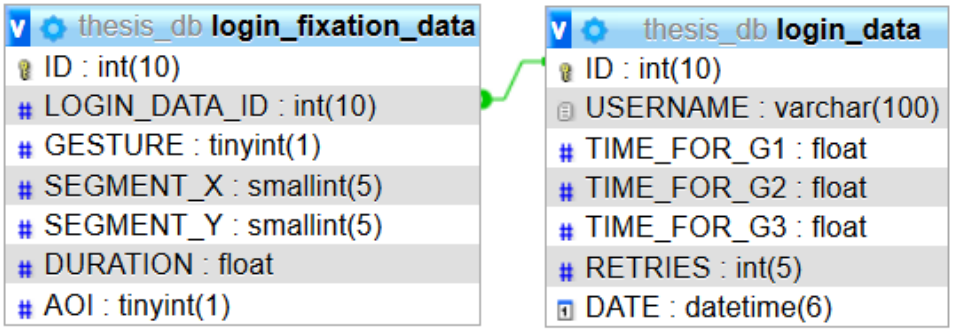

Figure 4.3 – login_fixation_data and login_data Tables in the Database

---

4.3 User Interface and Implementation

Before beginning implementation of the initial thoughts and design aspects of HoloPass, a thorough research and study of Microsoft's documentations regarding developing HoloLens applications as well as online forum FAQs had to be done. After completing the project setup so that we can later deploy it as a HoloLens application [Fig. 4.4], we begun constructing the user interface using the available HoloLens' Toolkit components based on Windows 10 Picture Password [Fig. 4.5].

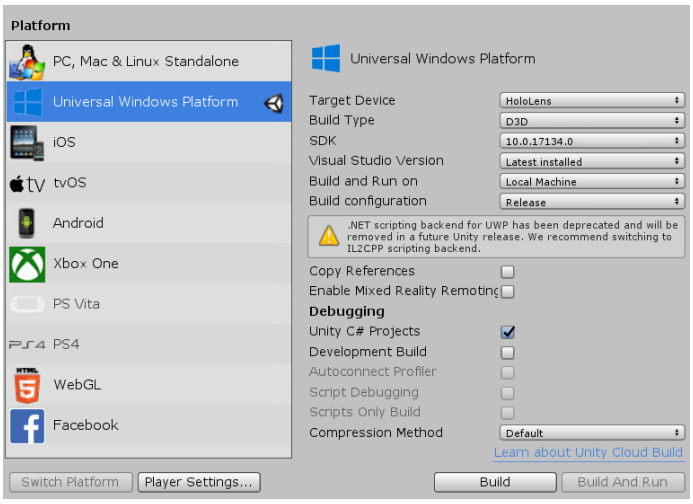

Figure 4.4 – Unity Build Settings for deploying on HoloLens

Figure 4.5 – Windows 10 Picture Password

4.3.1 Main Menu

The Main Menu [Fig. 4.6] is responsible for allowing the user to traverse through the application and choose which component of the application to use. The available options are Register, Login and Train. In each one of the three screens, there is a panel on the left-hand side available that contains instructions and navigation/control buttons and on the right-hand side there is a gameobject that contains a script which controls the currently

visible image which is selected by the user from a selection popup and the Manipulation Manager script which controls the manipulations made by the user in order to create the gestures, both of which will be explained later.

4.3.1.1 **User Interface**

The Main Menu's gameobject hierarchy [Fig. 4.7] consist of a main gameobject, with one child, and that child consists of 6 children. The main gameobject that holds all the child gameobjects, is a simple gameobject that serves as a container for the main scripts that control the functionality of the Main Menu buttons and the alignment of the menu based on the user's gaze, meaning that it will always remain at the user's head height. The child of the main gameobject is a dialog gameobject, available through the Toolkit, that helps by moving the main menu wherever the user's gaze is positioned so that it is always visible in front of the user. The BackPlate child gameobject is a simple container for the background material of the main menu. Similarly, the BackPlate (1) and BackPlate (2) are containers for the background material of the Register and Login buttons and the Train button respectively. The TitleText and TitleMessage gameobjects are TextMeshPro containers that are responsible for displaying the title and the message below the title of the main menu respectively. The ButtonParent gameobject, is a container that holds the three available buttons. The Register, Login and Train gameobjects are HolographicMeshButtons components (available through the Toolkit) that are ready-to-use, responsive holographic buttons. Each button is responsible for changing the view of the user to their specified panel.

4.3.1.2 **Implementation**

The core functionality of the main menu is to simply change the user's view to the selected panel. To achieve that, whenever a click of the clicker is detected, it checks to see if the user's gaze was on one of the three buttons based on their gameobject name and if it was, then it disables the visibility of the main menu and enables the visibility of the selected

panel. The way this works is, in the Menu script [Fig. 4.8] on the Menu gameobject, we define as interactable the buttons gameobjects which are the ones that are checked if they have been clicked, and we also define the Train, Register and Login panels that are the ones which will become visible as soon as one of the buttons is pressed. In the script, under the InputClicked function, we switch the possible gameobject names that may be clicked (the ones defined under the interactable list) and under each case, we disable the main menu gameobject and enable the according panel. This functionality is the same with all the buttons in the system, meaning, each button control script in each view of the application has the InputClicked function that switches the button that the user clicked.

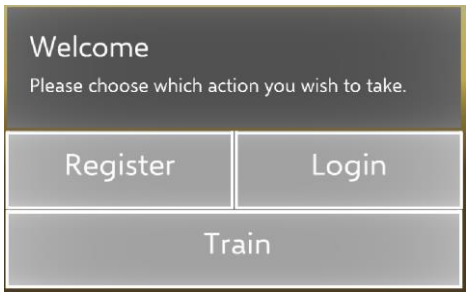

Figure 4.6 – Main Menu Interface

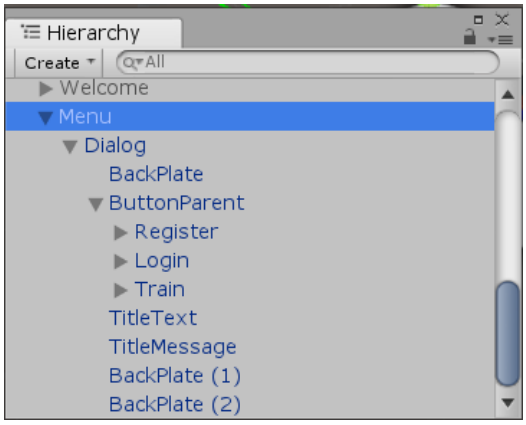

Figure 4.7 – Main Menu Hierarchy

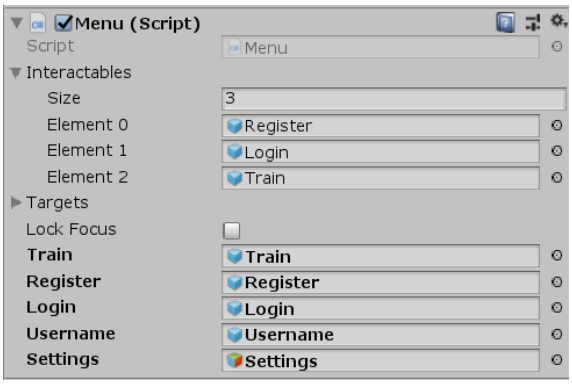

Figure 4.8 – Menu Script layout

4.3.2 Train

The Train screen is responsible for allowing the user to train using the available gestures and basic interaction methods with the system as long as they desire in order to get accustomed with the system and feel comfortable enough so that they can create their password with ease.

4.3.2.1 **User Interface**

The first main screen that will be explained is Training [Fig. 4.9] because it would be the one we also used for testing the mechanics of the password creation on. The gray panel on the left-hand side, is a simple RectTransform that has a Sprite Renderer component has a plain white sprite as the sprite attribute and the ImagesButtonBack material as a component material, that comes with the Toolkit in order to get the gray background. Furthermore, as a child component of the panel, there is a gameobject called Instructions which hold two more gameobjects called InfoTitle and InfoContent [Fig. 4.10]. The two gameobjects contain a TextMeshPro component each. The InfoTitle TextMeshPro text contains the title which is visible on the top-left side of the panel and the InfoContent TextMeshPro text contains the description text which is visible below the title of the panel.

Also, the Back button is available, which is a HolographicMeshButtons (available through the Toolkit) and it is child of a parent gameobject called ControlBtns that helps by rendering the button in view and by making it available for clicking. The image on the right, is a gameobject available to all screens that contains

4.3.2.2 **Implementation**

The only functionality available in the Train screen, is clicking the Back button. It is responsible for closing the train panel and displaying the main menu which was explained in the section 4.4.1 with the only difference being the script that controls the button functionality. In this case, the script is called Train Image Panel [Fig. 4.11] and it resides in the TrainImagePanel gameobject and it has as interactable only the Back button and the Menu and Train gameobjects for disabling and enabling them from view if the user clicks the back button.

Figure 4.9 – HoloPass Training Screen

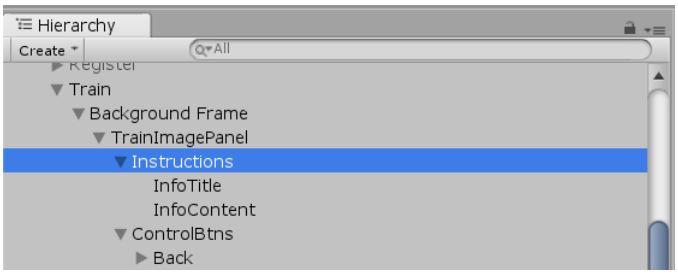

Figure 4.10 – Train Panel Hierarchy

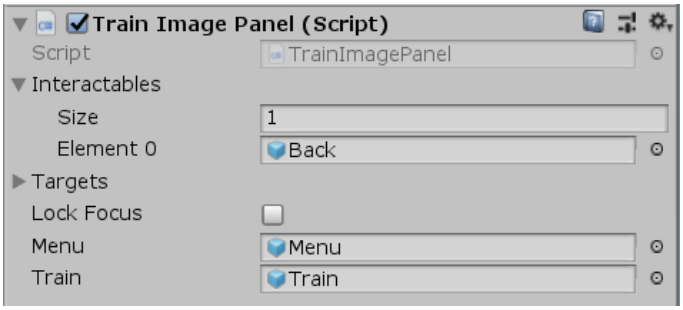

Figure 4.11 – Train Image Panel Script layout

4.3.3 Register

The Register screen is responsible for allowing the user to register a new account by inserting any combination of the three available gestures in a sequence of their desire on the selected image. The gesture drawing process will be explained later.

4.3.3.1 User Interface

(i) ***Username Popup:*** Once the user selects the Register option from the Main Menu, a Username popup [Fig. 4.12] is shown where it requests from the user to enter the username of their desire as the first step of creating an account. Once the user enters their password, they have to click the OK button in order to proceed to the next step of the registration process.

(ii) ***BrowseImagePanel:*** Initially, when the user enters the register panel, the first screen they see is the BrowseImagePanel [Fig. 4.13, Fig. 4.14]. The BrowseImagePanel gameobject is separated in two child objects, one responsible for displaying the panel title and the instructions below it and one responsible for rendering the buttons on the bottom side of the left panel in view and control the interaction of the user with them. On the right-hand side, the only component available is a placeholder image which instructs the user to select an image from the available ones in order to set it as their password background.

The user has three button options. The first one, Browse, opens a file browser [Fig. 4.15] which is a dialog just like the Main Menu that contains the images that the user can select one from to set as their background and two buttons, one for confirming the selection and another one for cancelling and closing the file browser. The second one, Continue, opens the registration panel which will be explained later. Finally, the last button, Menu, closes the browse panel and opens the Main Menu.

(iii) ***RegisterInfoPanel:*** When the user selects an image and clicks Continue, the RegisterInfoPanel panel [Fig. 4.16] becomes active. The RegisterInfoPanel gameobject is separated in three child objects. The first one, is responsible for the title and descriptive text that are visible on the left-hand side of the screen. The second one, is responsible for the numbers below the descriptive text which show the progress of the user regarding how many gestures they have entered so far. Finally, the last gameobject, is responsible for rendering the buttons on the bottom side of the left panel in view and control the interaction of the user with them. On the right-hand side, the user's selected image from the file browser is visible.

The user has two button options. One for starting over their password creation process if they have made any mistake during the registration process and the second one is for when the user has entered their password in order to proceed to

the next step of the registration process which is confirming the password and when clicked it displays the ConfirmInfoPanel.

(iv) ***ConfirmInfoPanel:*** Finally, the last gameobject available in the Register screen, is the ConfirmInfoPanel [Fig. 4.17]. This gameobject, is split into 3 child objects. The first one is responsible for the title and descriptive text that are visible on the left-hand side of the screen. The second one, is responsible for the numbers below the descriptive text which show the progress of the user regarding how many gestures they have entered so far in the confirmation process. Finally, the last gameobject is responsible for rendering the buttons on the bottom side of the left panel in view and control the interaction of the user with them. On the right-hand side, the user's selected image from the file browser is visible.

The user has three button options. The Retry button starts over the confirmation process, the Start Over button starts over the whole registration process and the Done button finishes the confirmation process.

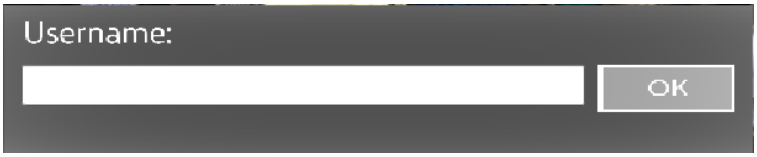

Figure 4.12 – Username Insertion Popup

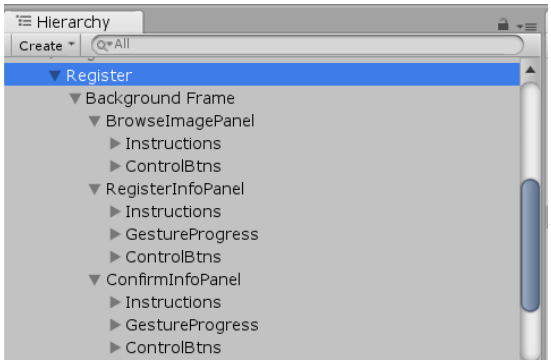

Figure 4.13 – Register Panel Hierarchy

Figure 4.14 – Browse Image Screen

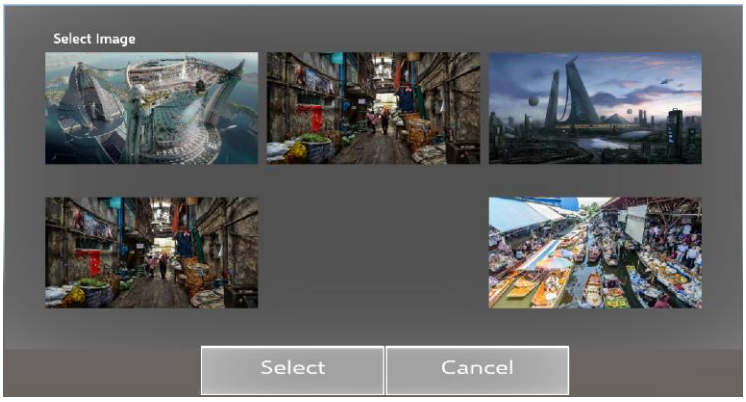

Figure 4.15 – File Browser

Figure 4.16 – Register Info Panel

Figure 4.17 – Confirm Info Panel

4.3.3.2 **Implementation**

(i) ***Username Popup:*** During the username insertion process, when the user is requested to enter a username, they have to gaze and tap on the input box in the Username Popup. When the user taps the input box, the Keyboard script on the canvas under the username gameobject is activated which opens a virtual keyboard supplied by the Toolkit and is bound to the Username Popup input field so that whatever the user types it is automatically displayed in the input field. When the user is done, they have to click the OK button in order to proceed.

(ii) ***BrowseImagePanel:*** When the user clicks the Browse button in the BrowseImagePanel screen and the file browser popup is shown, they are presented with options of images. When the user taps on an image, then, through script, the string name of the image is passed through the File Browser script to the BackgroundImg script which is responsible for displaying the selected image in the image panel. Once the Select button is pressed, then the Image script replaces the placeholder image with the selected image.

(iii) ***RegisterInfoPanel:*** During the registration process, whenever the user enters a gesture, the number regarding the currently entered gesture is highlighted to keep the user informed on their progress. This is managed by checking each time the

total inserted gestures through code, and in each occasion replace the Alpha channel of the color of the appropriate number with the value 255 in order to display the number with white color. Once the user is done with their gesture insertion, they press the Continue button and the gestures are saved in a list for checking them later with the confirmation step gestured to see if the passwords match.

(iv) ***ConfirmInfoPanel:*** During the confirmation process, the user has to reenter their gestures with the same sequence as in the registration process and the response from the system is the same as with the registration process. The number which is respective to the inserted gesture is highlighted. When the user is done with the confirmation and tap Done, the confirmation gestures are checked one-by-one with the registration gestures with a small offset being applied on the password since it is almost impossible to hit the exact same position each time for each gesture. If they are the same, then the list of gestures is serialized into JSON and they are sent to the endpoint which is then responsible for saving the gestures in the database with the correct format and the user is prompted back to the Main Menu. If the confirmation process is unsuccessful, an error message is displayed and the system requests from the user to re-confirm their password.

4.3.4 Login

The Login screen is responsible for allowing the user to login based on a pre-registered account by firstly inserting their password and then their three gestures in the same sequence on the image they selected during the registration process, which is automatically presented for them since it is saved with their password in the database. This gameobject [Fig. 4.18] is separated in two child objects, both of which will be explained in the next sections.

4.3.4.1 **User Interface**

(i) ***Username Popup:*** Once the user selects the Login option from the Main Menu, a Username popup [Fig. 4.12] is shown where it requests from the user to enter the username of their desire as the first step of creating an account. Once the user enters their password, they must click the OK button in order to proceed to the next step of the login process. If the entered username does not a exist in the database, an error message is displayed and the system requests from the user to re-enter their username. If not, the system proceeds to the LoginInfoPanel [Fig. 4.19].

(ii) ***LoginInfoPanel:*** During this process, the user is requested to enter their password that they had registered during the registration process. On the left-hand side panel, the user can see a title and descriptive text of the task at hand below the text a button for restarting the login process if they have made any mistakes during the gesture insertion.

(iii) ***LoginResut:*** When the user enters the wrong password, then this gameobject is activated which replaces the LoginInfoPanel left-hand side panel with one [Fig.4.20] that displays an error message and allows the user to restart the login process. Although, if the user enters the correct password then a welcome message is displayed, and the user is granted access to the requested service or resource [Fig. 4.21].

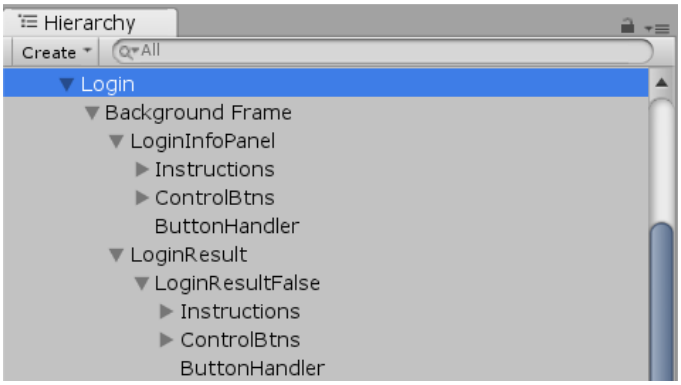

Figure 4.18 – Login Panel Hierarchy

Figure 4.19 – Login Info Panel

Figure 4.20 – Login Result

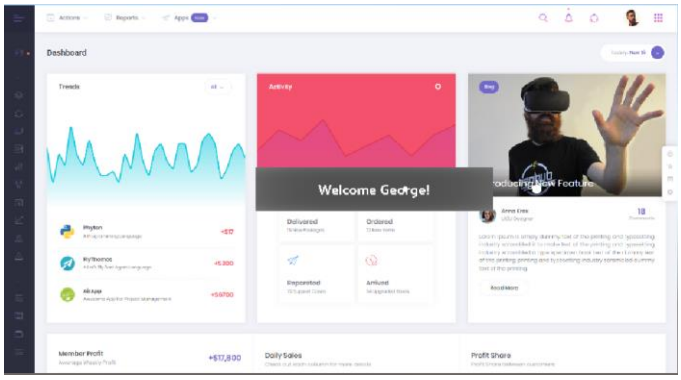

Figure 4.21 – Successfully logged in user

4.3.4.2 **Implementation**

(i) ***Username Popup:*** During the username insertion process, when the user is requested to enter a username, they have to gaze and tap on the input box in the Username Popup. When the user taps the input box, the Keyboard script on the

canvas under the username gameobject is activated which opens a virtual keyboard supplied by the Toolkit and is bound to the Username Popup input field so that whatever the user types it is automatically displayed in the input field. When the user is done, they have to click the OK button in order to proceed.

(ii) ***LoginInfoPanel:*** During the login process, the user has to re-enter their gestures with the same sequence as when they registered their password. When they enter three gestures, they are serialized into a JSON object and they are sent to the endpoint which retrieves their password based on their username and checks the two passwords if they match with a small offset being applied to the entered password since it is almost impossible to hit the exact same position each time for each gesture.

(iii) ***LoginResult:*** If the response from the endpoint after the check is False, then the login attempt was incorrect and the gameobject responsible for displaying the error message is activated. If the response from the endpoint after the check is True, then the login attempt was correct and all the gameobjects are disabled and the welcome message is activated to welcome the user.

4.3.5 Database Communication

This subsection will present the way passwords are saved in the database, how the background image of each user is retrieved and how the passwords are checked when logging in.

4.3.5.1 **Saving Passwords in the Database**

When a password is registered, all the gestures are serialized in JSON format and sent to the endpoint in order to be saved in the database. For each gesture, the system saves different attributes so that each gesture can be easily checked later when a login attempt is made. For the Dot gesture, the system saves only the X and Y positions on the image where it was registered. For the Line gesture, the system saves only the start point X and Y and

the end point X and Y positions on the image where it was registered. Finally, for the Circle gesture, the system saves only the X and Y positions of the circle's center and the radius of the circle which is calculated by the difference of the position of a point on the circumference of the circle from the center of the circle. Also, along with the gestures, the system saves the user's username which is entered before registering the password and the name of the selected image during the browse image phase of the password. These are used for retrieving the password later on when the user will login.

4.3.5.2 Background Image Retrieval

When a user attempts to login, when they enter their username, the username is passed to the endpoint which retrieves from the database the pre-selected image that the user had set as their background during login. Then, from the available background images, the system finds and displays the registered background image.

4.3.5.3 Password Check during Login

When the user attempts to login into the system, the entered gestures are sent to the endpoint along with the username and the endpoint retrieves the registered password from the database based on the given username. After the retrieval, the endpoint does a one-to-one check if the gestures match with a small offset on their positions since it is impossible to have perfect position match of all gestures. If all the gestures match, the endpoint returns a True flag to the system which states that the login attempt was successful. If at least one of the gestures does not match, then the check process is immediately interrupted, and a False flag is sent to the endpoint which states that the login attempt was unsuccessful.

4.3.6 Drawing Gestures

The most important mechanic we had to develop was the gesture drawing. Initially, we thought of allowing the user, by utilizing a tap-and-hold gesture, to draw on the image by moving their hand. So, if the user wanted to draw a circle, they would have to tap-and-

hold and move their hand in a circular movement. For the line, they would have to tap-and-hold and move their hand in a straight line. For the dot, the mechanic is simple, they would simply have to tap on the position of their desire. This was achieved by using Unity's and the Toolkit's Manipulation events that allow tracking of the user's hands and return different properties, for example the user's hand position in the world. The way this gesture drawing mechanic worked is, when the user initializes a tap-and-hold gesture for drawing a line or a circle the system would save the positions of the user's hand while it moved in a list. As the positions were being saved, they would also be drawn on a bitmap image which was transformed into a sprite image [Fig. 4.22, Fig. 4.23] and displayed to the user to give them the ability to preview what they were drawing.

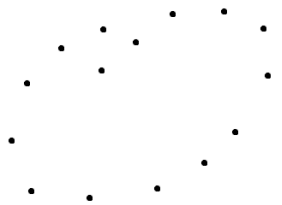

Figure 4.22 – Circle Gesture Drawing

Figure 4.23 – Line Gesture Drawing

After creating the drawing mechanic, we needed to find a way to know which one of the two gestures the user was drawing. For this, we utilized Microsoft's CustomVisionAI,

which is a vision classification API which classifies images that you provide through REST API calls into specific categories that you specify while training the vision model. A training sample of 20 images per gesture were provided for training this model. Once the user stopped the tap-and-hold gesture, the system would convert the bitmap image into a byte stream and send it to the CustomVisionAI endpoint and it would, in return, answer with the label of the provided image. If the image was a circle then it would return a JSON file containing the record “label:Circle” and if it was a line then it would return a JSON file containing the record “label:Line”. This way we would know how to handle the positions we had in the list of hand positions. If the gesture was a circle, then its center was calculated using trigonometry and the radius of the circle by finding its equation after finding the center. If the gesture was a line, then only the first and last position were used since the line would start and end on those two positions and no further information was needed. This mechanic idea was deemed malfunctioning since the CustomVisionAI would not always make the correct classification and the calculation algorithms for the center and radius of the circle would sometimes not work properly and we would get miscalculations. Also, the tap-and-hold mechanic and the way the drawn gesture was displayed to the user, was not user friendly and it was difficult as well as tiring to manage to achieve the exact position of the drawing you want. Moreover, using hand gestures made the input very sensitive to movement, so the slightest hand movement had great impact on the way the gesture was recorder. In order to tackle the last-mentioned problem, we decided to implement HoloLens’ Clicker which is a Bluetooth remote controlled clicker that allows for tap and tap-and-hold gestures but without having feedback about the user’s hand position, only about their hand movement.

After deciding to use the clicker, we had to rethink the drawing mechanic since it was buggy and also could no longer operate correctly since the clicker does not provide position feedback. As a result, we came up with a simpler to use and to develop mechanic. The functionality’s core would be the user’s gazing position. Each gesture would start from the gazing position and expand from there. In case the user wants to create a dot, the user would simply click the clicker while gazing at the position of their desire [Fig. 4.24].

Then, the X and Y positions of the dot on the background image would be recorded and saved. In case the user wants to create a line, they would have to gaze at the position where they want the line to start from and then tap-and-hold and move the clicker towards any direction they want the line to be expanded towards. The clicker's movement magnitude would be transformed into world movement and applied to the initial gazing position. That way, we were able to display, in real time, the creation of the line for the user to see and decide how long it would be on the image [Fig. 4.25]. When the gesture drawing is done, the X and Y positions of both the start and end position of the line are recorded and saved. In case the user wants to create a circle gesture, they would have to double-tap-and-hold on the position of their desire which will serve as the circle's center. From there on, the movement is the same as the line gesture, but the difference is that instead of controlling the length of the line, they control the size of the radius. Using a similar implementation as the line length control, we were able to draw a circle and, in-real-time, make it bigger or smaller according to the radius that the user was setting [Fig. 4.26]. When the gesture drawing was done, the X and Y positions of the circle's center, as well as the radius were recorded and saved. This approach was simpler to use and made the gesture drawing feature easy and after very little training a user could freely create any shape on any position they wanted.

Figure 4.24 – Dot Gesture

Figure 4.25 – Line Gesture

Figure 4.26 – Circle Gesture

Chapter 5

Recognition-based System – Design and Implementation

5.1 Introduction

This chapter will present the recognition-based system that we developed, which is a HoloLens graphical user authentication system for registering an account and later logging in, in order to access certain functionalities and resources. It is a replica of current state-of-the-art graphical user authentication systems that exist on desktop environments. Furthermore, the user interface and the functionality of the different modules that comprise the system will be discussed along with the architecture of the database as well as the process of implementing the object selection functionality. The only difference between this system and the previous one, in terms of technologies used, is that the recognition-based one does not use the clicker and Pupil Labs' hardware.

5.2 Database Architecture

Our database was structured and implemented using the same framework that was used in the Recall-based system which was discussed in Chapter 4. The table ***passwords*** [Fig. 5.1] is responsible for holding the necessary data for a user's password. Every registered user has one record in this table. The data in each record are: ***i) ID*** which is the record's ID in the database, ***ii) USERNAME*** which is the automatically generated username that we set up the system to generate for each user for ease of use, ***iii) KEY1*** which is the ID of the first key the user selected, ***iv) KEY2*** which is the ID of the second key the user selected, ***v) KEY3*** which is the ID of the third key the user selected, ***vi) KEY4*** which is the ID of the fourth key the user selected, ***vii) KEY5*** which is the ID of the fifth key the user selected. The table ***register_data*** [Fig. 5.2] is responsible for holding the necessary data for every registered user. Such data are: ***i) ID*** which is the ID of the registered user, ***ii) USERNAME*** which is the username of the registered user, ***iii) TIME_FOR_IMG1*** which is the time in milliseconds that it took the user to select the first item, ***iv) TIME_FOR_IMG2*** which is the time in milliseconds that it took the user to select the second item, ***v) TIME_FOR_IMG3*** which is the time in milliseconds that it took the user to select the third item, ***vi) TIME_FOR_IMG4*** which is the time in milliseconds that it took the user to select the fourth item, ***vii) TIME_FOR_IMG5*** which is the time in milliseconds that it took the user to select the fifth item, ***viii) TIME_TO_CONFIRM*** which is the time in milliseconds that it took the user to confirm their password by re-entering it, ***ix) RETRIES*** which is the total retries that it took the user in order to successfully register their password, ***x) RESETS*** which is the amount of times the user tapped the Reset button in order to re-enter their password and ***xi) DATE*** which is the date and time of the registered password. Finally, the table ***login_data*** [Fig. 5.3] has the same structure as ***register_data*** and, with the only difference being that there is no ***TIME_TO_CONFIRM*** column in ***login_***data.

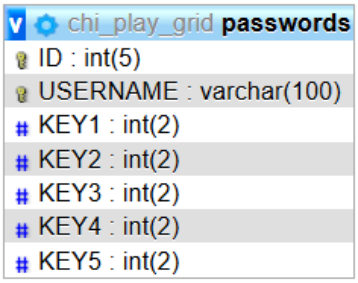

Figure 5.1 – passwords Table in the Database

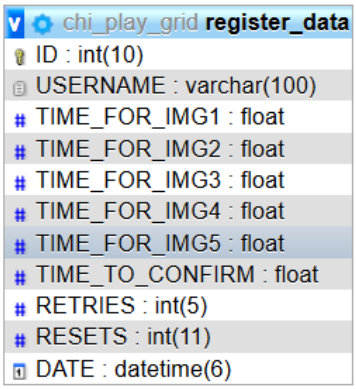

Figure 5.2 – register_data Tables in the Database

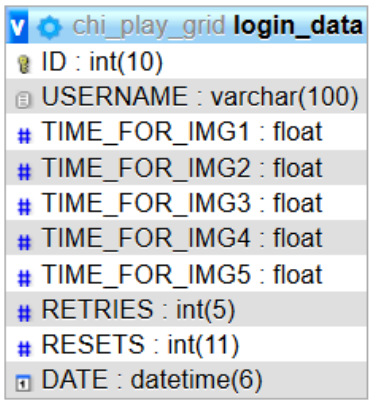

Figure 5.3 – login_data Tables in the Database

5.3 User Interface and Implementation

For implementing the HoloLens version of the state-of-the-art authentication system, we followed the guidelines of other authentication mechanisms [5, 7, 15], which specify implementation guidelines and restrictions. Moreover, the length of the password that the users would be allowed to create, is based on already-existing works [3, 4, 13]. For simplicity terms, we did not implement username insertion. Instead, a username is automatically generated and assigned to each user. We decided this implementation since it would be much slower, tedious and hard for the users to enter their username due to the difficulties in the keyboard usage we found out in the HoloPass development and evaluation.

5.3.1 Main Menu

The Main Menu [Fig. 5.4] is responsible for allowing the user to traverse through the application and choose which component of the application to use. The available options are Register, Login and Train. The functionality is similar to the recall-based system. The user taps an option and the respective screen to that option appears.

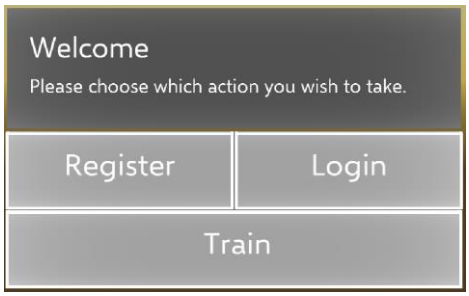

Figure 5.4 – Main Menu Interface

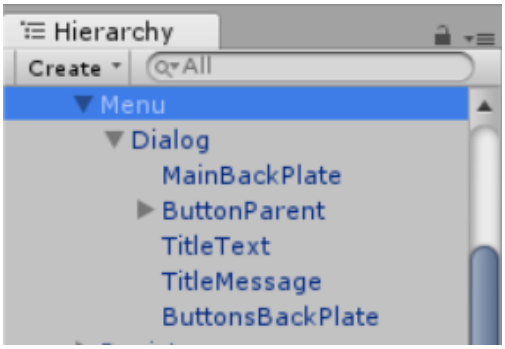

Figure 5.5 – Main Menu Hierarchy

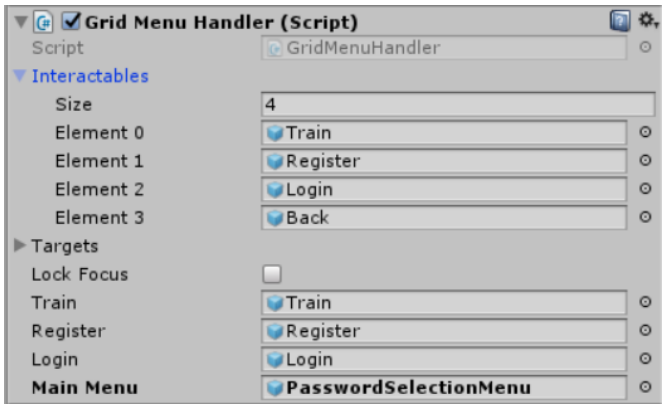

Figure 5.6 – Grid Menu Handler Script layout

5.3.1.1 **User Interface**

The Main Menu's gameobject hierarchy [Fig. 5.5] consist of a main gameobject, with one child which consists of 5 children. The main gameobject that holds all the child gameobjects, is a simple gameobject that serves as a container for the main script that control the functionality of the Main Menu buttons and the alignment of the menu based on the user's gaze, meaning that it will always remain at the user's head height. The child of the main gameobject is a dialog gameobject, available through the Toolkit, that helps by

moving the main menu wherever the user's gaze is positioned so that it is always visible in front of the user. The MainBackPlate child gameobject is a simple container for the background material of the main menu. Similarly, the ButtonsBackPlate is a container for the background material of the Register, Login and Train buttons. The TitleText and TitleMessage gameobjects are TextMeshPro containers that are responsible for displaying the title and the message below the title of the main menu respectively. The ButtonParent gameobject, is a container that holds the three available buttons. The Register, Login and Train gameobjects are HolographicMeshButtons components (available through the Toolkit) that are ready-to-use, responsive holographic buttons. Each button is responsible for changing the view of the user to their specified panel.

5.3.1.2 **Implementation**

The core functionality of the main menu is to simply change the user's view to the selected panel. To achieve that, whenever a click of the clicker is detected, it checks to see if the user's gaze was on one of the three buttons based on their gameobject name and if it was, then it disables the visibility of the main menu and enables the visibility of the selected panel. The way this works is, in the Menu script [Fig. 5.6] on the Menu gameobject, we define as interactable the buttons gameobjects which are the ones that are checked if they have been clicked, and we also define the Train, Register and Login panels that are the ones which will become visible as soon as one of the buttons is pressed. In the script, under the InputClicked function, we switch the possible gameobject names that may be clicked (the ones defined under the interactable list) and under each case, we disable the main menu gameobject and enable the according panel. This functionality is the same with all the buttons in the system, meaning, each button control script in each view of the application has the InputClicked function that switches the button that the user clicked.

5.3.2 Train

The Train screen is responsible for allowing the user to train using the available interaction methods with the system as long as they desire in order to get accustomed with the system, get used to the way the system responds back to their input and feel comfortable enough so that they can create their password with ease.

5.3.2.1 **User Interface**

The first main screen that will be explained is Training [Fig. 5.7] because it would be the one we also used for testing the mechanics of the password creation on. The screen consists of 25 buttons, each one representing one image. When a user taps on a button, it is automatically marked as selected to inform the user of their selection [Fig. 5.8]. The buttons are aligned in a 5x5 grid for good presentation. Also, a button below the grid is available with the title Menu.

5.3.2.2 **Implementation**

As mentioned before, the training screen has been implemented in order to help the users get accustomed with the functionalities of the system. The basic functionality which is universal to all three screens is the selection of keys. When a user gazes and taps on an item in the world, the TrainImageGridHandler script [Fig. 5.10] is checking the list of the Interactable components that have been referenced and if the clicked item's name is present then it shows the "Selected" text on the button. The second available functionality in this screen is the Menu button which disables the current view, resets all the selected keys to their unselected position and displays the Main Menu.

Figure 5.7 –Training Screen

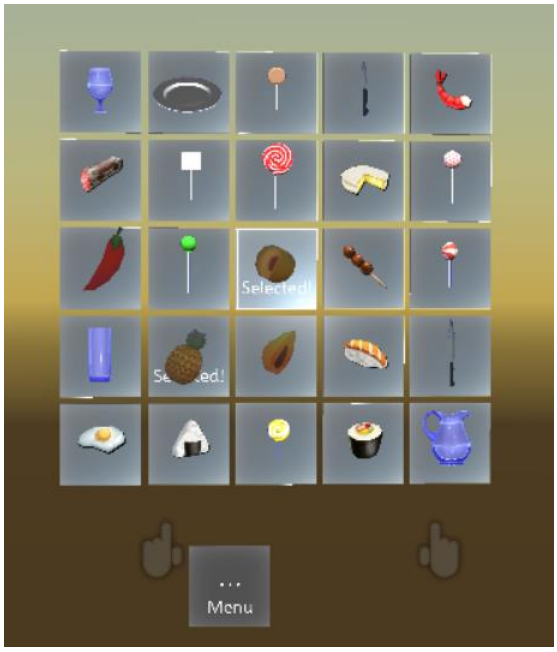

Figure 5.8 –Training Screen After Interaction

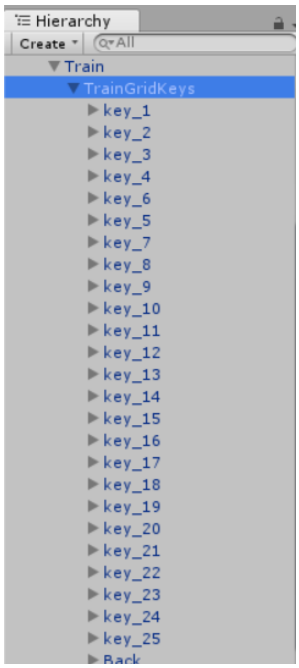

Figure 5.9 – Train Screen Hierarchy

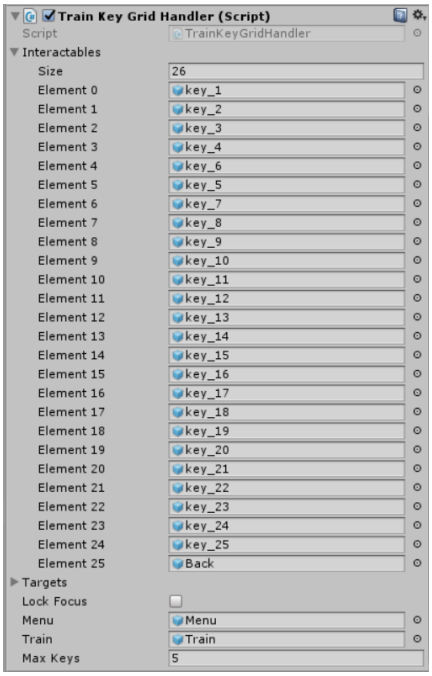

Figure 5.10 – Train Image Grid Handler Script layout

5.3.3 Register

The Register screen is responsible for allowing the user to register a new account by inserting any combination of five images, that they want from the available set, in a sequence of their desire on the selected image.

5.3.3.1 **User Interface**

When a user enters the registration screen [Fig 5.11], they are presented with 91 images of items that can be selected as their password. The allowed selection is five keys. When a user selects a key, it is marked as Selected and also presented above the grid next to the Reset button [Fig. 5.12]. This is so that at any moment during the password creation process the user can view their selections. Moreover, apart from the 91 keys, there are also available a Reset button, which resets the password, a Menu button, which disables the current screen and displays the Main Menu and finally, once the user has selected 5 keys, a continue button appears which allows for the password confirmation step where the user must simply re-enter their password in the same sequence that it was registered.

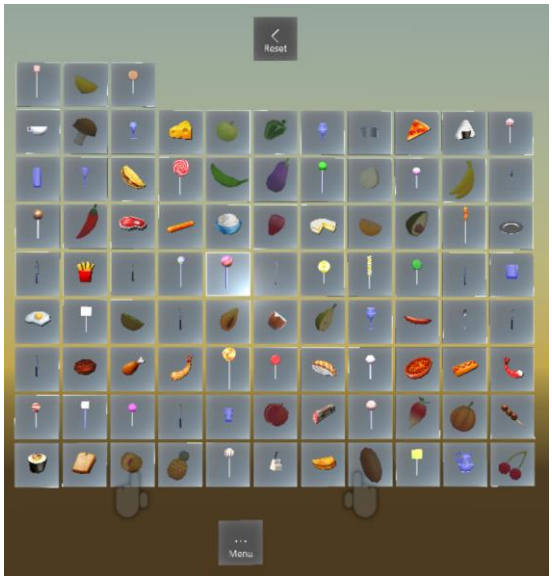

Figure 5.11 – Registration Screen

Figure 5.12 – Registration Screen with Selected Key

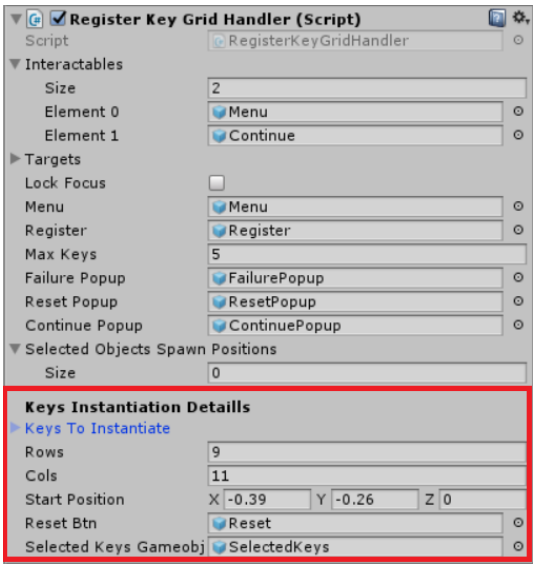

Figure 5.13 – Button Spawner component within Register Key Grid Handler Script

5.3.3.2 **Implementation**

(i) ***Register Step:*** For implementing this screen, a button spawner [Fig. 5.13] had to be implemented that will receive all the keys references in a list and it would generate a grid with said keys according to the rows and columns and starting position specified by the developer. Each key is then displayed on the screen below the reset button and above the menu button. When the user taps on a button, the RegisterKeyGridHandler script checks the name of the clicked key in the Interactables list and if it is found, according actions are taken. If a key has been pressed, then it is displayed above the grid for the user to see at all times as mentioned before and also, it is marked as Selected. If a user taps the Reset button, then all the key buttons are set to their original unselected state. When a user makes a selection, a script is checking the selected items list and if the length of the list is 5, meaning that the limit of selected keys has been reached, the Continue button is displayed and when it is pressed the password confirmation process begins.

(ii) ***Confirmation Step:*** Once the confirmation step has been activated, the available processes are the same as in the registration step. The only difference is that when the user re-enters their password and it is correct, then they are redirected to the Main Menu of the application. If the confirmation is unsuccessful then the user must re-confirm their password. At any time of the confirmation step they can Reset their password and the selected keys are all displayed above the grid as before.

5.3.4 Login

The Login screen is responsible for allowing the user to login based on a pre-registered account inserting their password in the same sequence they entered it during the registration process.

5.3.4.1 **User Interface**

The gameobject layout is the same as in the Registration screen. There is a grid which is automatically generated, but instead of having 91 keys there are only 25 [Fig. 5.14]. Also, there is a Menu button below the grid and a Reset button below the grid. Last but not least, all the image selections that the user makes, are marked as Selected and displayed above the grid next to the Reset button [Fig. 5.15].

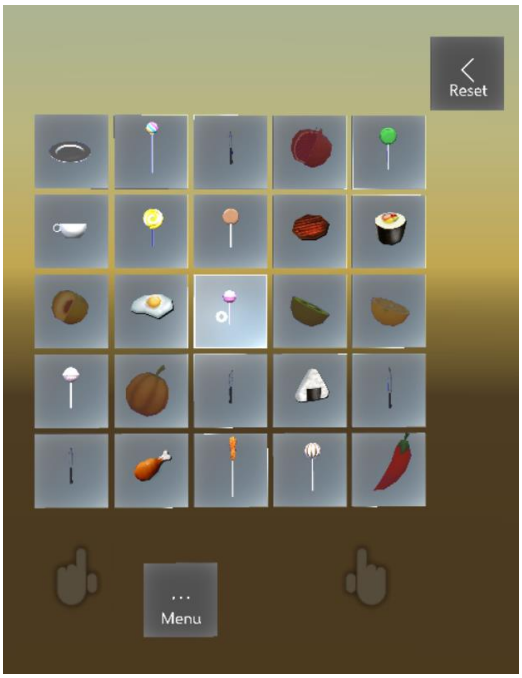

Figure 5.14 – Login Screen

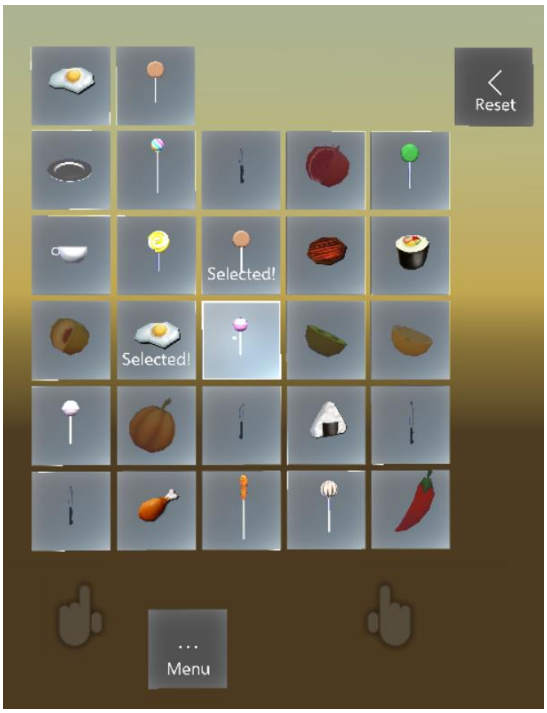

Figure 5.15 – Login Screen after two key selections

5.3.4.2 **Implementation**

The functionalities of this screen are almost the same as in the registration screen. A user is presented with 25 images, within which their 5-image password is included. The other 20 remaining images are all randomly selected from the image pool that was present in the registration screen. When a user enters their 5-image password in the same sequence as in the registration process, then the system redirects them to the Main Menu. If at any point the user has made any wrong selection, they can tap the Reset button and they will have to insert their password again. The button recognition and tapping operates in the same way as in the training and registration screens.

5.3.5 Database Communication

This subsection will present the way passwords are saved in the database and how the passwords are checked when logging in.

5.3.5.1 Saving Passwords in the Database

When a user completes the password registration process, the ids of the 5 keys are sent to the API in order to be saved in the database. A query is initially executed which counts the already inserted passwords in the database and increments that value by one. That number is the username of the registered user and it will be saved along with their keys. Once the username has been generated, an insertion query is executed that inserts in the database the keys, the username, the date and the total resets and retries the user did until their successful registration.

5.3.5.1 Password Check during Login

Once the user has entered their password during the password login process, the selected 5 keys are sent to the API in order to be checked for validity. The API, firstly, executes a query with which it retrieves the password of the last saved record in the database. That happens due to our implementation decision to not include username insertion, so every user has to complete their registration step and then their login step before continuing with the next user. One the 5-key stored password has been retrieved, each key is checked one-by-one with the entered ones by the user. If they match, then a true response is returned to the system, otherwise a false response is returned.

Chapter 6

Evaluation

6.1 Introduction

This chapter will discuss about how we evaluated the two developed systems for collecting data in order to study if PGAs are a good alternative and viable authentication solution for MR contexts. To do so, thirty people participated in the user study for the first system and thirty people for the second one. The goal for both systems was to present the users with a scenario where they would have to interact with the systems in a sequence of steps in order to complete an account registration process. Finally, they had to answer a questionnaire for each system in order to provide feedback in terms of usability, likability, perceived security and functionality of the systems. Each user, before participating in the user study, had to complete a consent form, where they stated that we were allowed to

monitor their behavior and interaction and also record their eye-gaze (for the first experiment).

6.2 Recall-based System

6.2.1 Evaluation Scenario

Before evaluating the Recall-based authentication system, we firstly had to think of a real-life scenario that we would present to the users, for them to interact with. First of all, we explained the purpose of this system's development and how it can be used to, hopefully, improve the authentication process in Mixed Reality environments. Furthermore, the main modules that comprise the system were explained and the way to interact with them, and most importantly the way to draw the gestures onto the background image. After this brief explanation of the system, we proceeded with presenting the users with a scenario that represents an every-day occasion where someone may want to use HoloPass instead of the conventional text-based authentication method. The first interaction module they had to complete was a training module. The tasks for this module were:

(i) First of all, they had to traverse through the menu in order to enter the training screen
(ii) Then, they had to complete a dot gesture
(iii) Then, they had to complete a line gesture
(iv) Then, they had to complete a circle gesture
(v) Lastly, if they wanted, they could keep drawing gestures for as long as they liked in order to get as much accustomed with the drawing process as possible
(vi) Once they were done, they had to return to the main menu screen

After completing the training session, each user proceeded with completing the registration process of the system. The tasks for this session were:

(i) Enter the registration screen using the according option in the main menu

(ii) Type in a username of their selection
(iii) Browse for a background image (the same image was available for all users, so we don't have any biased data)
(iv) Enter a three-gesture password using any combination of the available gestures
(v) Proceed to the confirmation step of the password
(vi) Confirm the password by re-entering their three gestures at the same locations as they entered them while registering the password

Finally, after the registration step, it was required for the users to proceed to the login step where they would use their password to enter a dummy application. The tasks were:

(i) Enter the login screen using the according option in the main menu
(ii) Type in the username that they selected while registering an account
(iii) Enter their three-gesture password combination on the background cue-image (which is the same as the one they selected while registering)

6.2.2 Evaluation Process

The objective of carrying out this experiment was to evaluate if this new approach, in terms of authenticating users in Mixed Reality Head Mounted Displays, is more preferable by the users than the current authentication module, without affecting the security performances of the passwords. In order to do so, we decided to acquire quantitative, as well as qualitative data. For receiving the quantitative data, we implemented timers and other measurements for each task the user was executing. In detail, we measured:

(i) Time to select their background image from the image set
(ii) Time to draw the first gesture
(iii) Time to draw the second gesture
(iv) Time to draw the third gesture

(v) How long it took for each user to complete the registration task (both registering and confirming the password together)

(vi) How long it took for each user to complete the login task

(vii) How many retries it took the user to complete the registration task

(viii) How many retries it took the user to complete the login task

Moreover, each user visited an empty room in the University on a previously agreed date and time. At the end of each session, we asked users to complete a questionnaire [Fig. 6.1] with which we could see their preference in terms of difficulty in usability and preference of authentication system and also acquire our qualitative data.

The above process was the same for both the users that used the desktop control version and the users that used the HoloLens version. The only difference was in the questionnaire for the desktop control version did not include the questions that were relevant to the HoloLens version.

1. Was this your first time using the HoloLens?
2. What did you like about the application?
3. What didn't you like about the application?
4. What other electronic devices do you use in your everyday life?
5. How difficult did you find the gesture creation (1 – very difficult, 5 – very easy)?
6. How tiring did you find the gesture creation using the clicker (1 – very tiring, 5 – not tiring)?
7. Did you find any one of the gestures hard to draw? If yes, which one?
8. Did you, at any point of the experiment, tried to do something which you find difficult to do? If yes, what was it?
9. Did you, at any point of the experiment, felt like you needed a break? If yes, when?
10. Which authentication method would you prefer to login in HoloLens?
11. Which authentication method would you prefer to login in your everyday computer usage (e.g., emails, social networks, etc.)
12. Do you like graphical passwords as an alternative authentication method in HoloLens?

Figure 6.1 – Questionnaire of first User Study

6.2.3 Evaluation Analysis of Recall-based Desktop *vs.* MR

For the user study, we set three hypotheses that we wanted to check. Those hypotheses were:

H01. There is no significant difference between the time needed to create a picture password between users that utilize a mixed reality device *vs.* a desktop computer
H02. There is no significant difference in strength of user-generated picture passwords between users that utilize a mixed reality device *vs.* a desktop computer
H03. There is no general preference of users towards picture- or text-based passwords, considering main effects and interactions with respect to device used (mixed reality *vs.* desktop)

We recruited thirty participants, 20 males and 10 females of ages ranging from 22 to 40 years of age (m = 31.7, sd = 6.1), with limited to none experience on mixed reality devices and with no previous experience on picture passwords. Moreover, we followed a between-subjects design, so we formed two groups: i) The first one, where the participants had to interact with the picture password in Microsoft's HoloLens and included half of the participants; ii) The second one, where the rest of the participants had to interact with a desktop PGA.

For the analysis of our user study results, no outliers were found, and the data are mean ± standard error.

6.2.3.1 Investigation of *H01*

In order to investigate the validity of *H01*, we ran an independent-samples t-test, so that we can check if there is a statistically significant difference between the means in our two groups. We had set the user group (Mixed Reality *vs.* Desktop) as the independent variable and as the dependent variable, the time to create the picture password. For the first group, we found a time of 16.69 seconds for their password creation time and for the second

group, a time of 12.88 seconds, with a mean difference of 3.81 ± 3.02 (95% CI, -1.39 to 10.01), t(27) = 1.261, p = .281. Such a difference is not significant, so as the final result we found out that *there is no significant difference in Password Creation time between the two groups*. [Fig. 6.2] displays the creation times per user group for better readability.

6.2.3.2 **Investigation of *H02***

For evaluating this hypothesis, we had to run a brute force attack on the user generated passwords of each group in order to find out the total tries needed to break each password. For that we implemented a simple 2D array traversation, where we tested each coordination for a gesture match. The gestures were tested one-by-one and if a match was found, we continued to the next gesture of the password. In the end, the total tries for each gesture were multiplied with the rest and the final number was the total tries for breaking a password.

In order to investigate the validity of *H02*, we ran an independent-samples t-test to check whether the two groups had difference in password strength. We had set the user group (Mixed Reality *vs.* Desktop) as the independent variable and the password strength as the dependent variable. For the first group, we found a guessability value of 306 billion tries and for the second group, a guessability value of 310 billion tries, with a mean difference of 4.69 ± 3 billion guesses (95% CI, -1.08 to 1.47), t(27) = 1.562, p = .130. Such a difference is not significant, so as the final result we found out that *there is no significant difference in Pictue Password Strength between the two groups*. [Fig. 6.2] displays billion tries per user group for better readability.

6.2.3.3 **Investigation of *H03***

In order to investigate the validity of *H03*, we ran a chi-square test to check the association between the device type and the preference of the authentication type used (Text *vs.* Picture password). The expected frequency for all the cells was greater than five, so Yate's

correction for continuity was used. The results showed a statistically significant association between authentication type preference and device type with results: $\chi 2(1) = 8.571$, $p = .003$. Participants who interacted with the HoloLens, significantly preferred the picture password authentication mechanism ($p < .001$) and those who interacted with the desktop PGA significantly preferred the conventional text-password authentication mechanism ($p < .001$). The latter can be explained due to the familiarity of users when it comes to using textual passwords instead of graphical passwords. [Fig. 6.3] displays billion tries per user group for better readability.

6.2.3.4 **Summary**

Summing up, findings from this user study revealed that the differences in textual and picture passwords creation time as well as the number of guesses after a Brute Force attack that are required to break the passwords were not significantly different between the Mixed Reality and the Desktop group. Furthermore, after conducting questionnaires after the completion of the experiment with each user, the qualitative results that were received, showed a strong preference of picture password systems in mixed reality.

	Creation time (sec)	Guessability (billion)
Mixed Reality	16.69 (10.5)	306 (7)
Desktop	12.88 (4.96)	310 (1.1)

Figure 6.2 – Password creation times and guessability per user group

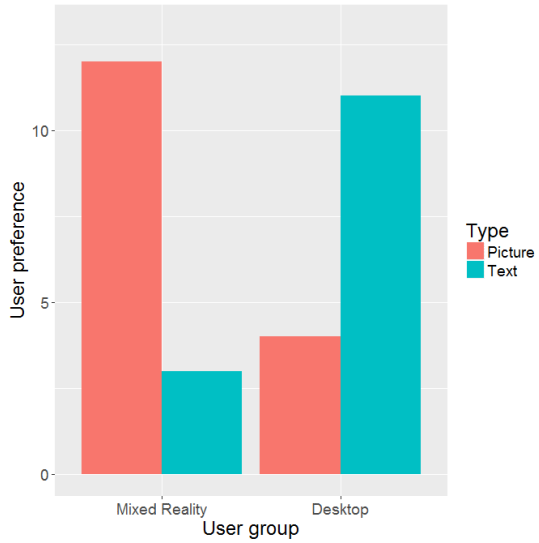

Figure 6.3 – User authentication preference per user group

6.3 Recognition-based System

6.3.1 Evaluation Scenario

Before evaluating the Recognition-based authentication system, we firstly had to think of a real-life scenario that we would present to the users, for them to interact with. Initially, we explained the purpose of the system and how it can be used in MR systems. Furthermore, we explained the main modules that compose the system, as well as the way to interact with them. Finally, we presented the users with a scenario that represents a common, every-day usage of the system where someone wants to use an application in HoloLens but has to log in to their account first. The first module that the users had to complete was a training module which required from the users to:

(i) Traverse through the menu and enter the training screen

(ii) Tap on three items to select them

(iii) If they wanted, they could keep tapping on items for as long as they liked, to get used to the interaction with the system

(iv) Once done, exit the training module and enter the main menu screen

After completing the training session of the system, the users were requested to proceed to registering a password. The tasks for this module were:

(i) Enter into the registration screen using the appropriate button in the main menu

(ii) Select any combination of five items that will serve as their password

(iii) Continue to the confirmation step of the password creation process

(iv) Re-enter the same five items in the same sequence as in the first registration step in order to confirm their password

Finally, after completing the registration step, they would have to log into a dummy program by entering their password. The tasks were:

(i) Enter the login screen by selecting the appropriate option in the main menu

(ii) Select the same five-item combination they entered in the registration step, in the same sequence, in order to login

6.3.2 Evaluation Process

The objective of carrying out this experiment was to evaluate if this new approach, in terms of authenticating users in Mixed Reality Head Mounted Displays, is more preferable by the users than the current authentication module, without affecting the security performances of the passwords. In order to do so, we decided to acquire quantitative, as well as qualitative data. For receiving the quantitative data, we implemented timers and other measurements for each task the user was executing. In detail, we measured:

(i) Time to select the first key

(ii) Time to select the second key

(iii) Time to select the third key

(iv) Time to select the fourth key

(v) Time to select the fifth key

(vi) How long it took for each user to complete the registration task (both registering and confirming the password together)

(vii) How long it took for each user to complete the login task

(viii) How many retries it took the user to complete the registration task

(ix) How many times did the user press the Reset button during the registration task

(x) How many retries it took the user to complete the login task

Moreover, each user visited an empty room in the University on a previously agreed date and time. At the end of each session, we asked users to complete a questionnaire with which we could see their preference in terms of difficulty in usability and preference of authentication system and also acquire our qualitative data.

After completing the process with each user, they were given to complete a questionnaire [Fig. 6.4] that it was used to receive the qualitative data regarding the project.

1. Was this your first time using the HoloLens?
2. Comments about the application?
3. What other electronic devices do you use in your everyday life?
4. How difficult was it to decide your password (1 – very difficult, 5 – very easy)?
5. Did you, at any point of the experiment, felt like you needed a break? If yes, when?
6. How long do you think it took you to create your password? (short, medium, long)
7. Using what logic did you choose your password?

Figure 6.4 – Questionnaire of second User Study

Furthermore, [Fig. 6.5, Fig.6.6] show the general selections that the users made in total on each grid system by coloring each grid item based on the selection frequency that they have been selected.

6.3.3 Evaluation Analysis of Recognition-based Desktop *vs.* MR

In order to continue with tha data analysis, we had to pull the study results from [3]. This study was focused on picture passwords on desktop environments, while utilizing a grid of 90 items as the users' array of options and the users had to select five keys as their password. For that study 24 participants were recruited, 9 males and 15 females of ages ranging from 18 to 22 years of age (m = 19.92, sd = 1.21).

For the user study, we set three hypotheses that we wanted to check. Those hypotheses were:

H01. There is no significant difference between the time needed to create a picture password between users that utilize a mixed reality device *vs.* a desktop computer
H02. There is no significant difference in strength of user-generated picture passwords between users that utilize a mixed reality device *vs.* a desktop computer
H03. There is no difference in the user experience of recognition-based graphical passwords that are deployed on mixed reality *vs.* desktop devices.

We recruited 30 participants, 16 males and 14 females of ages ranging from 18 to 32 years of age (m = 22.5, sd = 2.60), with limited to none experience on mixed reality devices and with no previous experience on picture passwords. Moreover, we followed a between-subjects design, so we formed two groups: i) The first one, where the participants had to interact with the picture password in Microsoft's HoloLens and included 30 of the participants; ii) The second one, where the participants were from [3] and had to interact with a desktop PGA.

For the analysis of our user study results, no outliers were found, and the data are mean ± standard error.

6.3.3.1 Investigation of *H01*

In order to investigate the validity of *H01*, we ran an independent-samples t-test, so that we can check if there is a statistically significant difference between the means in our two groups. We had set the user group (Mixed Reality *vs.* Desktop) as the independent variable and as the dependent variable, the time to create the picture password. For the first group, we found a time of 55.42 seconds for their password creation time and for the second group, a time of 32.08 seconds, with a mean difference of 23.34 ± 6.82 (95% CI, 9.65 to 37.03), $t(27) = 3.422$, $p = .001$. Such a difference is significant, so as the final result we found out that *there is significant difference in Password Creation time between the two groups* which makes this form of picture password more efficient on desktop devices, when compared to MR devices.

6.3.3.2 Investigation of *H02*

In order to check the validity of *H02*, we ran a brute force attack of every possible 5-key password combination in order to find out the total attempts needed in order to break each password in both Desktop and HoloLens. The average needed attempts for the Desktop grid system were 2,708,735,227 (2.708 billion), while for the HoloLens grid system 2,325,933,494 (2.325 billion). The attempts for both systems are in the same scale so we can declare that there are no differences in terms of password strength.

6.3.3.3 Investigation of *H03*

In order to investigate the validity of this hypothesis, we collected qualitative data from our user study and also used the qualitative data used in [3].

From [3], participants of the 90-image grid, reported that they browsed through most of the grid before making their selection. Moreover, their selection was affected by their preference of categories (e.g. hobbies, food, etc.). Meanwhile, others created a password that describes a story based on their individual experiences. A user stated: “I love sweets, so I selected my favorite sweets such as Haribos”, while another stated, “I am hungry now, so I selected my favorite food, starting from pizza”. In addition, most of the users stated that they did not face any difficulties during login and they also responded positively when asked if they believe that they would remember their password after one month.

For our user study in MR, most of the users reported that they browsed through most of the grid before making their password selection. Additionally, for most of them, their password selection was also affected by their category preference or it was a story based on their individual experiences, similar to what Desktop users experienced. Although, most of them reported that the whole process was a bit tiring because of the constant head movement in order to see all of the items. Finally, almost all of the users had no trouble during login and most of them responded positively when asked if they though that they would remember their password after one month. Some user statemets are: “I really like the simplicity of the grid. It’s just plain icons of every-day items, so it is easier to remember my password”, “Even though the system is simple with just a simple click for interacting, I find it a bit tiring that I have to keep moving my head around in order to see the whole grid”.

Based on those comments, we can see that the general preference of such password schemes in both systems is acceptable with good user experience, in general, although some improvements in terms of comfortability can be made in the HoloLens system.

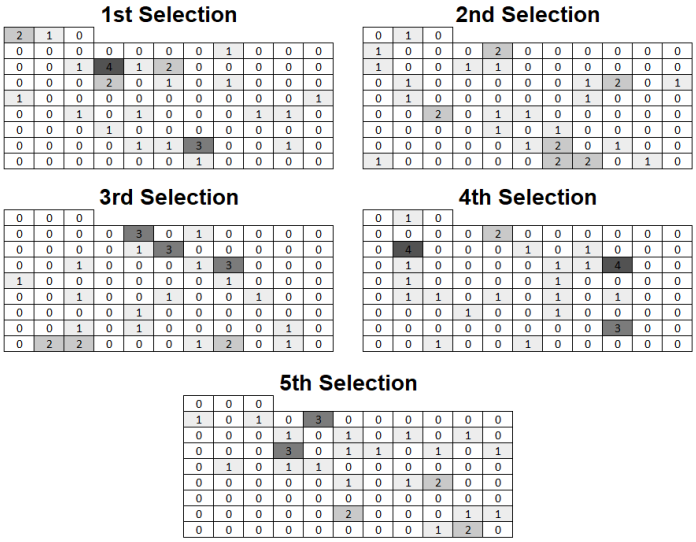

Figure 6.5 – Users' selections on MR Recall-based

Figure 6.6 – Users' selections on Desktop Recall-based

6.3.4 Evaluation Analysis of Recall-based MR *vs.* Recognition-based MR

For the user study, we set one hypothesis that we wanted to check. That hypotheses were:

H01. There is no significant difference between the time needed to create a picture password between users that utilize the Recall-based *vs.* the Recognition-based graphical passwords on mixed reality devices.
H02. There is no significant difference between the time needed to log into a picture password system between users that utilize the Recall-based *vs.* the Recognition-based graphical passwords on mixed reality devices.
H03. There is no difference in the user experience of recall-based *vs.* recognition-based graphical passwords that are deployed on mixed reality devices.

We used the data gathered from the other two studies for both, the Recall-based system and the Recognition-based system and also, we extended the first study in order to have more data. In total, 55 participants, 37 males and 18 females of ages ranging from 18 to 32 years of age (m = 22.82, sd = 2.63), with limited to none experience on mixed reality devices and with no previous experience on picture passwords. Moreover, we followed a between-subjects design, so we formed two groups: i) The first one, where the participants had to interact with the recall-based picture password in Microsoft's HoloLens and included 25 of the participants; ii) The second one, where the rest of the participants had to interact with the recognition-based picture password in Microsoft's HoloLens.

For the analysis of our user study results, no outliers were found, and the data are mean ± standard error.

6.3.4.1 **Investigation of *H01***

In order to investigate the validity of *H01*, we ran an independent-samples t-test, so that we can check if there is a statistically significant difference between the means in our two groups. We had set the user group (Recall-based *vs.* Recognition-based) as the independent

variable and as the dependent variable, the time to create the picture password. For the first group, we found a time of 55.42 seconds for their password creation time and for the second group, a time of 23.29 seconds, with a mean difference of -32.12 ± -45.53 (95% CI, -45.53 to -18.72), t(27) = -4.812, p = .000. Such a difference is significant, so as the final result we found out that *there is significant difference in Password Creation time between the two groups* which makes the Recognition-based form of picture passwords more efficient in MR, when compared to the Recall-based form.

6.3.4.2 **Investigation of *H02***

In order to investigate the validity of *H01*, we ran an independent-samples t-test, so that we can check if there is a statistically significant difference between the means in our two groups. We had set the user group (Recall-based *vs.* Recognition-based) as the independent variable and as the dependent variable, the time to create the picture password. For the first group, we found a time of 15.03 seconds for their password creation time and for the second group, a time of 16.17 seconds, with a mean difference of -1.14 ± 2.11 (95% CI, --5.41 to 3.13), t(27) = -.540, p = .593. Such a difference is not significant, so as the final result we found out that *there is no significant difference in Password Login time between the two groups*.

6.3.4.3 **Investigation of *H03***

In order to investigate the validity of this hypothesis, we collected qualitative data from both user studies.

For the Recognition-based system, the user reviews can be found in section 6.3.3.3.

As for the Recall-based system, most of the users stated that the gesture-creation process was quite easy, although improvements can be made for the circle-creation process as well as the real-time drawing of the line. Moreover, participants found it easier during login and that is explained due to the experience they gain through interaction with the system.

Some user statements were: “I liked the variety of gestures and the ease of their creation.” and "In the beginning it was quite hard because it is my first time but with more practice, I could create a password in no time".

Based on those comments, we can see that the general preference of such password schemes in both systems is acceptable with good user experience, in general, although some improvements in terms of comfortability can be made in the Recognition-based system and some improvements in terms of the gesture-creation process in the Recall-based system.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This dissertation's purpose was the design and implementation of two Graphical Password Authentication systems for Mixed Reality systems and specifically Microsoft's HoloLens. The first system is a recall-based system, where the user has to draw on a background cue image their pre-registered sequence of three gestures (dot, line, circle); and the second system is a recognition-based system, where the user has to recognize and select their sequence of five keys from an array of images.

By implementing these authentication schemes, we were able to study whether users prefer picture passwords instead of conventional text passwords when they are trying to authenticate themselves in Mixed Reality contexts. Our studies' results showed that such a hypothesis is valid and further study as well as improvement in terms of development and design are needed. Furthermore, such schemes can be further expanded in other Head Mounted Displays such as Virtual and Augmented Reality HMDs.

7.2 Limitations

During the development of these systems, the evaluation and also the comments received from the users, we were able to detect limitations regarding the technologies used. These limitations are:

1. The processing power of the HoloLens is still not very good. For that reason, optimization and low-poly textures have to be used in order to escape frame drops and input lag while operating the device.
2. The Field of View of the HoloLens (FOV) is very low. The FOV is only 30°×17.5° [14, 16] and that creates problems in terms of immersion and operability, since many items are rendered outside of the user's FOV and they have to move their head constantly in order to view them.
3. HoloLens has an uneven weight distribution and the weight is more noticeable on the front of the device, so most of the users were complaining about the device being very heavy on their nose and forehead. Even though a strap is provided that helps by shifting the weight distribution a bit towards the back, the difference does not help.
4. The device is not friendly to glass-wearing users, so we had to reject many participants who were wearing glasses because HoloLens would not fit on their head.
5. In the first experiment, while operating Pupil Labs' hardware for HoloLens, it would quickly heat up and the exposed heat was really noticeable on the users' eyes.
6. Moreover, on the first system, while using Unity 3D and Pupil Labs' software, Unity would crash and close. With further investigation we found out that the two applications present crashing behavior for unknown reasons while operating together.

7.3 Future Work

With this thesis we tried to cover the whole spectrum of graphical passwords by developing both a recall- and a recognition-based authentication scheme. In that regard,

both systems are prototypes with the plain functionality of registering and logging into an account without any regard on security and usability.

The recall-based system can be further developed in order to improve the gesture drawing mechanics. For example, new custom gestures can be introduced in HoloLens that would generate new input values that could be used for drawing the gestures. Also, the user interface can be further designed to improve readability and usability so that moving from one screen to another can be animated and further leverage the capabilities of Mixed Reality in this regard. Finally, the security aspect of the system can be studied and develop mechanisms that would offer security against more advanced attacks as well as shoulder-surfing attacks.

The recognition-based system can be further developed in order to improve the usability of the system. For example, animations and new interaction modules can be introduced in order to improve the user experience. Also, the system's user interface can be further designed and enhanced so that the system can be more easily operated. Finally, the system's security should be studied in order to generate hack-protection mechanisms to prevent attacks and solidify the security of the system.

Bibliography and References

[1] Adams, A., & Sasse, A (1999). Users Are Not the Enemy. Commun. ACM. 42. 40-46. 10.1145/322796.322806.

[2] Belk M., Fidas C., Germanakos P., Samaras G. (2017). The interplay between humans, technology and user authentication: a cognitive processing perspective, Computers in Human Behavior, 184-200, 76: 184-200.

[3] Belk, M., Pamboris, A., Fidas, C., Katsini, C., Avouris, N., Samaras, G. (2017). Sweet-spotting security and usability for intelligent graphical authentication mechanisms, ACM Web Intelligence, ACM, 252-259

[4] Biddle, R., Chiasson, S., & van Oorschot, P. (2012). Graphical passwords: Learning from the first twelve years. ACM Computing Surveys, 44(4), 41

[5] Brostoff, S., & Sasse, M. A. (2000). Are passfaces more usable than passwords? a field trial investigation. In People and Computers, Usability or Else, 405-424.

[6] Chiasson, S., Van Oorschot, P., & Biddle, R. (2007). Graphical password authentication using cued click points. In Computer Security–ESORICS, 359-374

[7] Dhamija, R. & Perrig, A. (2000). Deja vu-a user study: Using images for authentication. In USENIX Security Symposium, volume 9, page 4.

[8] Dunphy, P., & Yan, J. Do background images improve draw a secret graphical password? In ACM CCS 2007, 36-47

[9] Findlater, L., Wobbrock, J., & Wigdor, D. (2011). Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces. In ACM CHI 2011, ACM Press, 2453-2462.

[10] George, C., Khamis, M., Zezschwitz, E. V., Burger, M., Schmidt, H., Alt, F., & Hußmann, H. (2017). Seamless and secure vr: Adapting and evaluating established authentication systems for virtual reality. In USEC2017.

[11] Jermyn, I., Mayer, A., Monrose, F., Reiter, M., & Rubin, A. (1999). The design and analysis of graphical passwords. In USENIX Security Symposium 1999.

[12] Johnson, J.J., Seixeiro, S., Pace, Z., van der Bogert, G., Gilmour, S., Siebens, L., & Tubbs, K. (2014). Picture gesture authentication. Retrieved from https://www.google.com/patents/US8910253

[13] Katsini, C., Fidas, C., Belk, M., Samaras, G., & Avouris, N. (2019). A Human-Cognitive Perspective of Users' Password Choices in Recognition-Based Graphical Authentication. International Journal of Human-Computer Interaction, 1-13

[14] Kreylos, O., (2015). "On the road for VR: Microsoft HoloLens at Build 2015, San Francisco". Doc-Ok.org. Retrieved May 13, 2015

[15] Mihajlov, M. & Jerman-Blazic, B. (2011). On designing usable and secure recognition-based graphical authentication mechanisms. Interacting with Computers, 23(6):582-593

[16] Milgram, P., & Andkishino, F. (1993). A taxonomy of mixed reality visual displays. IEICE Trans. on Information and Systems E77-D, pp. 13–21.1.1

[17] Roesner, F., Kohno, T., & Molnar, D. (2014). Security and privacy for augmented reality systems. Commun. ACM 57, 4, 88-96.

[18] Schneegaß, S., Oualil, Y., & Bulling, A. (2016). SkullConduct: Biometric user identification on eyewear computers using bone conduction through the skull. In ACM CHI2016, ACM Press, 1379-1384

[19] von Zezschwitz, E., De Luca, A., & Hussmann, H. (2014). Honey, I shrunk the keys: Influences of mobile devices on password composition and authentication performance. In NordiCHI 2014, ACM, 461-470.

[20] Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., & Memon, N. (2005). Authentication using graphical passwords: effects of tolerance and image choice. In ACM SOUPS 2005, ACM Press, 1-12.

[21] Yadav, D. K., Ionascu, B., Ongole, S. V. K., Roy, A., & Memon, N. (2015). Design and analysis of shoulder surfing resistant PIN based authentication mechanisms on google glass. In Financial Cryptography and Data Security 2015, Springer Verlag.

[22] Yan, Q., Han, J., Li, Y., & Deng, H. (2015). Leakage Resilient Password Systems: Attacks, Principles, and Usability. 10.1007/978-3-319-17503-4_1.

[23] Yu, Z., Liang, H. N., Fleming, C., & Man, K. L. (2016). An exploration of usable authentication mechanisms for virtual reality systems. In IEEE APCCAS 2016, IEEE, 458-460.

[24] Zhao, Z., Ahn, G., Seo, J., & Hu, H. (2013). On the security of picture gesture authentication. In USENIX Security 2013. USENIX Association, 383-398