Virtual Login in XR

Yujian Zhao, Yvonne Zhou
yzhao124@umd.edu, skyzhou@umd.edu

Thank for **Htet Aung**(htetmyataung2027@gmail.com) helping setup the database

1. Introduction

The normal way we used to signup and login is through the simple process of setting up usernames and passwords on the website. The password will be encrypted and stored. The whole process is on the screen and users will only interact with the keyboard. In this project, we will present a way that allows users to set up the password and log in with the password in the XR experience. We design two different scenes for the XR login process: Diagon Alley and Memory Palace.

2. Related Work

In *Deja Vu–A User Study*[1], it summarizes from some cognitive science experiments papers, that humans are much easier to recognize things, which here refers to images, than recall the information, which here refer to text-based passwords. It also points out that using images recognition rather than string password can lessen users' cognitive load and provide a more pleasant user experience. While image recognition is easier to be memorized and pleasant to be used in user authentication, *User authentication by Secured Graphical Password Implementation*[2] committed graphical password schemes are secure. In our project, we want to propose a new approach, virtual login scheme, using in XR application. Virtual login has similarity with graphical passwords, in which they both use human's object recognition as password. Therefore virtual login should have the same benefits as graphical login. Meanwhile, since virtual login allows users to interact with virtual objects, additional info of user's interaction would augment the security of user authentication.

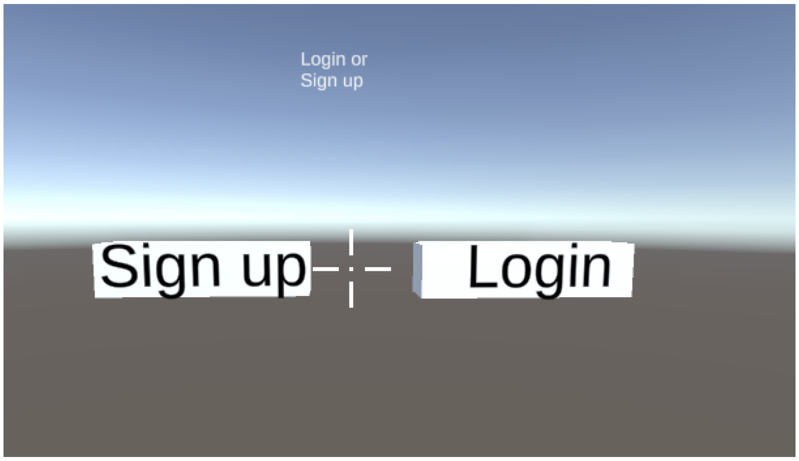
The encryption scheme for traditional textual passwords is hashing, *Defuse Security*[3] website article has a very detailed explanation on the hashing algorithm, and shows it is the best way to encrypt textural passwords. So our project will use hashing as the base encryption algorithm to encrypt our virtual passwords, but with little twist.

3. Our Contribution

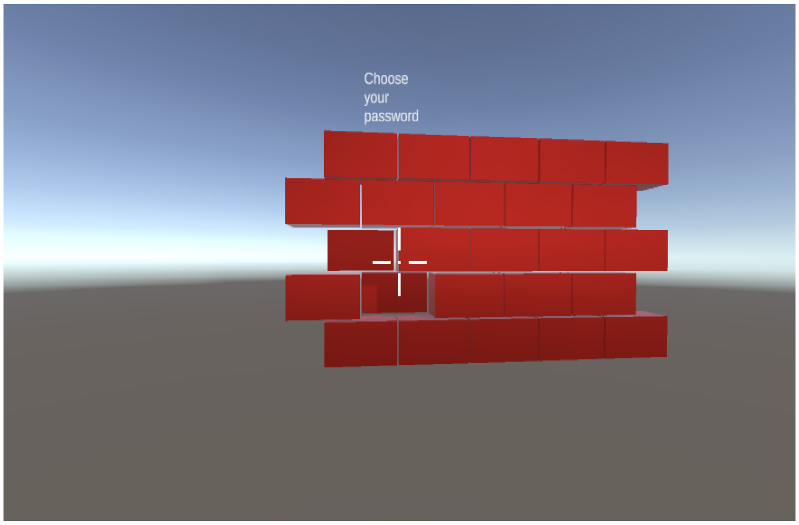
The most common authentication login method is textual password authentication, There are also some applications of graphical passwords. Our project is proposing a state of art authentication login approach, virtual login. We grant users to interact with virtual objects, and use hashing cryptography schemes to encrypt both objects' information and user's movements as authentication passwords.
Below we present two XR scenes in virtual login: Diagon Alley, and Memory Palace.

3.1 Diagon Alley

We get this idea from the famous movie, Harry Potter. In the movie, Harry Potter was able to enter the Diagon alley by hitting certain bricks on the wall. We try to implement a similar scene in Unity for signup and login processes, and we use MongoDB to store the username and password.



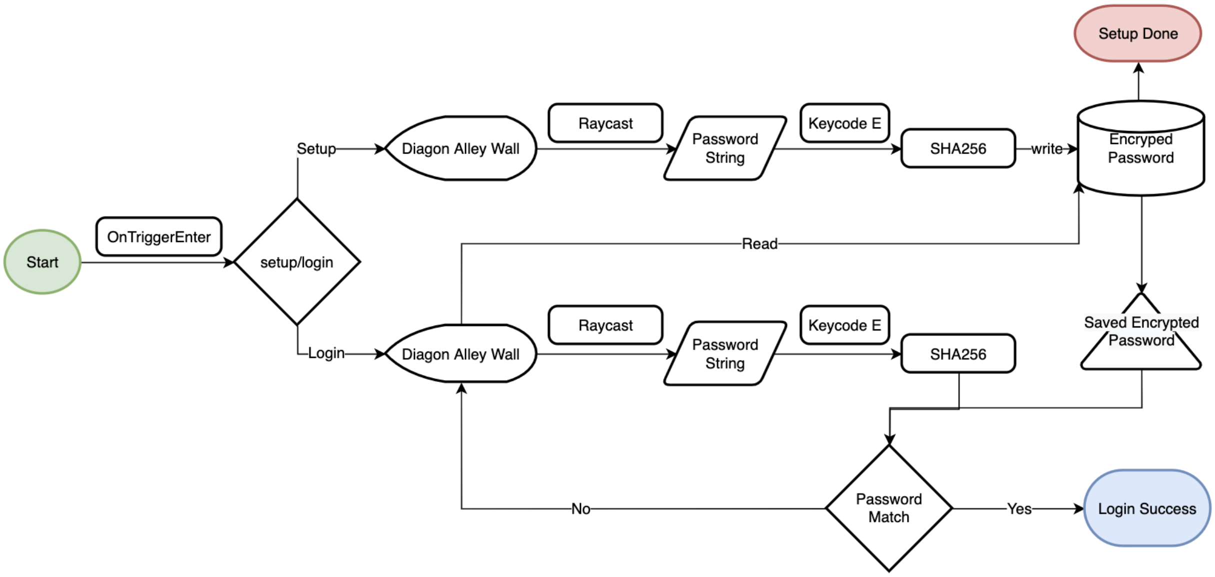
At the start of the scene, we created two cubes that represent the signup and login. Users can select which process to precede with by moving the camera and making a collision between the camera and the cube object. The text on the top of the screen shows the instruction on what the user should do in this step. The aim sign in the middle of the screen is used to help the user hit the right target in the next step. The instruction and the aim sign are attached to the camera which will move with the camera.



When the user collides with either the signup box or the login box, a wall of red bricks will be generated automatically.
Users can use the aim sign in the middle to help them point to the right brick they want.

This process is implemented using Raycast in Unity. After pressing "R" on the keyboard, a string containing the name of the bricks being hit will be recorded. Pressing "E" on the keyboard will be the sign to end the process of choosing the password. Then the recorded string will be encrypted and stored in the database. We did not design the scene for entering usernames in VR, since the purpose of this project is to design VR scenes for choosing passwords. In our implementation, the username is hard-coded in order to put the username-password pair into the database. The text on the top will change to "Success".

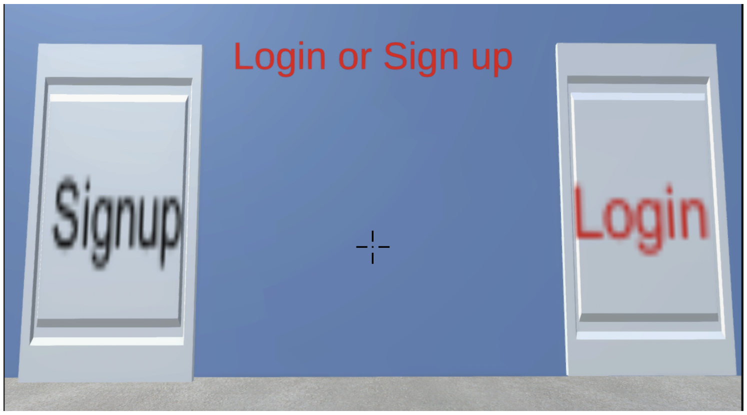
For the login process, the user may enter the wrong password by accident. Therefore, we support the re-entering of the password after the failure. If the initial password is wrong, the wall will reset and the user can enter the password again.



Working Flowchart of Diagon Alley

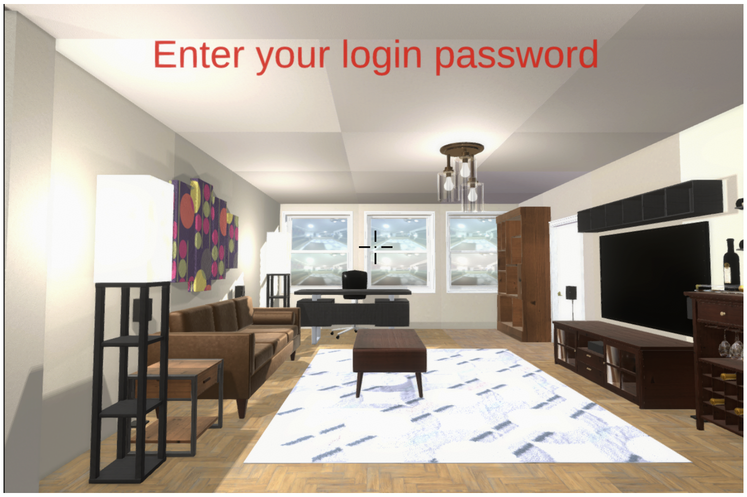
3.2 Memory Palace

According to Wikipedia, the memory palace (the method of loci) is "a strategy of memory enhancement which uses visualization of familiar spatial environments in order to enhance the recall of information". We use this concept to create a virtual scene for users to use the pattern of picking up different objects in the room as the password. We use the template of the apartment created by Break Project Studio from the Unity Asset Store, https://assetstore.unity.com/packages/3d/props/apartment-kit-124055#publisher. Everything else from the looking of the apartment was created by ourselves.

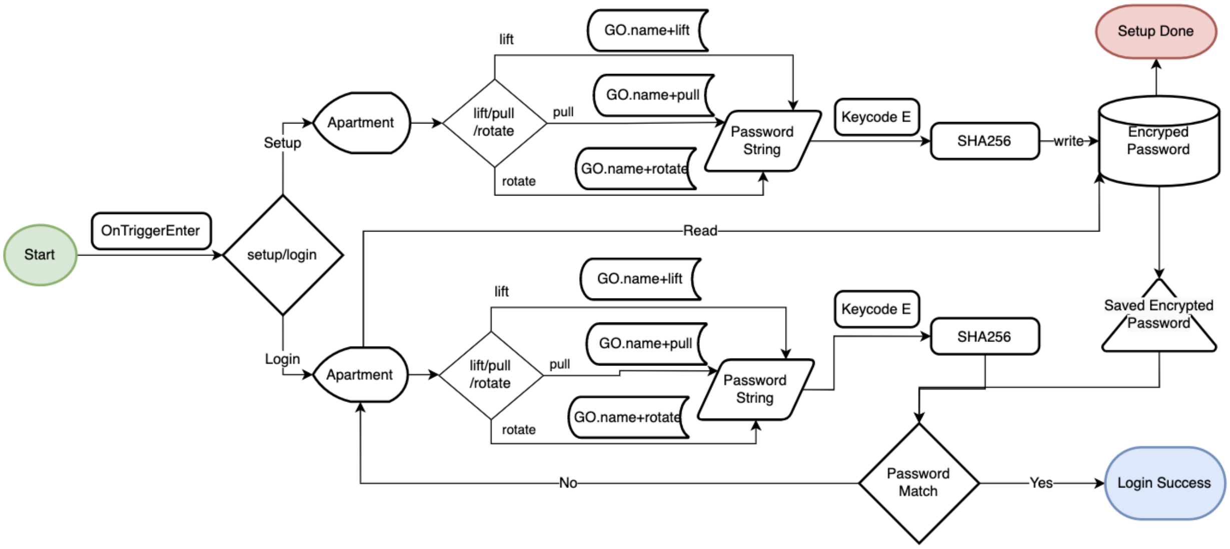


When users enter the scene, they will see two doors representing the signup and the login. Users can move the camera to hit one of them and tell the system which process they choose to proceed with. When users pass through this door, they will see the entrance of the apartment.

After entering the room, users can start to choose objects for signup or login. The object selection process is implemented by Raycast in Unity. In our scene, when the ray hits one object, there are three different types of movement that can be applied to the object. If the user presses “R” on the keyboard, the object will be rotated. If the user presses “M” on the keyboard, the object will be moved up. If the user presses “N” on the keyboard, the object will be moved forward. The name of the object plus the way of its movement will be appended to the password string. The end of inputting the password will be the press “E” on the keyboard. In the signup process, the password string will be encrypted and stored in the local file. In the login process, the encrypted password will be compared to the password in the local file the verify the users’ identity.



Resetting the password is still supported in this scene. The movement of the object is temporary. The object will move back after a few seconds. Therefore, when the password is wrong, the scene will remain unchanged and users can reselect their password.



Working flowchart of Memory Palace

3.3 Authentication

As an authentication login method, one major aspect needs to be considered is: are users' passwords secure? How to prevent the password from breaching, while the applications that users are using are under attack. The solution is never save the password itself directly, instead, always encrypt the password before saving down. And guarantee that the adversary is unable to know any information from the encrypted password. Employing a hashing function on password is a proven secured method. So in our project we are going to use cryptographic hash functions, SHA256, as our encryption scheme. The reasons we pick SHA256 is that, it is extremely hard for the attacker to reconstruct the password, which requires 2^256 attempts; a very minor change of input password would alter the hash value dramatically; and it is collision free.

However before applying SHA256, there is some preliminary work we need to do. Unlike the traditional textual password, which hashing function can simply map a text-based password to another pseudo random string. Virtual login gets users' input as virtual objects and their movements. Therefore, before using SHA256, our system will convert the virtual password as a string list combining the user-picked objects' information and their corresponding movement. The list will be appended when users select more objects to interact with. Then we use SHA256, to encrypt the converted password.

4. User Study

There are some user studies we would like to follow up to testify our new authentication approach. Basically, we want to compare the utility of traditional textual password to our virtual password. The study chart may look like something below:
1. Testify to the failure rate of users when they use different login approaches.
2. Time spent for users to process setup/login.

	Textual password	Virtual Login
One the day create		
(7/30/60) days after create		

3. Entertainment rate from users for text login and virtual login.
These user studies are planned as our future work.

5. Conclusion and Future Work

We have successfully implemented two ways of virtual logins. Our work allows users to have a virtual experience in choosing the password. Our work also broadens the idea of how to design the authentication process.

After this project, there are still many future works that can be done. First, we should do a user study to gather feedback about the experience of using our virtual logins. Second, we can come up with other virtual scenes that can be used as an authentication process. At last, we can explore different ways of encrypting the password and we can attack the process to test if it is safe or not. We hope our work can inspire further research to develop scenes for virtual logins and enhance the understanding of how to combine the concepts of XR and cryptography.

6. Other Project Resources:

6.1. Video demos

- XR Diagon Alley login demo: https://www.youtube.com/watch?v=edZ-AGMHNII&t=3s
- XR Memory Palace login demo: https://www.youtube.com/watch?v=obv9mejClt4

6.2 Source code

- Unity Project for Diagon Alley: https://github.com/YujianZhao-080910/838C-Final-Project-Wall
- Unity Project for Memory Palace: https://github.com/YujianZhao-080910/838C-Final-Apartment
- Database: https://github.com/denteyon/DB-backend

7. References

1. Rachna Dhamija and Adrian Perrig. Deja Vu–A user study: Using images for authentication. In the 9th USENIX Security Symposium (USENIX Security 00), Denver, CO, August 2000. USENIX Association.
2. S. K. Bandyopadhyay, D. Bhattacharyya and P. Das, "User authentication by Secured Graphical Password Implementation," 2008 7th Asia-Pacific Symposium on Information and Telecommunication Technologies, 2008, pp. 7-12, doi: 10.1109/APSITT.2008.4653531.
3. Defuse Security. (n.d.). *Salted password hashing - doing it right*. CrackStation. Retrieved May 18, 2022, from https://crackstation.net/hashing-security.htm