



Group 24

IBM Design Thinking – VR and AI

User Manual

Change Log:

Version	Date	Changes
1.0	15/3/2021	<ul style="list-style-type: none">- Room Generation- Movement and looking around- Interaction with virtual objects(i.e. Whiteboard and pen)- Database Design via MongoDB- Interface - Menu- IBM AI - Watson Speech to text

Contents:

1 Overview	4
1.1 Product Description	4
1.2 How to use the product manual	4
2 Prerequisites	5
2.1 How to set up	5
2.1.1 Obtaining Source Code & Unity	5
2.1.2 IBM Cloud Access	6
2.1.3 Loading software on to VR Headset	9
2.2 Hardware instructions	11
2.2.1 How to use the controller	11
3 Using the software	12
3.1 Conformance to requirement specifications	12
3.2 General user flow within the software	14
3.3 Software Features	15
4 Information for Developers	22
4.1 Unity maintenance/modification	22
4.1.1 Upgrading the version of unity used for this software	22
4.1.2 Functions within unity	24
4.1.3 Updates for Assets and Packages	25
4.2 IBM Watson	28
4.2.1 Requirements to use Watson	28
4.2.2 Unity settings and prerequisites	30
4.3 Database maintenance	33
4.3.1 Access to the cluster	34
4.3.2 Installing MongoDB shell and Node.js	34
4.3.3 Database Structure	35
4.4 Issues with the current program	36
4.4.1 Table of current issues	36

4.4.2 Current issues with the menu and log in section	38
4.5 Future changes and implementations	42
4.5.1 Multi user functionality	42
4.5.2 Additional Watson services	42
4.5.3 Database implementation - importing and exporting work	42
4.5.4 Additional sticky note function	43
4.5.5 Customisation	43

1 Overview

This is a manual and supporting documentation for a VR implementation of the IBM Design Thinking Process.

IBM uses Design Thinking for solving problems, building products and providing services for their customers. Design Thinking is a strategic framework for problem-solving and innovation. There are many variations of Design Thinking, but the main elements of the IBM process include: focusing on user needs, restless invention, constant sharing and reflection of development progress amongst team members, and frequently engaging with testers for real-world feedback.

1.1 Product Description

The goal of our project is to improve the effectiveness of IBM's Design Thinking process by making it more engaging using VR. As such, the software that we have built is a VR program that recreates the IBM Design Thinking process in a virtual environment. This software is intended to be used by IBM employees that wish to engage in IBM Design Thinking.

Here are the main features included in the software:

- Multiple rooms equipped with appropriate furniture and tools for engaging in each stage of IBM Design Thinking.
- Ability to navigate between these rooms and interact with the virtual objects.
- Watson Speech-to-Text functionality for transcribing speech onto virtual sticky notes.
- Draw on virtual whiteboards with virtual pens.

More information on how to access or modify these features will be further elaborated on in the sections below.

1.2 How to use the product manual

This document will act as a manual, both for users and future developers of the software.

For the users, we will provide instructions on how to use the product; including how to set up the software for their use, and how to interact with it. We will provide examples of user journeys, so that the user is aware of how each function works so that they can use it effectively.

For the developers, we will provide detailed instructions on how they can contribute to the code, allowing them to implement changes to the software, and how to set up the software so that they can make these changes. As well as this, we have provided instructions on how they are able to maintain and upgrade the system, any issues within the current system with suggested solutions, and suggestions for future functions/ improvements within the software.

2 Prerequisites

In order to use this software, there are a few prerequisites. These include: Oculus Quest 2 headset (with at least 200MB memory) and controllers, alongside added software which we will now explain how to set up.

2.1 How to set up

This software was created with the Unity Game engine using Unity Editor. Ensure that you at least use Unity version 2020.3.22f1 or above. If you do not have Unity, you may download it from <https://unity.com/download>.

2.1.1 Obtaining Source Code & Unity

In order to set up the system to allow you to run it and make changes, you will need to follow the following steps:

1. You will first need to clone the system repository, into a folder of your choice, this can be done using the command line, or GitHub desktop. Using the command line, this can be done by running -

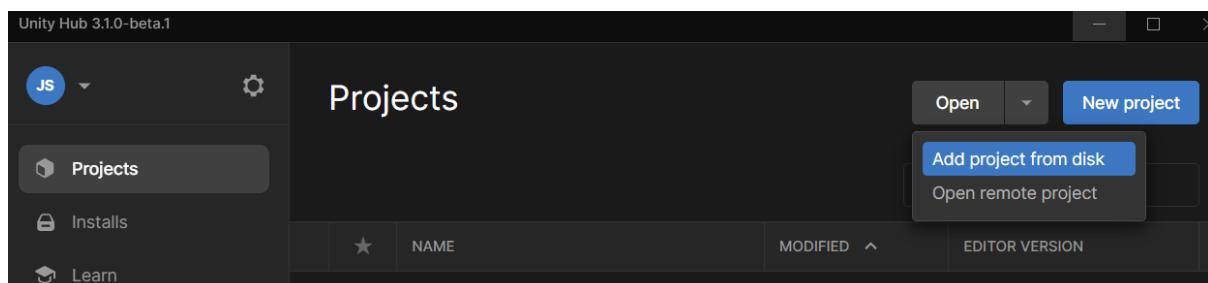
`git clone`

https://github.com/Vincent-Zhenhao-ZHAO/COMP2281_VR-AI-Design-Thinking

```
C:\Users\lucas>git clone https://github.com/Vincent-Zhenhao-ZHAO/COMP2281_VR-AI-Design-Thinking
```

This will create a remote repository, which will allow you to look at the code and all the files used for the project on your own machine.

2. Once you have cloned the repository, open up Unity Hub. Select the “Projects” tab on the left, click on the drop-down arrow next to “Open”, and select “Add project from disk”.

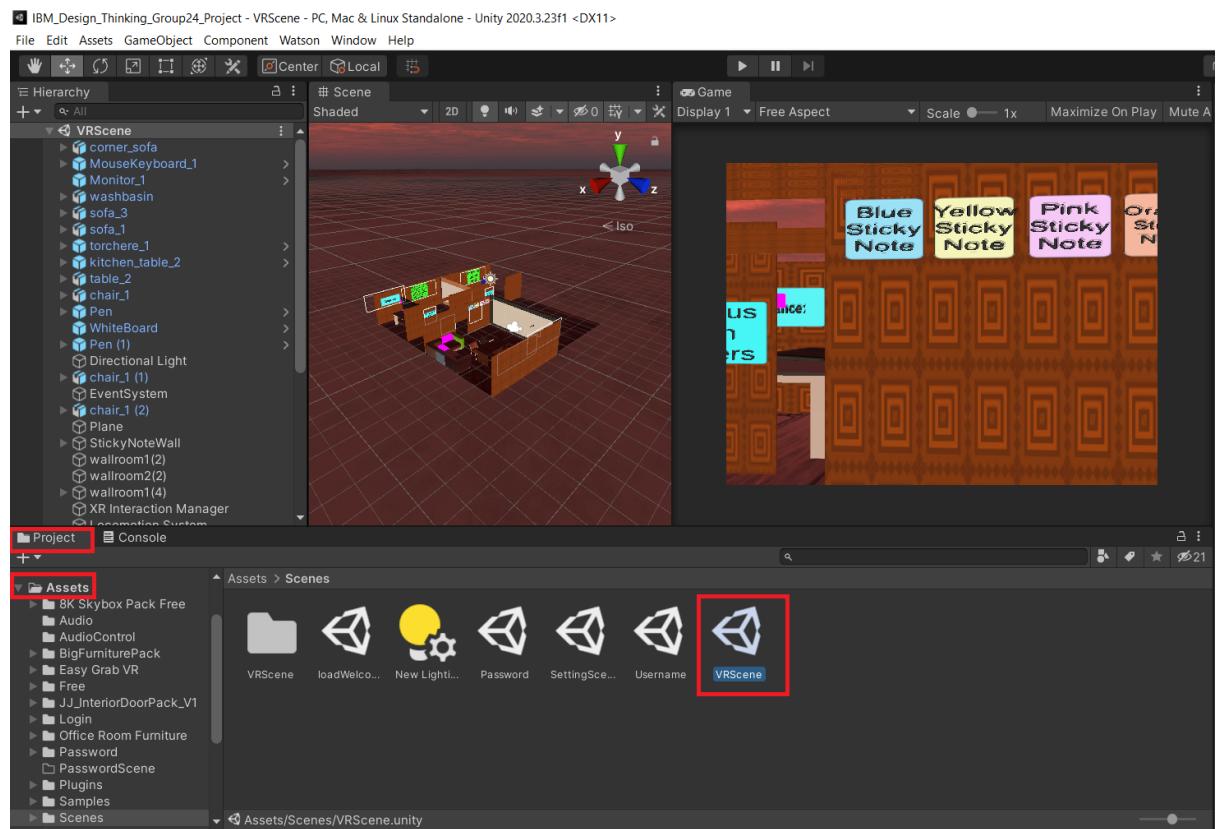


3. Navigate to the folder where you cloned the repository. Select the “IBM_Design_Thinking_Group24_Project” folder and click “Add project”. Unity will then import all the assets and open up the software within Unity Editor. (If you encounter a warning regarding using a different editor, just ensure that you select an editor version that is 2020.3.22f1 or later).

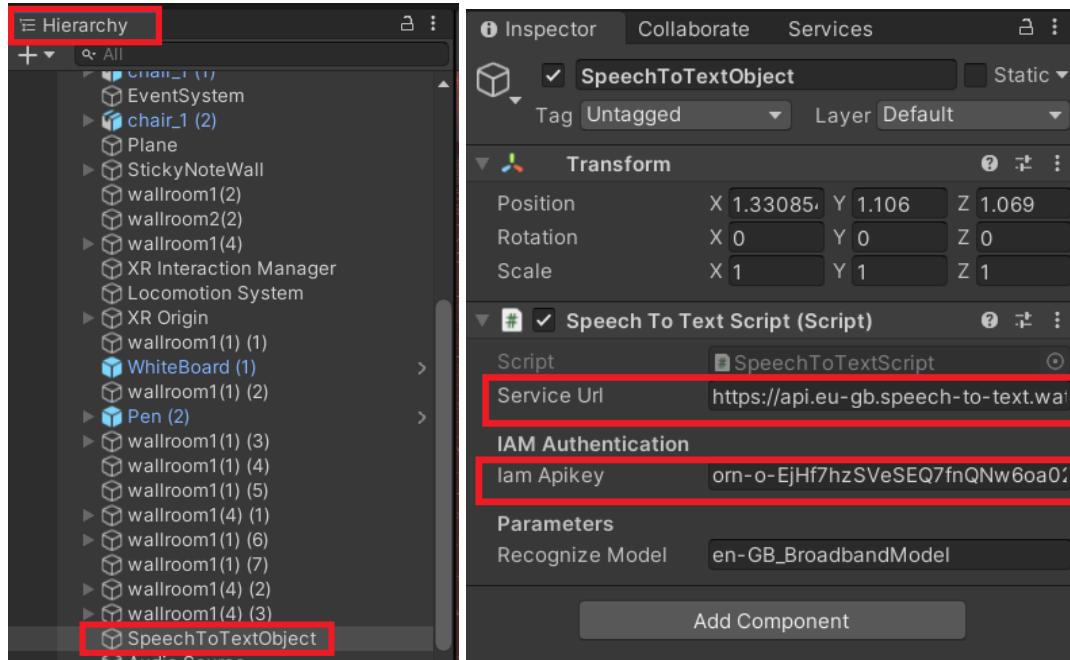
2.1.2 IBM Cloud Access

This software uses Speech-to-Text technology which is powered by IBM Cloud. There are scripts within our code that make API calls to IBM Cloud to use the service. In order to use this service, you need to have a valid IBM account and have an existing Speech-to-Text service set up. If you have neither of those, you may find more information on how to obtain them in the XXXXX section below. First, we will navigate to the component within Unity that requires you to input your IBM credentials.

4. Select the “Project” tab on the bottom half of the Unity editor, next to the Console tab. Click on “Assets”, double-click the “Scenes” folder, and then double-click “VRScenes.unity”. This will open up the scene with all the rooms.



5. In the “Hierarchy” tab on the top left, expand the “VRScene”, scroll down and click on “SpeechToTextObject”. An inspector tab should appear on the right side. Expand the “Speech To Text Script (Script)” header if not already expanded. This is where you will need to input the IBM API credentials, in the “Service Url” and “Iam Apikey” fields.

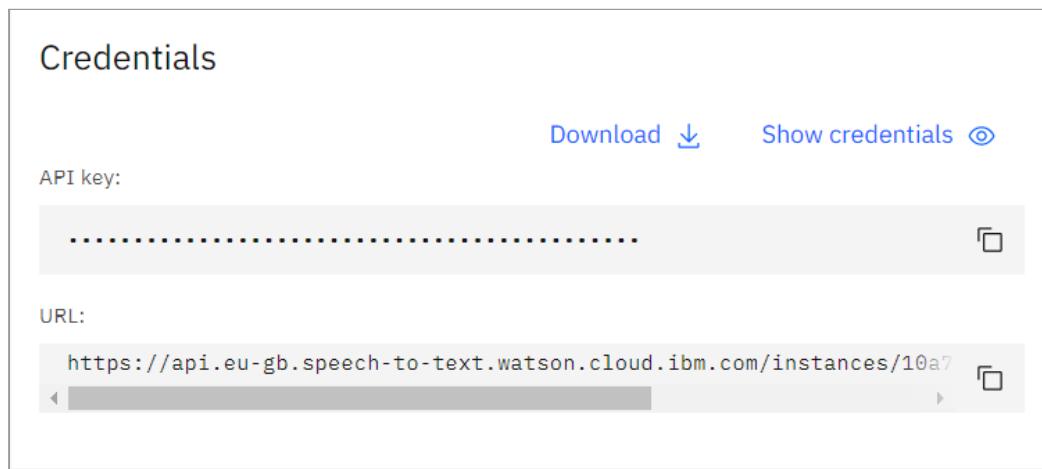


6. Log in to your IBM Cloud account at ibm.com/uk-en/cloud. At the home page/dashboard, under the “Resource Summary” section, click on “Services and Software” and select the Speech-to-Text service.

The image shows the IBM Cloud dashboard. On the left is a sidebar with icons for various services. The main area is titled 'Dashboard'. Below it is a 'For you' section with three cards: 'Build' (Explore IBM Cloud with this selection of easy starter tutorials and services), 'Monitor your resources' (Get visibility into the performance and health of your resources), and 'An essential guide to Kubernetes' (Deploy, scale, and manage your containerized applications with Kubernetes). At the bottom is a 'Resource summary' section. It shows '1 Resources' and a 'Services and software' link. There is also a notification icon with a green checkmark and the number '1'.

Services and software (1)				
Speech to Text-d2	Default	London	Speech to Text	Active

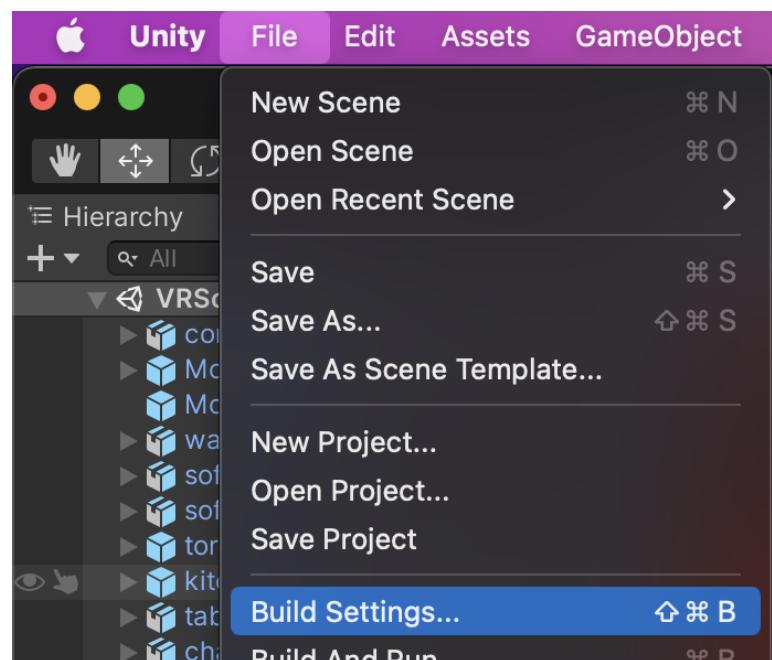
7. You should now see the credentials page as shown. Copy them to the corresponding fields in the “Speech To Text Script” component of the “SpeechToTextObject” that you found in Unity in Step 5. Now the scripts in Unity will have the necessary credentials and information to access your Watson Speech-to-Text service.



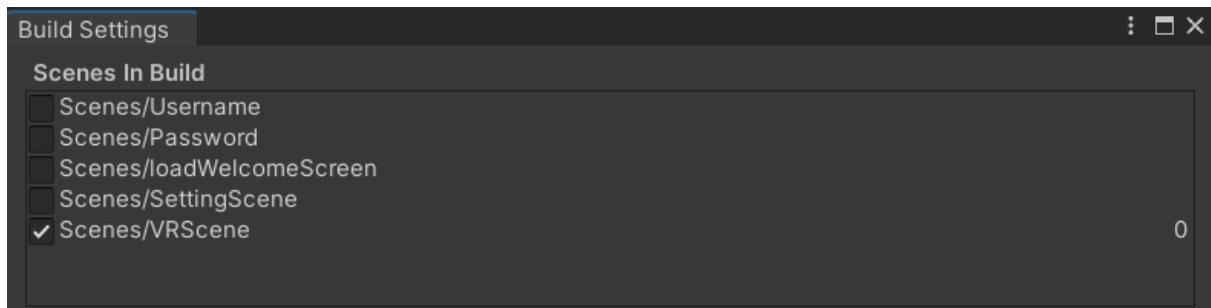
2.1.3 Loading software on to VR Headset

There are now a few steps required to load the system into the Oculus Quest 2 Headset.

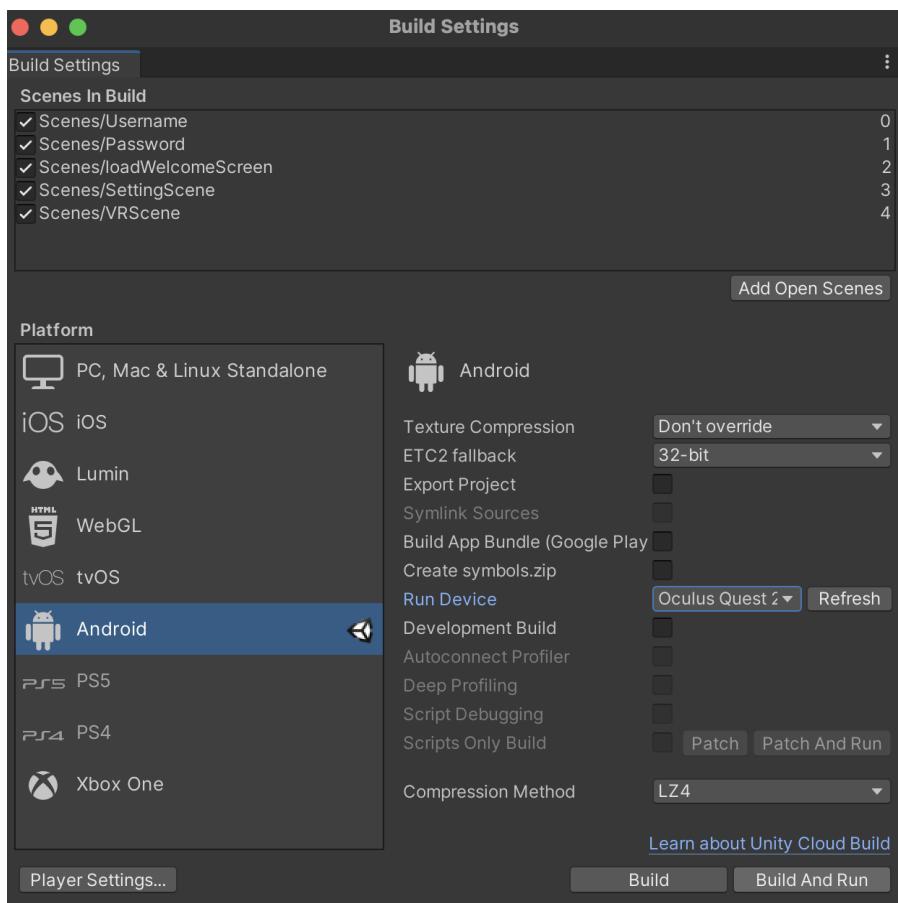
8. Check the order of the scenes in building settings.
9. Unity -> File -> Build Settings...



- Check if the scenes have this order. If not, drag and drop the scenes into the following position:

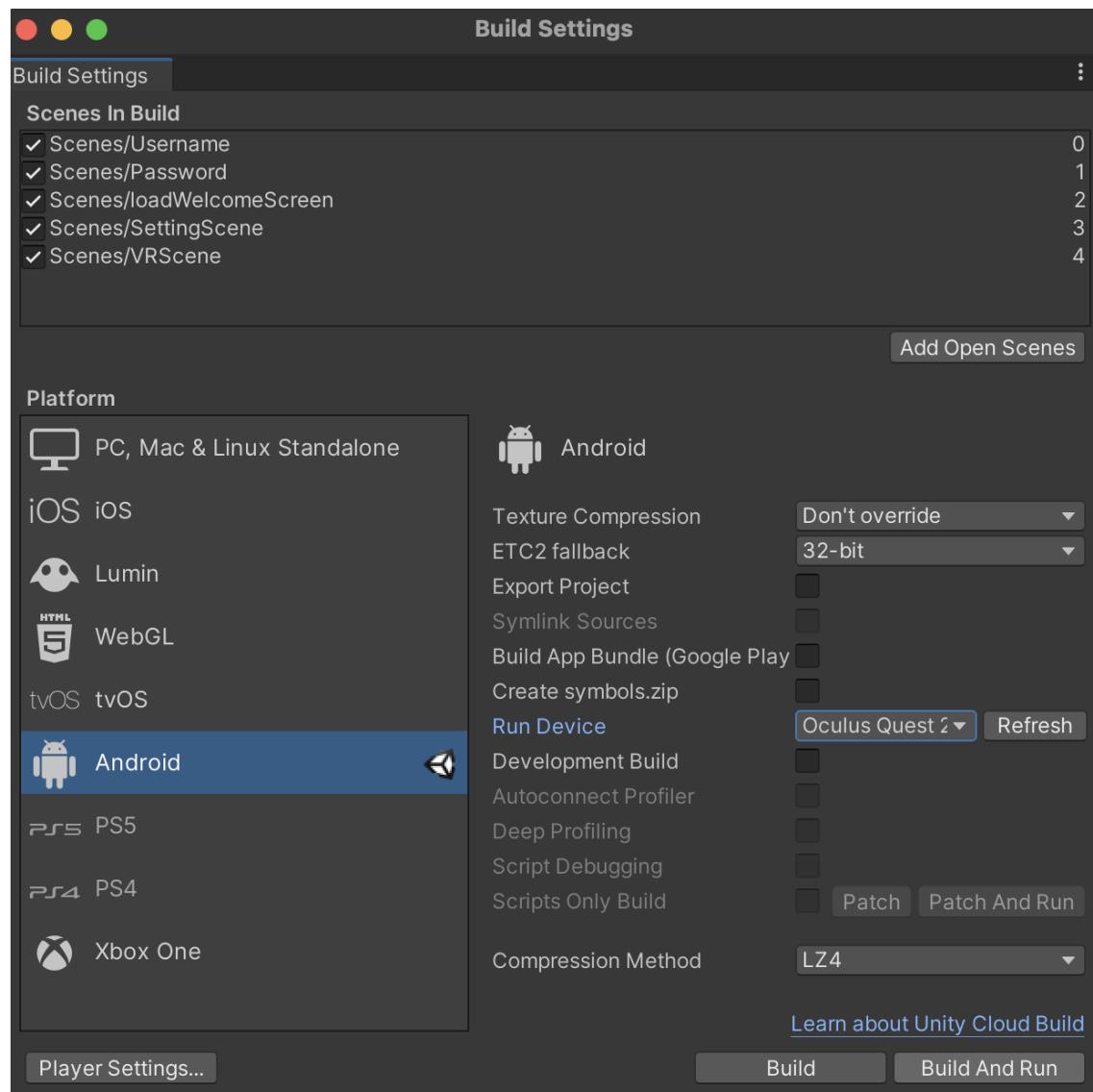


-We were unable to get the menu and log in system working within the VR environment, however if you select the remaining scenes in this order, the system will run within Unity.



10. Firstly, connect your machine to the Oculus Quest 2 headset using a USB-C cable.

Follow the next steps to allow data access to Oculus Quest 2:

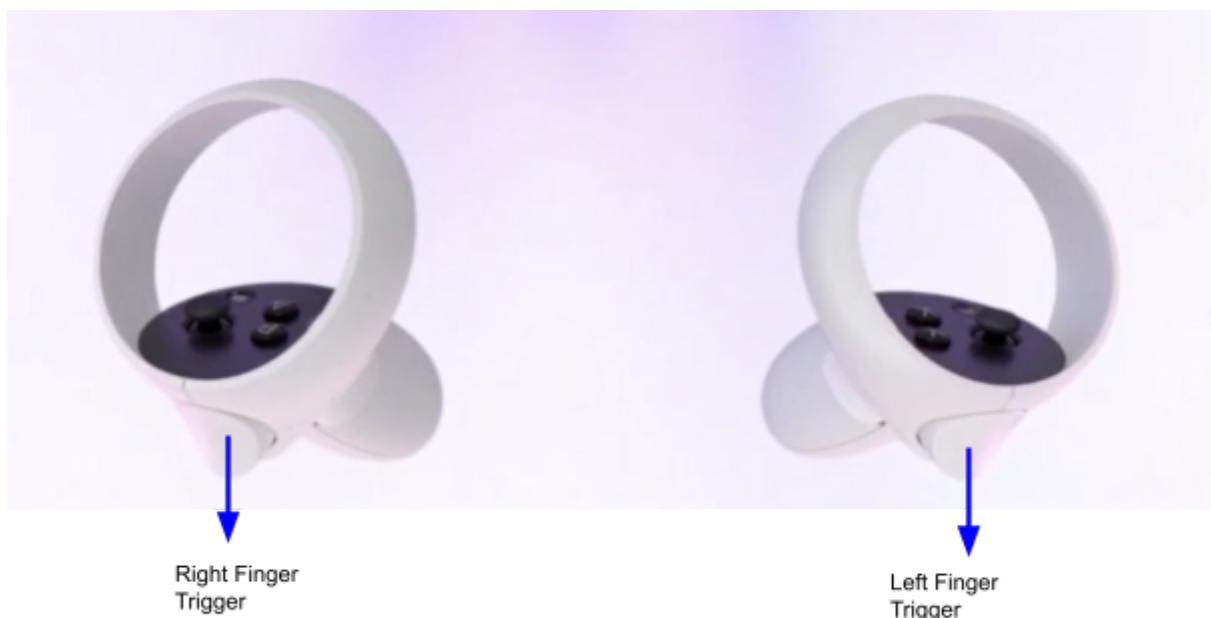


Go to Unity -> File -> Building Setting -> Select Android -> Run Device Select Oculus Quest2 -> Build and Run

11. After this, you should be able to put the Oculus Quest 2 headset on, and the software should be loaded - you will be within the virtual environment.

2.2 Hardware instructions

2.2.1 How to use the controller



- Movement: move left joy stick around to move forward and backward.

- Turn around: move right joy stick around to turn left or right.
- Grab objects: press and hold hand trigger to grab objects, release to drop the objects.
- Click button: press Right Finger Trigger or Left Finger Trigger to click the button.
- Call menu function: press left menu and right menu at the same time to call menu.
- Reset position: press left menu and right menu at the same time and hold 3 seconds to reset the standing position.

3 Using the software

3.1 Conformance to requirement specifications

This project was commissioned by John McNamara, university programs lead at IBM; we were provided with a project brief, with descriptions of the desired software as shown below:

“Create a VR tool which will support users engaging in the IBM Design Thinking Process. The D.T process has a number of elements to work through. Use VR to lead the user through the stages of D.T and then allow the users to capture what has been designed. Use Watson Assistant, Text to Speech and Speech to Text to allow the environment to lead the user through the steps and the user to interact more naturally with the environment.”

From this brief, a document containing functional and nonfunctional requirements was created, containing functions we deemed appropriate to make the requested software.

Of those requirements, these are the functions we determined to be the most vital for the software, as they conform to the requirements requested by the client.

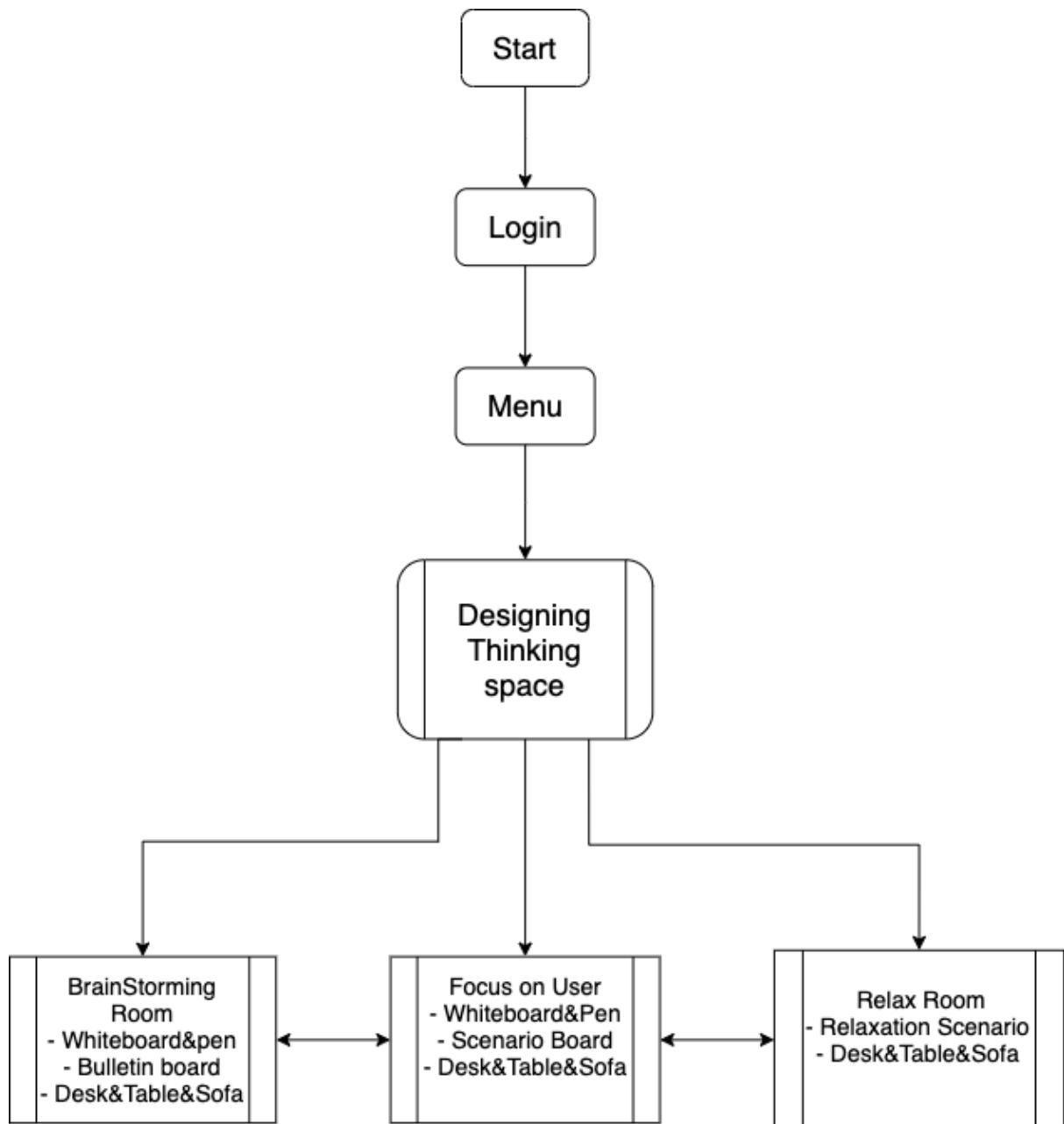
TASK	DELIVERABLE
FR1 – Interface (Lucas)	
1.1	Menu
1.2	Settings
1.3	Multi-user functionality
FR2 – Account (Lucas)	
2.1	Log in
2.2	Profile
FR3 – DT Environment (Vincent)	
3.1	Room generation
3.2	Movement & Looking around
3.3	Interacting with virtual objects
3.4	Adding/deleting objects
3.5	Environment customisation
3.6	Load/play stored videos
3.7	Navigation within rooms
FR4 – IBM AI (Jake)	
4.1	Watson Assistant
4.2	Watson Speech-to-Text
FR5 – Data (Jessie)	

We will reference the functional requirements listed here throughout the manual, when they are mentioned in the software.

We were able to complete a large portion of the most important requirements (as shown throughout the user journey), however there are also some which we are currently working on (the database for importing and exporting notes created within the software), and others that we are leaving for future work, such as the Watson Assistant, multi user functionality, and a custom profile which can be used to customise the account of the current user.

We can now demonstrate how we have completed these functional requirements through the project user flow and overview of each component, alongside the user journeys that show how the client can use the software.

3.2 General user flow within the software



3.3 Software Features

Navigation and Movement (FR3.2 Movement and Looking around)

1. Movement within the virtual world is a combination of two movements. To move forwards and backwards, use the left joystick. To turn and face left or right, use the right joystick. You can move between rooms by walking through the doors.
2. Click the left menu and right menu at the same time to exit the program.

Design Thinking Rooms (FR 3.1 Room generation)

There are currently two working rooms that we have in place, alongside a room for resting, to simulate a working environment.

3. The first room is the BrainStorming Room: This room contains a pen, a whiteboard, a table, a desk, and a sofa. Within this room, the user may pick up a pen and write on the whiteboard. There are several instructions in place to assist the player in following the procedures as described in the IBM Design Thinking Process. The purpose of this area is to allow users to collect further ideas and collaborate with colleagues - the idea is to brainstorm ideas using the sticky notes and the whiteboard.



4. The second room is designed to guide you through the “user empathy” stage - here you answer questions designed to put you in the point of view of the user of the

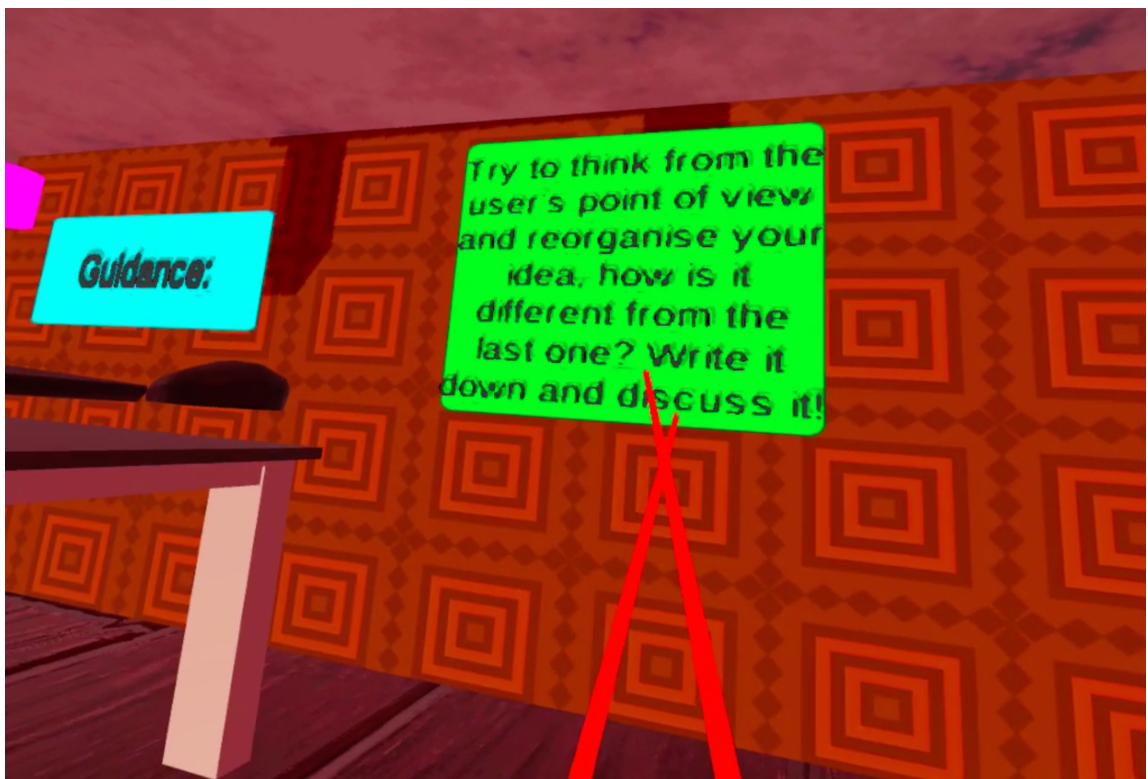
product you are trying to create, allowing you to design solutions for problems they may face : A pen, a whiteboard, a table, and a computer monitor are included in this room. The user may pick up a pen and write on the whiteboard. There are several frameworks in place to assist the player in following the procedure. The purpose of this room is to allow users to rethink their ideas, putting focus on the user's needs.



5. Relaxation room: this room is added to provide a fun alternative environment that is not part of the Design Thinking Process to relax in..

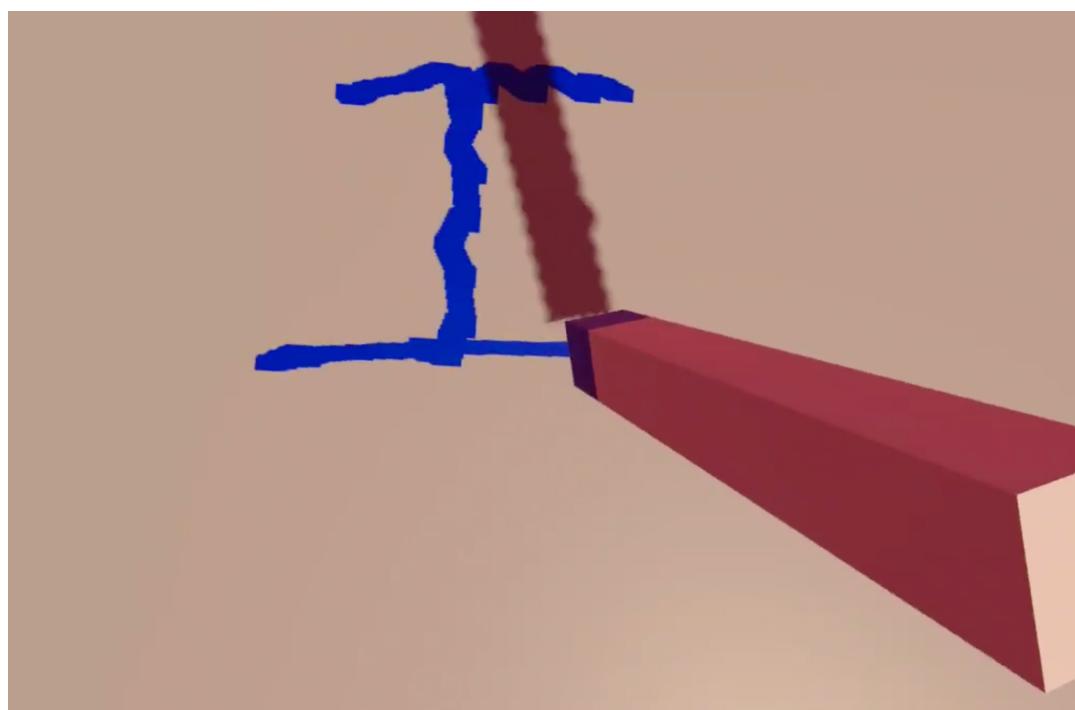


6. If you need help with finding a room, there are signs outside each room. If you need a reminder regarding a specific design thinking stage, or some guidance on how to use the room, check the instructions on the wall to guide you through the process.



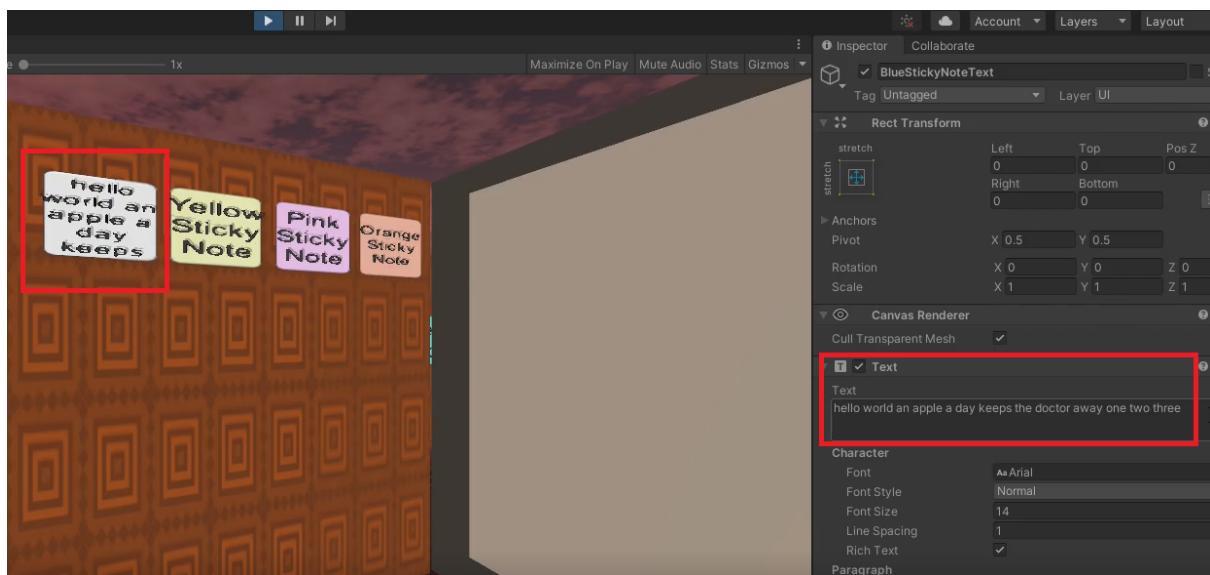
Pen & Whiteboard

7. Grab the pen by using (right hand trigger), and write on the whiteboard as normal while keeping the trigger held down.



Speech-to-Text (FR4.2 Watson Speech-to-Text)

8. In the BrainStorming Room, there are sticky notes on the wall next to the whiteboard. By clicking on the blue sticky note, you can enable Speech-to-Text transcription. Click it again to store and display the transcript onto the sticky note. (Note that this functionality currently only works within the Unity Editor and not when using the VR Headset. It is also only limited to the blue sticky note, and can only be edited once. More information on these issues and limitations can be found in [Section 4.4.1](#))



4 Information for Developers

We will now introduce things of note that developers may be interested in when taking over the responsibility for this software, including current issues, future changes they want to make, and ways to maintain the system.

4.1 Unity maintenance/modification

4.1.1 Upgrading the version of unity used for this software

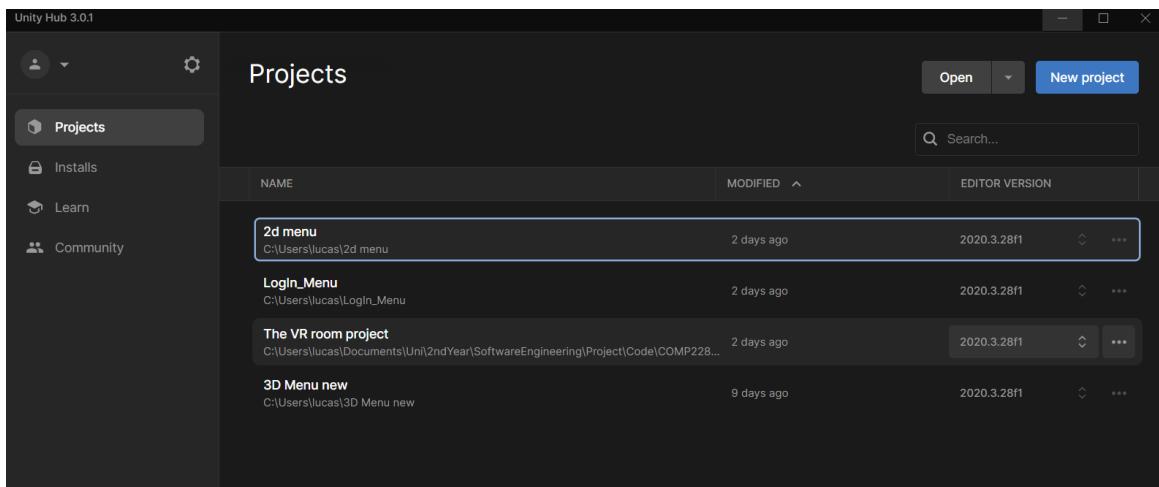
Current version of Unity used is 2020.3.22f1

Preparing the project folder for an upgrade

1. Ensure the current projects runs, and there are no compilation errors
2. Ensure you know all of the packages that are used in the project (can be done by looking within the package manager). At the time of writing this, the project relies on **Oculus XR Plugin, XR interaction Toolkit, and XR Plugin Management**.
3. Ensure the project has been backed up.

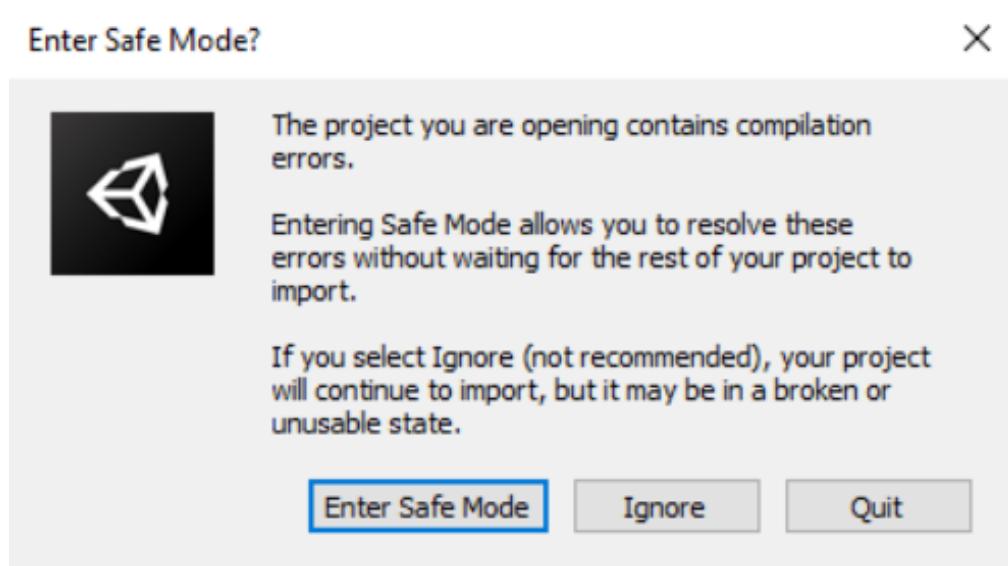
Checking for new updates, and upgrading to a new Unity version

1. When opening Unity or Unity Hub, there should be a pop up that informs you of any updates when they become available.
2. Using Unity Hub, download the new version of Unity.
3. Change the Unity Version against the project to the new downloaded version.
4. Open the project using Unity Hub.



Fixing Errors within Safe Mode

1. When opening the project, there may be some compilation errors. If so, you may be prompted to enter safe mode.



2. Check the console for any errors, fix anything related to the code.

Packages and Assets

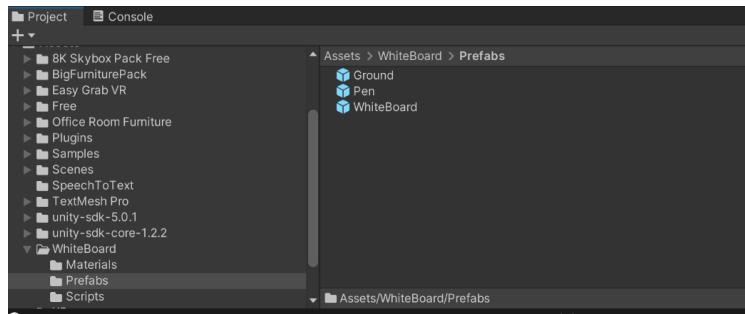
Check that there are no issues with Assets and Packages that worked in the older version of unity that may not be compatible with the newer version. Some pop ups may appear to help you deal with any of these issues.

4.1.2 Functions within unity

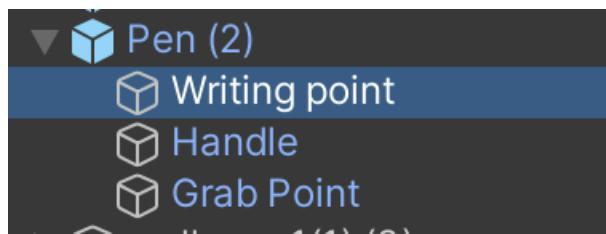
In this section, we will detail the main interactive functions that we have implemented within Unity, should you need to make any updates or changes to it. We will provide a short description of what the feature does, how it works within Unity and how we have implemented it.

1.) Pen:

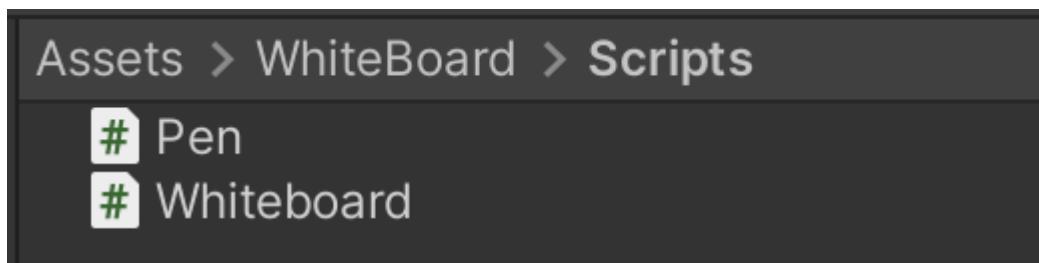
- How to find this function: Go to Whiteboard -> Prefab -> pen



- How to use this function: You can simply drag it into the scene and use it.
- How to change features: double click 'pen', and there are three properties that you can change:



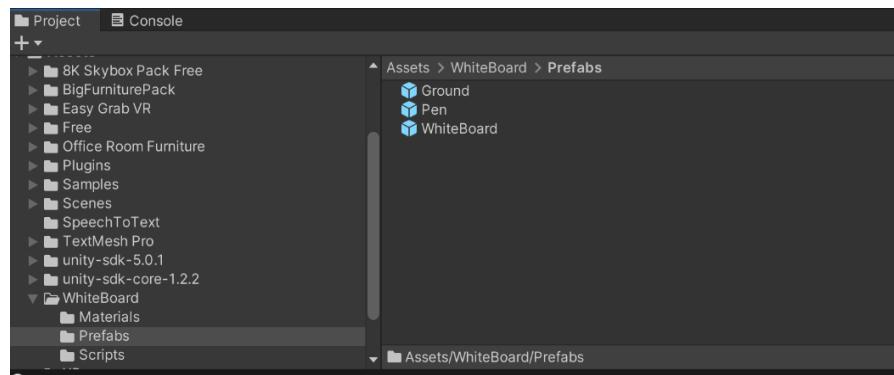
- Writing point: the area which is used to write things down.
- Handle: the place where the user can handle the pen
- Grab point: place where the VR interaction points to. It is a part of the handle.
- Pen: you are able to change the size of the pen and the physics effects when using it.
- where is the script of the pen: Its in assets -> Whiteboard -> Scripts



- What's the usage of this script: This script enables the pen to write on the whiteboard, and there is only one main function called Draw - this function is to make sure the pen can only write on the whiteboard.

2.) Whiteboard:

- How to find this function: Go to Whiteboard -> Prefab -> whiteboard



- How to use this function: You can simply drag it into the scene and use it.
- Where is the script of the whiteboard: Its in assets -> Whiteboard -> Scripts
- What's the usage of this script: This script enables the whiteboard to be an interface that can be written on - without this script, it is not able to be written on.

4.1.3 Updates for Assets and Packages

Assets

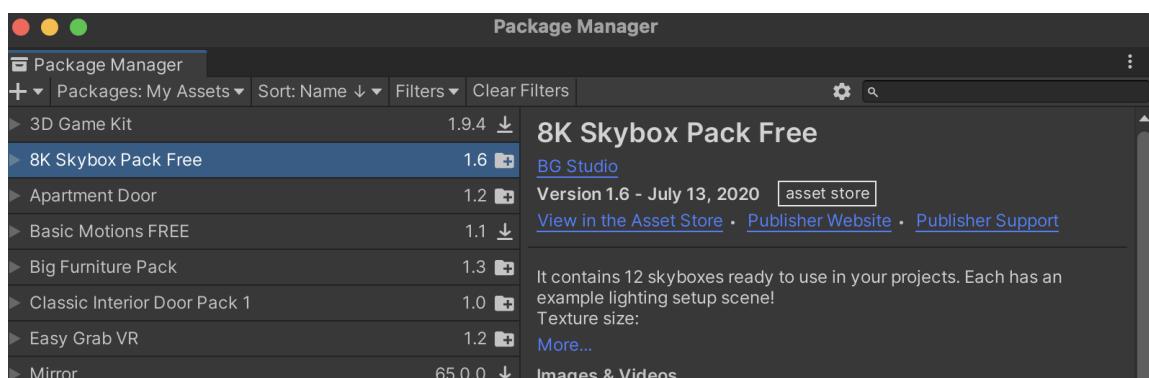
If you choose to modify/update the assets that we have used within the software, below is the link to where they were obtained, along with a list of the assets used at the time of writing.

<https://assetstore.unity.com/>

List of assets:

- 8K Skybox Pack Free
- Big Furniture Pack
- Easy Grab VR
- Office Room Furniture
- Wooden Floor Materials

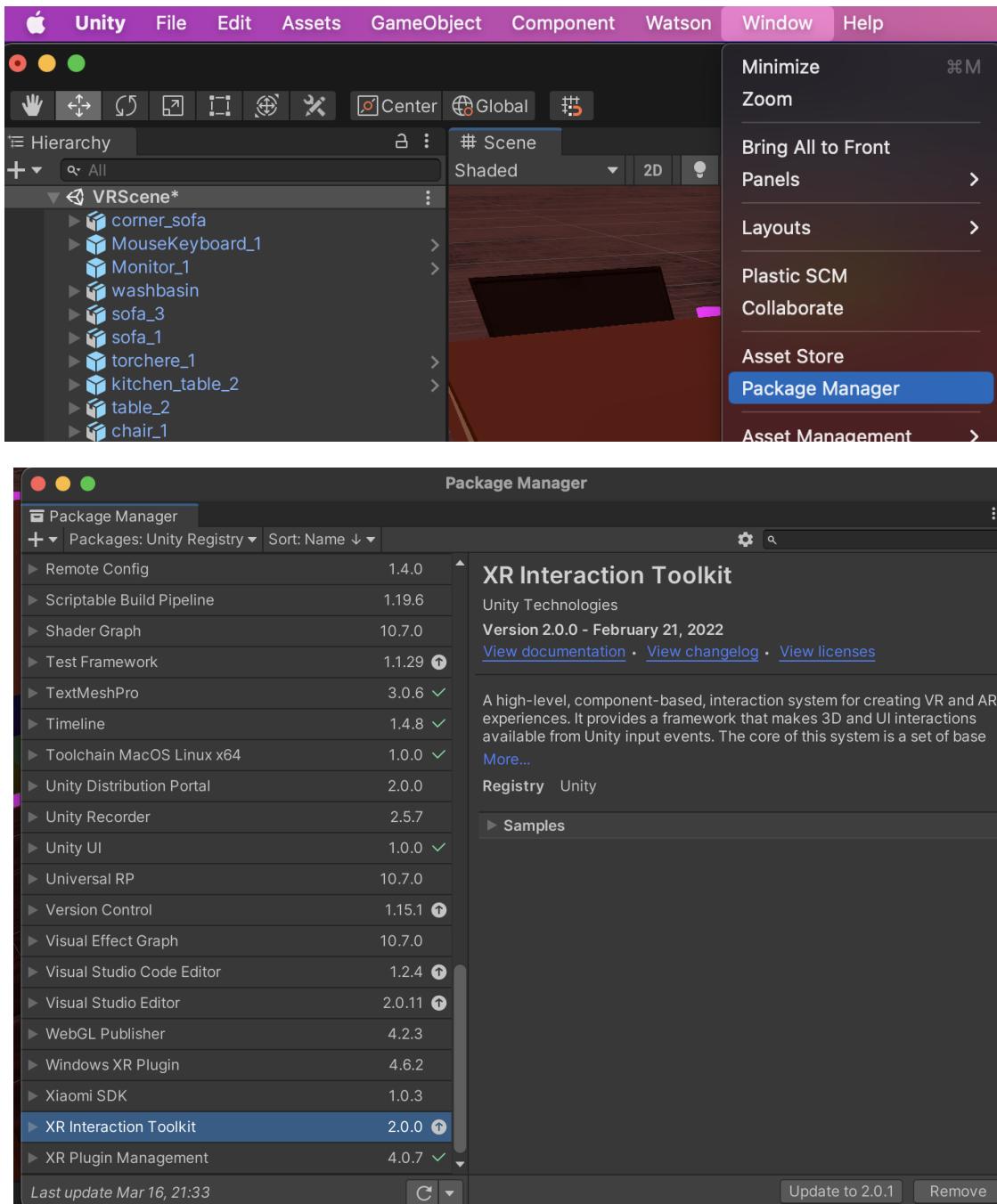
If you need to redownload any assets, due to changes you may have made, go to unity -> window -> package manager -> my assets -> click the asset and import it.



Packages

Should you need to redownload and import the appropriate packages, this can be done by going to (unity -> window -> package manager -> Unity Registry -> click the find the packages and import it.) You will need to import the following packages:

- Oculus XR Plugin
- XR Interaction Toolkit
- XR Plugin Management



4.2 IBM Watson

What is Watson and why do we use it?

As per the request of the client, this software uses the IBM Watson Speech-to-Text service. This is a service powered by IBM Cloud that uses Artificial Intelligence to transcribe speech into text. Within our software, users can click on virtual sticky notes within the rooms and transcribe their speech into text, which is stored onto the sticky notes. In terms of design choice, this is to enable more natural and convenient recording of ideas and designs within the software, as opposed to writing or typing within VR. Furthermore, some users might have inconvenience with excessive hand movements so speech might be a more user-friendly alternative. Other uses for IBM Watson such as Watson Assistant and Speech-to-Text will be discussed in [Section 4.5.2.](#)

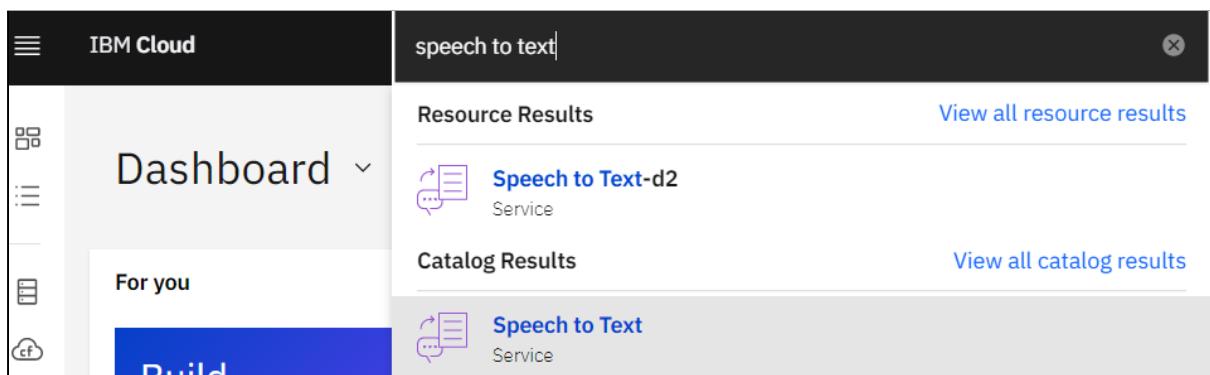
4.2.1 Requirements to use Watson

IBM Account

As Watson is powered by IBM Cloud, a valid IBM account is required. (If you do not own one, you can sign up for a free account with limited privileges at <https://www.ibm.com/cloud/free>. More features can be unlocked by upgrading to paid tiers or with feature codes obtained from academic initiatives and other programs). For the purposes of accessing Speech-to-Text, a Lite account is sufficient.

Setting-up Speech-to-Text & Obtaining Credentials

If you do not have a Speech-to-Text service set-up, here's how you may set one up. First, in the search bar at the top of the IBM Cloud homepage, search for "Speech to Text". Select the option that appears under "Catalog Results".



Next, choose the desired plan and create the service.

The screenshot shows the IBM Cloud Catalog interface for creating a new service. The top navigation bar includes 'Catalog /' and the service name 'Speech to Text'. Below the service name, it says 'Low-latency, streaming transcription'. There are two tabs: 'Create' (selected) and 'About'. A dropdown menu for 'Select a location' shows 'London (eu-gb)' as the chosen option. Another dropdown for 'Select a pricing plan' shows 'Lite' as the selected plan, which includes '500 Minutes per Month' and is 'Free'. A note states: 'The Lite plan gets you started with 500 minutes per month at no cost. When you upgrade to a paid plan, you will get access to Customization capabilities.' Below this, there's a note: 'Lite plan services are deleted after 30 days of inactivity.' To the right, a summary panel displays the service details: 'Speech to Text', 'Location: London', 'Plan: Lite', 'Service name: Speech to Text-9j', and 'Resource group: Default'. It also indicates the service is 'Free'. A checkbox for accepting license agreements is checked, and a 'Create' button is highlighted in blue. A 'Terms' link is also present.

Your Speech-to-Text service has now been created. To view it, return to the home page/dashboard. In the “Resource Summary” section, click on “Services and Software” and select the Speech-to-Text service.

The screenshot shows the IBM Cloud Dashboard. On the left, a sidebar lists various service icons: Build, Cloud, Data, Functions, Kubernetes, Metrics, Monitoring, Network, Object Storage, and VM. The main area features a 'Dashboard' header with a dropdown arrow. Below it, a 'For you' section contains three cards: 'Build' (Explore IBM Cloud with this selection of easy starter tutorials and services), 'Monitor your resources' (Get visibility into the performance and health of your resources), and 'An essential guide to Kubernetes' (Deploy, scale, and manage your containerized applications with Kubernetes). At the bottom of the dashboard, the 'Resource summary' section shows '1 Resources' and 'Services and software' with a count of '1'. A 'View all' link is also present.

^ Services and software (1)				
 Speech to Text-d2	Default	London	Speech to Text	<input checked="" type="checkbox"/> Active

On this page, you will find the credentials required to access Speech-to-Text within Unity. Instructions to using these credentials are provided in Steps 4 to 7 in the Setting Up guide in Section XXXX.

4.2.2 Unity settings and prerequisites

Watson SDK

Watson has a Software Developer Kit (SDK) for Unity. This can be found at <https://github.com/watson-developer-cloud/unity-sdk>. This repository contains useful information on using Watson within Unity.

For our software, we downloaded two sets of Assets of the SDK from these repositories:

<https://github.com/watson-developer-cloud/unity-sdk/releases/tag/v5.0.1>

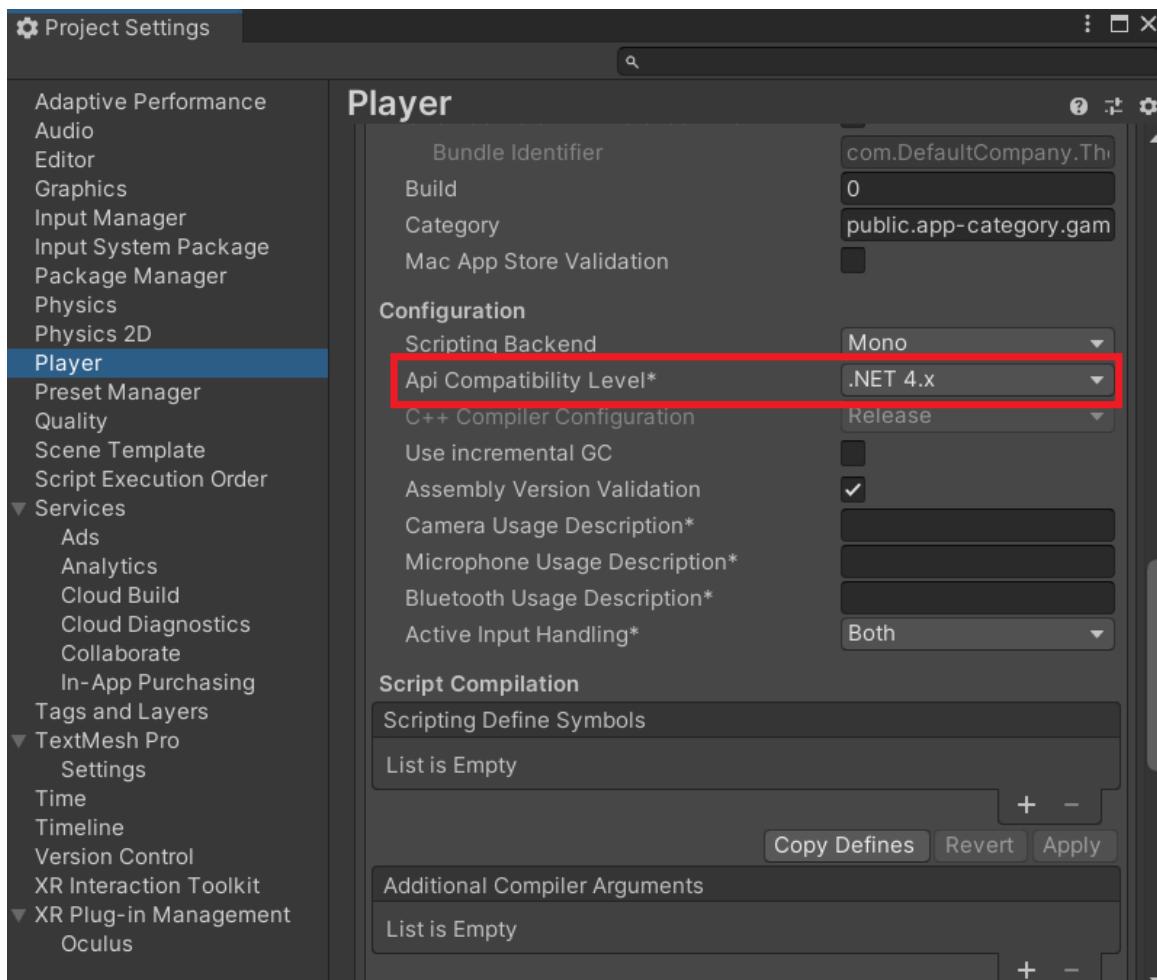
<https://github.com/IBM/unity-sdk-core/releases/tag/v1.2.2>



From each repository, both files were downloaded, extracted and its contents were copied to the “Assets” folder within the Unity project file (“IBM_Design_Thinking_Group24_Project”). The zip folders contain the scripts that provide Watson services. If you are downloading them to replace the folders that we have already included in the software, do note that the original Unity library and the downloaded SDKs both have a copy of “NewtonSoft.dll”. It needs to be removed from the downloaded files or there will be a compilation error due to both having the same .dll file with the same name.

API Compatibility Level

For Watson SDK to work, the API Compatibility Level in the Unity Editor was set to “.NET 4.x”. Should you need to change this option, this setting can be found via Edit -> Project Settings -> Player, scroll down to the “Other settings” tab and expand it.



How does Watson Speech-to-Text work?

Speech is detected from a user's microphone and is sent to IBM Watson. Watson uses Artificial Intelligence and Natural Language Processing to attempt to decipher what is being said and comes up with a list of alternatives. The most likely alternative (or more, if desired) is then picked and sent back to the software integrated with Speech-to-Text. This text can then be used to perform other actions if necessary, like voice commands.

API call

Within Unity, this technology is leveraged by creating a script that makes API calls to Watson to use the Speech-to-Text service. This script is called “SpeechToTextScript” and is located

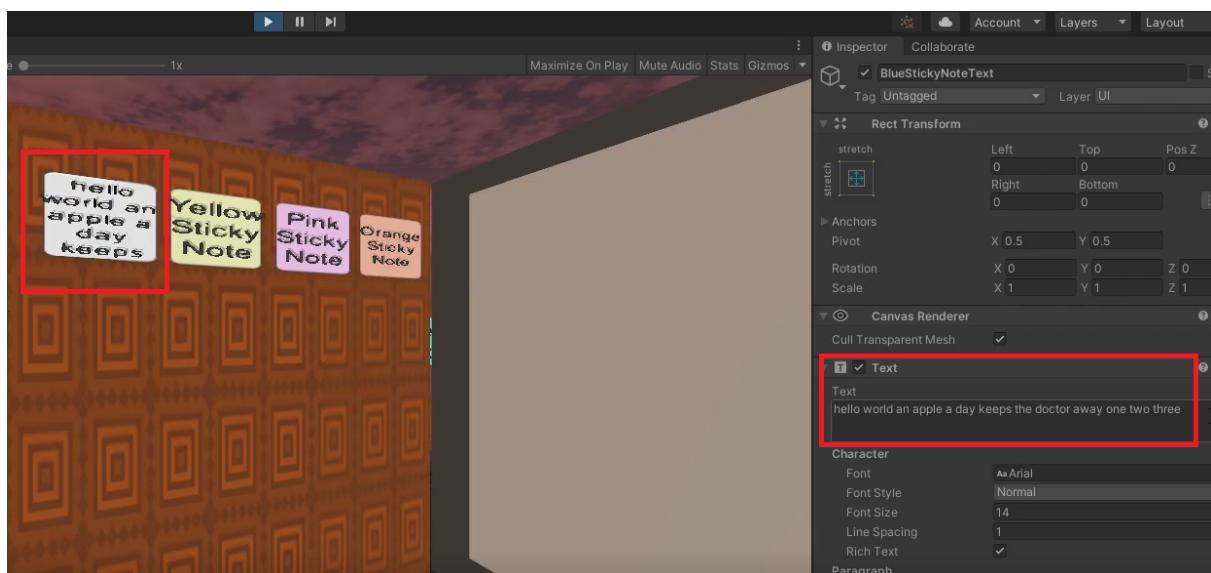
within the Assets -> SpeechToText folder, and is attached to the “SpeechToTextObject” object in Unity in the “VRScene.unity” scene. It relies on the Watson SDK and stores variables that can alter the characteristics of the Speech-to-Text service, such as the maximum number of alternatives returned (MaxThreshold).

```
_service.RecognizeModel = (string.IsNullOrEmpty(_recognizeModel) ? "en-US_BroadbandModel" : _recognizeModel);
_service.DetectSilence = true;
_service.EnableWordConfidence = true;
_service.SilenceThreshold = 0.01f;
_service.MaxAlternatives = 1;
_service.OnError = OnError;
_service.InactivityTimeout = -1;
_service.WordAlternativesThreshold = null;
_service.EndOfPhraseSilenceTime = null;
_service.StartListening(OnRecognize, OnRecognizeSpeaker);
// _service.Keywords = _keywordsSetHere;
// _service.KeywordsThreshold = 0.01f;
```

Should you wish to know more or make changes, a list of all the variables can be found in the “SpeechToTextServiceExtension” script within the “unity-sdk-5.0.1” folder.

Transcribing speech on to an object

While “SpeechToText.cs” provides the functionality, there is another script called “TextChangeScript.cs” that is attached to a sticky note object (VRScene>StickyNoteWall>StickyNoteCanvas>BlueStickyNote) in our virtual room. With the script, upon clicking the sticky note, it creates an instance of the Speech-to-Text service. Upon clicking it again, it retrieves a transcript of the recorded speech which replaces the contents of the sticky note’s text box.



4.3 Database maintenance

4.3.1 Access to the cluster

Firstly, a MongoDB account is required to be created to review the illustration of data flow. Thus, an invitation can be sent to your email address asking you to join the project. A new user can be created to use the cluster under the project. Simply replace the <username> and <password> with your username and password in the header of app.js.

```
mongodb+srv://<username>:<password>@cluster0.wagkr.mongodb.net/myFirstDatabase  
?retryWrites=true&w=majority
```

4.3.2 Installing MongoDB shell and Node.js

To apply the database interaction file, it is essential that MongoDB and Node.js are installed. The suggested version of MongoDB is v5.0.6 or newer and v14.15.4 or newer for Node.js.

There are multiple ways to download the packages. Node.js has some pre-built installers which can be found on <https://nodejs.org/en/download/>. There is another suggested way using homebrew especially for macOS and Linux users via terminal.

To download the homebrew, copy the below code in the terminal and run it.

```
/bin/bash           -c           "$(curl           -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.  
sh)"
```

You can use the command line below to test if the brew is installed.

```
brew --version
```

If the brew is installed, the terminal will display the version of homebrew.

```
[(base) zhangs-mbp:database shinypipibear$ brew --version  
Homebrew 3.4.1  
Homebrew/homebrew-core (git revision 6c3c8a34df5; last commit 2022-03-15)
```

Then go to the repository where the program is located in the terminal. And enter the following code to install the Node.js.

```
brew install node
```

When it's installed, your terminal will show something similar like this.

==> Checking for dependents of upgraded formulae...

==> No broken dependents found!

==> Caveats

==> node

Bash completion has been installed to:

/usr/local/etc/bash_completion.d

For MongoDB, pre-built installers for windows and macOS can be found on https://www.mongodb.com/try/download/community?tck=docs_server&_ga=2.100185818.539045203.1647365739-294434534.1643294266. Choose the MongoDB Community Server and the suitable platform and preferred package for you to download.

MongoDB Community Server

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

Give it a try with a free, highly-available 512 MB cluster.

Available Downloads

Version: 5.0.6 (current)

Platform: macOS

Package: tgz

Download Copy Link

Current releases & packages
Development releases
Archived releases
Changelog
Release Notes

For windows users, further detailed configuration is on <https://docs.mongodb.com/v4.4/tutorial/install-mongodb-on-windows/> to set MongoDB as your windows service. To run the MongoDB service, open the command interpreter and enter the command line below. If it starts running, then it's successfully installed. The service is checking if your computer can connect to the MongoDB server.

```
"C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe"
```

For macOS users, homebrew is suggested to install the MongoDB shell. Go to the program repository then enter the codes below in the terminal.

1.MongoDB homebrew tap for MongoDB and the database tools

```
brew tap mongodb/brew
```

2.Install MongoDB in the terminal

```
brew install mongodb-community@4.4
```

Now, you have the MongoDB shell and Node.js fully installed on your computer and ready to be applied in the program. Enter the following code in the program repository to start the server.

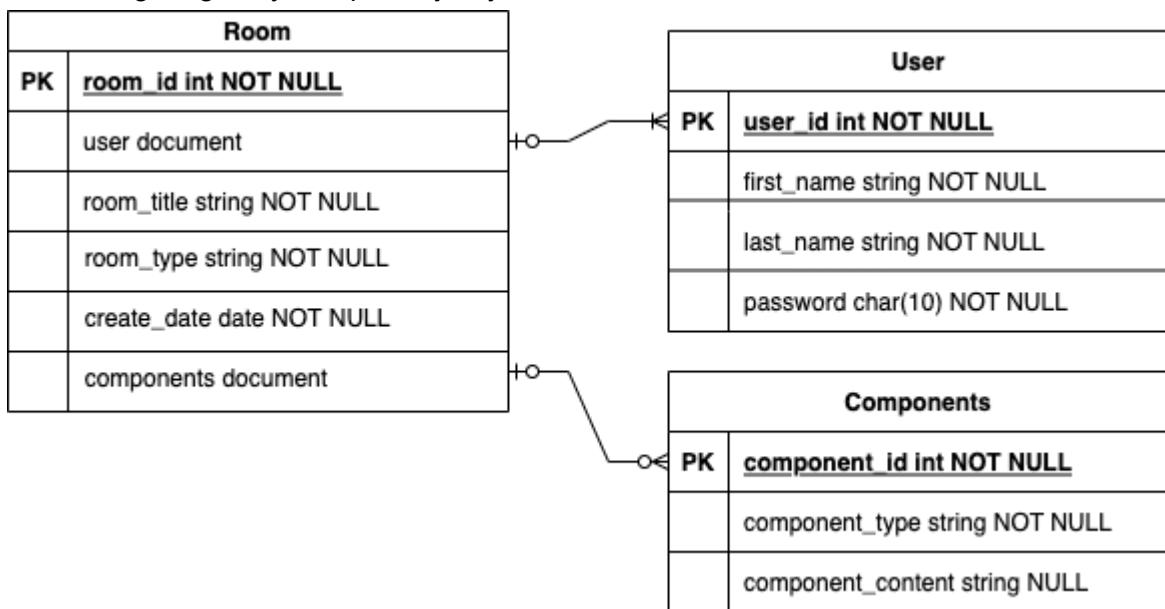
```
node app.js
```

If the server is successfully running, your terminal window will display the following. It should be a blank newline under the code line.

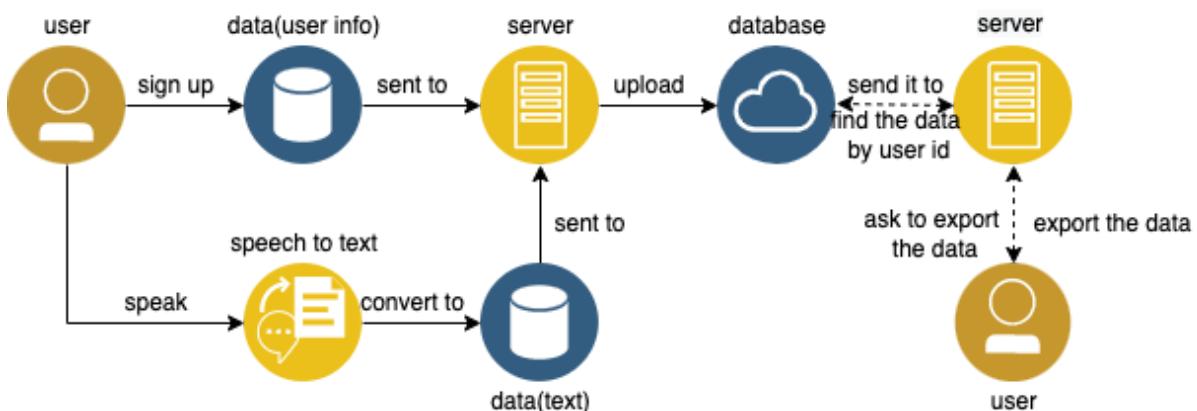
```
[(base) zhangs-mbp:database shinypipibear$ node app.js
```

4.3.3 Database Structure

The diagram below shows the ER model which states the relationships among different data fields using foreign key and primary keys of documents.



The graph below represents the interaction relationships among users, data, server and database.



4.4 Issues with the current program

This section contains issues that have yet to be resolved/ may need some troubleshooting

4.4.1 Table of current issues

Issues	Potential Solutions
Lack of username and password validation	Have conditions within the relevant scripts that ensure that the username and passwords written match a certain set of criteria - such as length of password, types of characters used
Lack of username and password security	Link the database with the rest of the system - store the usernames and passwords in mongodb where they can be encrypted - make api calls to see if the usernames and passwords match those that have already been stored.
Inability to register a new user	Make API calls to the mongodb database, sending the data contained within the username and password fields as json.
Menu and log in do not load when using the VR headset	Look into converting the 2d menu into a 3d menu, as 2d sections within the VR environment can cause issues.
Currently don't connect each users IBM account for use with Watson	This may require users to have a paid account, as currently you have to manually enter the API key to a field within the unity script.
Currently only one of the sticky notes has functionality to display what has been said by the user as text - Watson Speech to Text	Extend the existing function within C# to the empty sticky notes.
Once you have added text to a sticky note, you are unable to re record and edit the text.	
As soon as you enter the environment, Watson starts listening and transcribing everything you say, ideally it would only be triggered when clicking on sticky note	Do some more research into C# and the Watson Speech to Text scripts and packages
The Watson trigger for the sticky note does not work when accessing the software via the VR headset	Use two buttons within the environment to turn Watson on, and then turn it off. Currently there are bugs as it is unable to be triggered properly.

If you ‘push’ too hard when using the pen on the whiteboard, text doesn’t appear	Perhaps provide a tutorial upon entering the room that informs the user of how to interact with the room, reminding them to not ‘press’ too hard.
After holding the pen, if you don’t directly drop it straight down, it will fly away	Make some changes to the physics effects within unity, especially the gravity component.
The scenes within the software may not render properly	Ensure that all the correct assets have been installed, and you are not using a version of Unity lower than 2020.3.22f1

4.4.2 Current issues with the menu and log in section

When a user initially launches the VR application, they will be prompted to sign in using their IBM account details (**currently you can put “user” for the username, and “password” for the password**). In the future, we plan to link IBM accounts and validation to the software through API).

This functionality doesn’t work within the Oculus Quest 2, however when the software is just run within unity, this is how it functions.

1. Input username (client can input “user” as username to access the next section).(FR 2.1 Log in page)

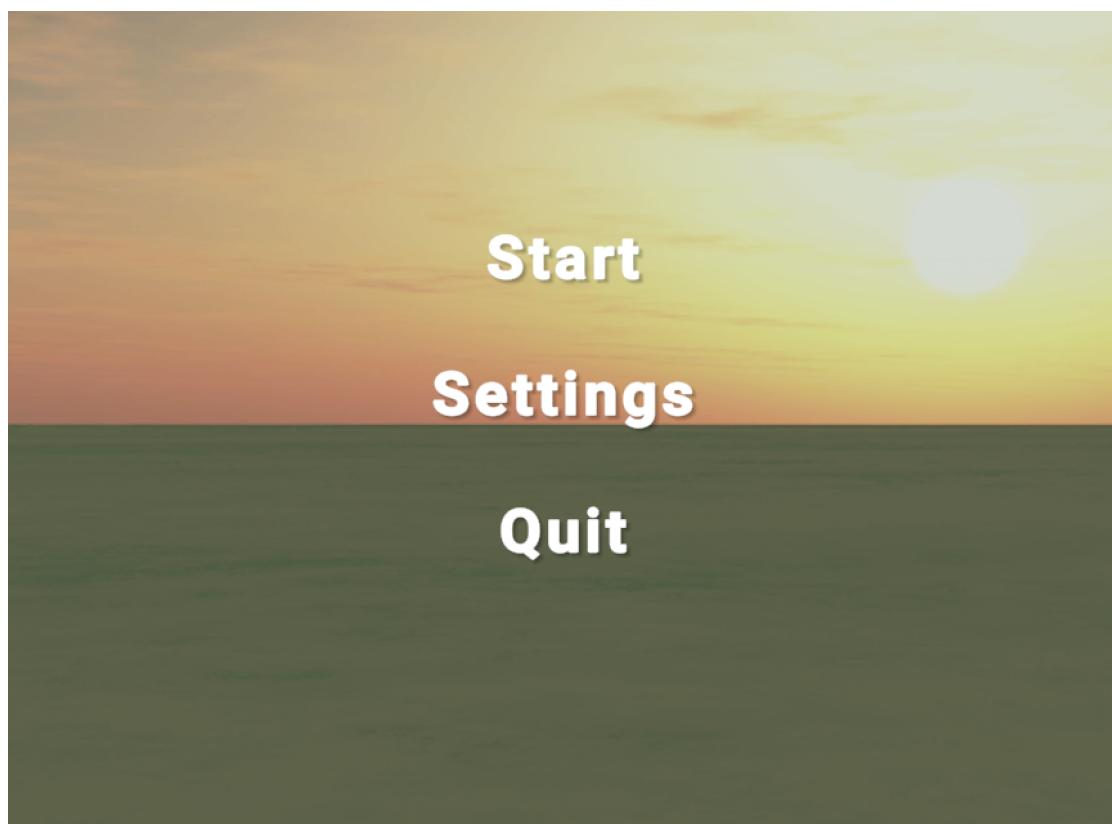


2. Input password (client can input “password” as password to access the main menu).(FR 2.1 Log in page)

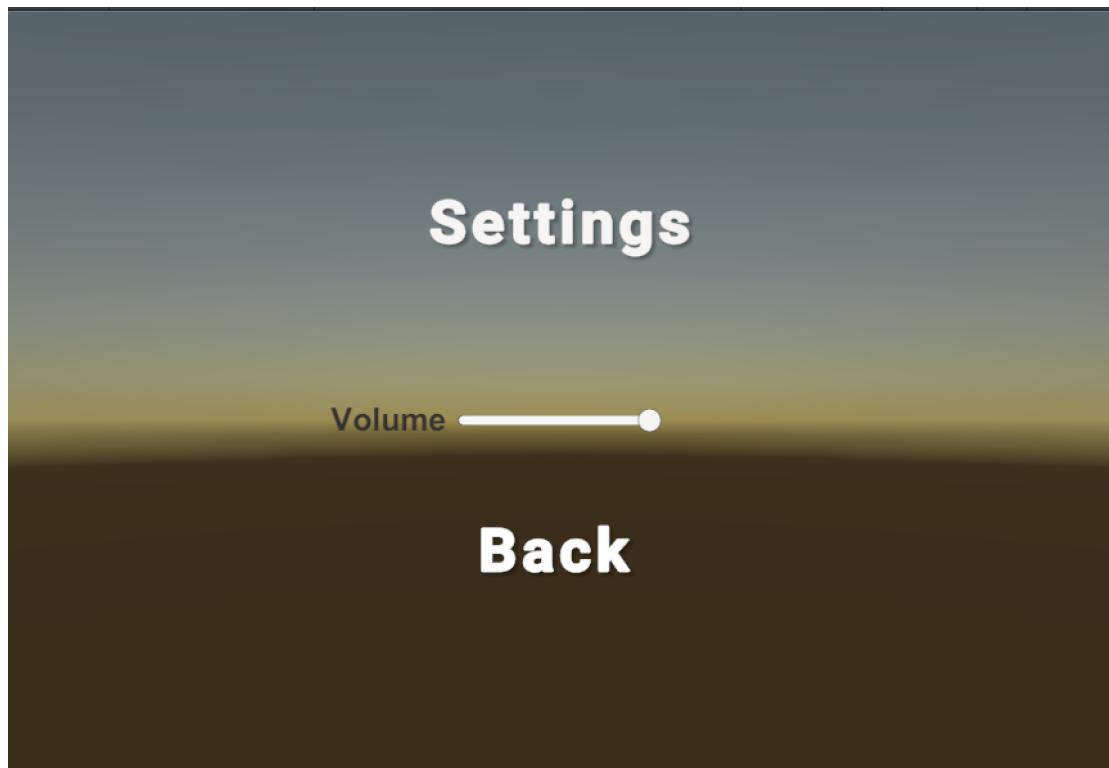


Then, the user will be sent to the menu, where they will find three options:
(FR1.1 Menu)

- Start: loads up the Design Thinking Virtual Environment.
- Settings: allows the user to modify some system settings
- Quit: terminates the application



3. When the user clicks Setting, they are able to adjust the volume of the background music.(FR1.2 settings)



4. When the user clicks Start, they are sent to the Design Thinking environment. The environment is currently split into three separate rooms.

The rest of the functionality is demonstrated within the software features section.

4.5 Future changes and implementations

This section contains functions that we would like to implement in the future

4.5.1 Multi user functionality

As outlined at the beginning of the user manual, there are various functions we would like to implement in the future, that were described as some of the functional requirements. The most important functionality that we would like to add would be the capability for multiple users to interact in the same room at the same time. This would allow for users all over the world to collaborate in real time, using the design thinking processes outlined within our rooms to build solutions together. This would greatly improve the effectiveness of the VR software, as currently, it is only built for individual users. This could perhaps be done using Normcore.io, a framework that can be used within unity to implement multiplayer functionality. Normcore is a networking plug-in and hosting service, which can be used to scale a project for many users. It can also be used to host multiple rooms, so if we were to scale this product, we can host all the rooms separately and have multiple users working together in each room.

4.5.2 Additional Watson services

Another feature we would like to add would be the use of a bot that guides you through the design thinking process, asking specific questions to help keep you engaged with a task, and leads you through the steps required within each room. This would be powered by Watson Assistant and IBM cloud, and would follow you around to keep you on task. We could generate an avatar to have a visual representation of the assistant using Morph 3d, and use the asset IBM Watson SDK for unity to integrate the Watson assistant with Unity. We would implement Watson Speech to Text to communicate with Watson Assistant. Following on from these Watson implementations, we could also use Watson Text to Speech within each room, to output the instructions for users that may have visual impairments.

4.5.3 Database implementation - importing and exporting work

Although we have the database design in place and the ideas in place, we are yet to fully integrate our database with the current software. Doing so would allow us to add another function, that would allow us to import work we have been working on, and make more changes to it within the virtual environment using the whiteboard feature and sticky notes, and watson speech to text. As well as this, when we have finished making notes and brainstorming, we would like to be able to export these notes into the database, so that they are safely stored away; this is much easier than the current brainstorming methods which would require many sticky notes, and can quickly become disorganised. This would be done by creating some communication between the unity and mongodb api's, to allow them to interact, and store the data from unity within the mongodb database, and then import the data from mongodb to unity.

4.5.4 Additional sticky note function

Following on from this, we will also need to improve the functionality of Watson Speech to Text within our system, especially when working with sticky notes. For example, in order to make the most of using sticky notes, we will need to implement a function that allows us to add and delete sticky notes, while preserving the functionality that allows us to use speech to text when working with them.

4.5.5 Customisation

In order to make the experience more engaging for users, implementing a greater number of customisation options would allow each user to make their environment individual and well suited for them. This would include, the ability to change the colour and style of the walls in each room, customise their avatar and change the outer environment such as the sky.