VR Application For Learning Sign Language

Computer Science for Games
Bradley Wilson - 19809461
Karina RE & Jennie H

This project aims to make a virtual reality (VR) application to teach fundamental signs to be used in social VR scenarios.

Github: https://github.com/Spot-Dev-0101/VRSigns

1) Introduction

1.1) Project summary

During the pandemic, I have spent a significant amount of time in social VR spaces, like VRChat. These are amazing environments that allow the users to be anything or anyone they want within the safety and comfort of their own homes. During my time in these spaces, I have met all types of people from around the world. The most common type is a mute, someone that doesn’t speak either by choice or for medical reasons. These people often struggle to communicate because the only form of communication is to use their voice. Recently VR sign language has become very popular and I have observed many people express a desire to learn it, however, there aren’t many resources available so the best way to learn it is from someone that already knows it.

Due to the lack of resources and tools, I will set out to create a VR application that aims to teach the user basic signs. It will support the Valve Index controllers, this is because they have tracking or individual fingers which are required and are very popular. I will also need to develop a gesture recognition system that is able to accurately detect what the user's hands are doing and match it to a database of signs. Additionally, there should be a system to indicate to the user what they are doing wrong when attempting to do a sign. Finally, the experience needs to be engaging to keep the user interested and wanting to learn.

I think this project is important for the future of VR social communications because of the rapid hardware improvements and the constant creation of new virtual environments, such as the multiverse that is being developed by Meta. The more people that know how to speak sign language the better the experience will be for everyone.

2) Methodology

2.1) Development tools

There are three viable options for development tools I could use for this project. Firstly, is creating a world in VRChat. This is a fantastic option because my target audience is the VRChat player base so having the learning environment in the game means less work on the user's end and a more seamless experience. However, the tools for developing a world are very limited, for example, the SDK does not give me access to the user's finger positions so I can not make a real-time sign recognition system.

Secondly, I could use Unreal Engine. Unreal has good support for creating VR applications, however, I'm not very confident with it and I have very little experience using it.

Finally, I have chosen to use Unity. This is because it is incredibly popular and has fantastic support for all VR platforms. Additionally, I already have experience creating 2D and 3D games with it so I can transfer that knowledge. This will allow me to create the best experience possible because I will have full access to all the hardware information and I am not limited by the VRChat SDK.

2.2) Project management

Methodology

One of the methodologies I could use for this project is waterfall. This is good because it keeps a rigid structure that can be followed. However, I will be using agile due to how flexible it is. Agile lets me continuously test and go back and make changes easily, whereas waterfall is more punishing.

Project planning/upkeep

It is incredibly important to keep track of progress so I will be using Trello to have a board of what I need to do, what I'm currently working on, and what has been finished. Doing this will make it easier for me to know what I need to do and keep up my motivation because I can see the list of to-do items shrink over time.

For project upkeep I am going to use GitHub. This will ensure I never lose any work and I can easily go back to an old version if I have any issues.

2.3) Testing/Surveys

Firstly, I will perform my own testing where I compare what happens with what I expect to happen and write notes on how it performed. This lets me easily fix bugs because I will have a record of what happened.

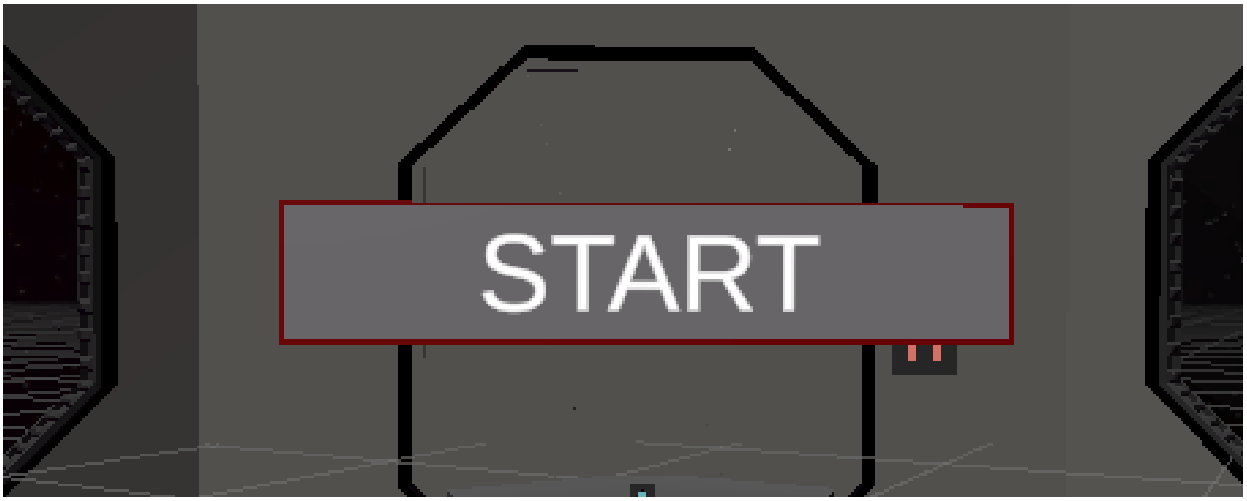
Secondly, I will perform user surveys where I give the application to a variety of people and see how they interact with it. This will be incredibly useful because as the developer I already know how the system works and what to expect so having outside perspectives gives me an insight into how normal users feel about certain aspects.

3) Product description

3.1) Interface

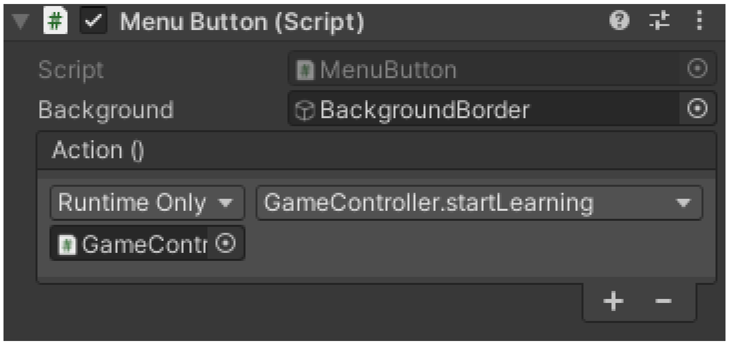
3.1.1) Main Menu

The user interface needs to be as simple and easy to understand as possible so they don’t become confused and want to leave. For this reason, the main menu is simply one button. When it is pressed they are taken to the learning mode.



(Fig 1. Main menu start button)

When implementing the button I wanted it to be as modular as possible so I could make as many buttons as I needed without having to write separate code for every one. Each button has the Menu Button script attached which allows me to select an action for it to perform. This script also implements my Interactable interface which means there is no code needed to tell the interaction system that there is another button.



Menu Button (Script)

Script MenuButton

Background BackgroundBorder

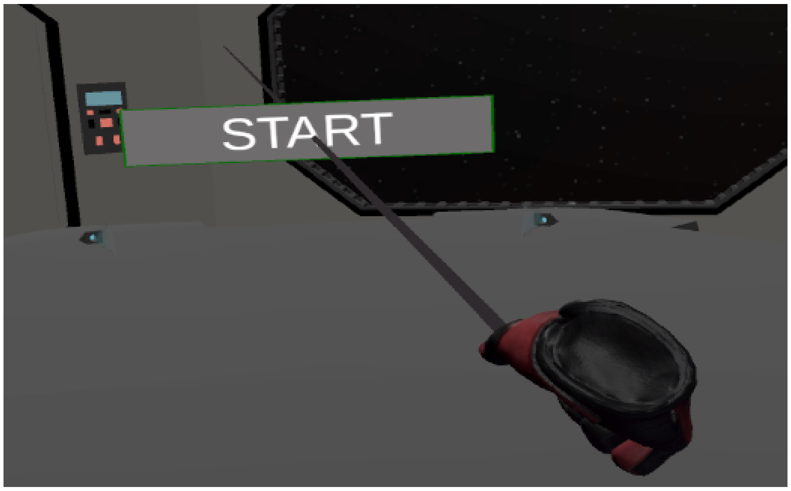
Action ()

Runtime Only GameController.startLearning

GameContr

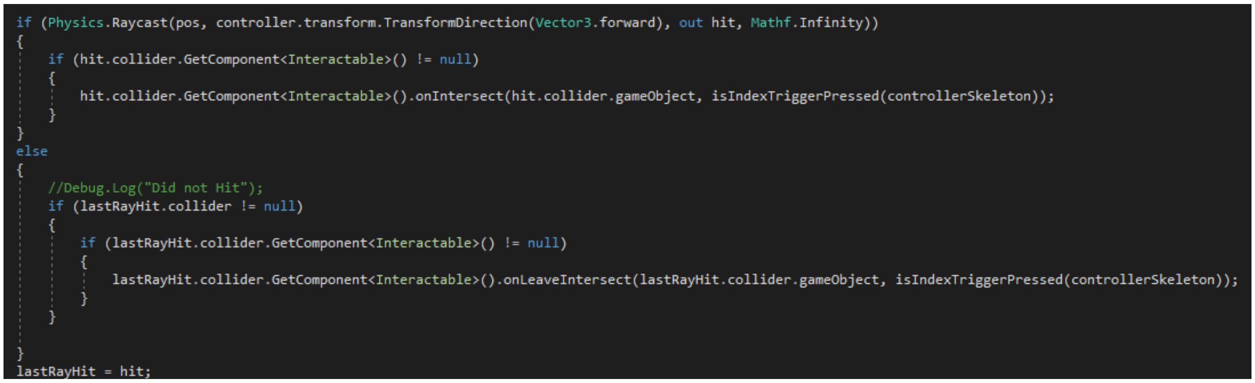
(Fig 2. Menu Button script)

To interact with a button there is a laser pointer that comes from the user's hand. They point it at the button, which will turn green, and pull the trigger to activate it. This is incredibly simple and will not cause any confusion.



(Fig 3. The button is green because it is being intersected by the laser)

The button interaction is handled by, the previously mentioned, Interactable interface. If the laser coming from the user’s hand intersects an object that has the interface it will run a method in that object saying it is being hovered over and if the trigger is being pulled. A method is also run when the object is no longer being pointed at.

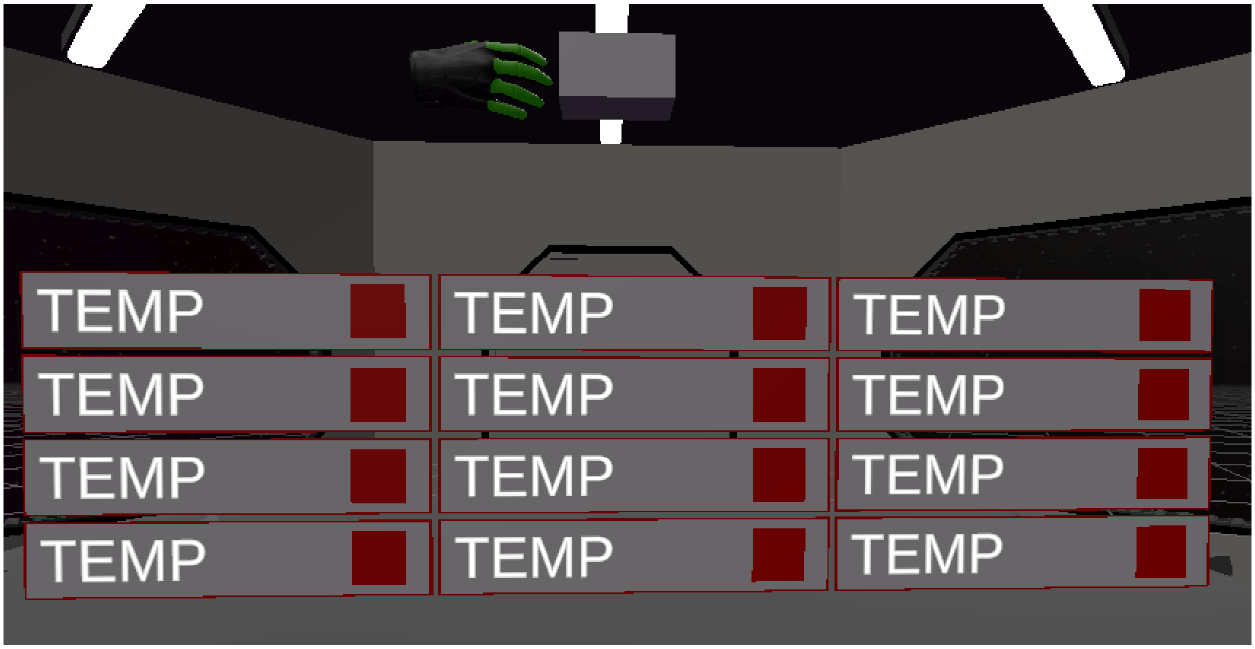


```
if (Physics.Raycast(pos, controller.transform.TransformDirection(Vector3.forward), out hit, Mathf.Infinity))
{
    if (hit.collider.GetComponent<Interactable>() != null)
    {
        hit.collider.GetComponent<Interactable>().onIntersect(hit.collider.gameObject, isIndexTriggerPressed(controllerSkeleton));
    }
}
else
{
    //Debug.Log("Did not Hit");
    if (lastRayHit.collider != null)
    {
        if (lastRayHit.collider.GetComponent<Interactable>() != null)
        {
            lastRayHit.collider.GetComponent<Interactable>().onLeaveIntersect(lastRayHit.collider.gameObject, isIndexTriggerPressed(controllerSkeleton));
        }
    }

}
lastRayHit = hit;
```

(Fig 4. The code for interacting with a button)

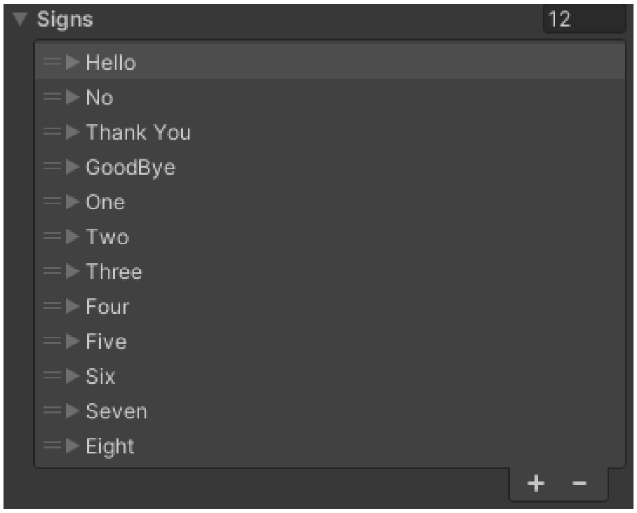
3.1.2) Learning mode interface

The interface while in learning mode is just as simple as the main menu. At the top is the feedback system and an animation of the sign the user needs to perform. Under that is the list of words. The currently selected one is highlighted in yellow, once it is completed it will turn green and the next sign down will be selected.



(Fig 5. Learning mode interface)

Once again it was important to make this as modular as possible so I can add more signs easily. The parent object has the SignList script which contains all the information about the signs needed to run the list. Doing it this way let me simply create a prefab for each item and add it to the list.



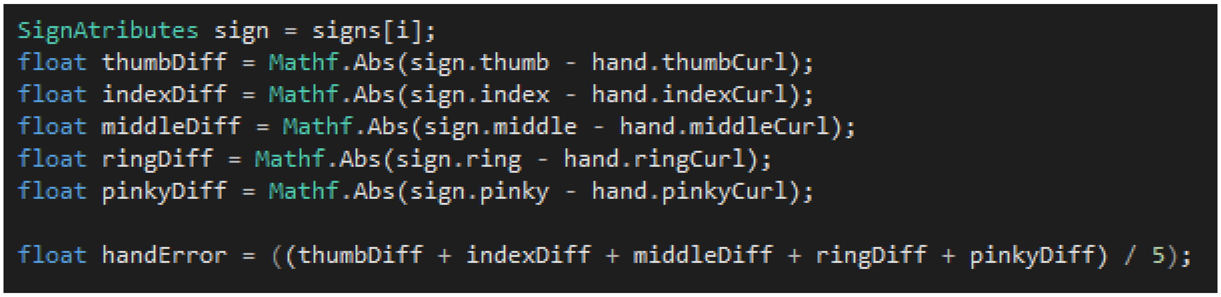
Signs 12

- Hello
- No
- Thank You
- GoodBye
- One
- Two
- Three
- Four
- Five
- Six
- Seven
- Eight

(Fig 6. SignList script)

3.2) Gesture recognition

The gesture recognition system needed to be as simple as possible so it can run quickly and efficiently while accurately detecting which sign the user is performing. To achieve this each finger is given a value between zero and one, zero is fully extended and one is fully retracted. These values are then compared to a list of signs and an error value is given for each one. The sign with the lowest error is then selected.



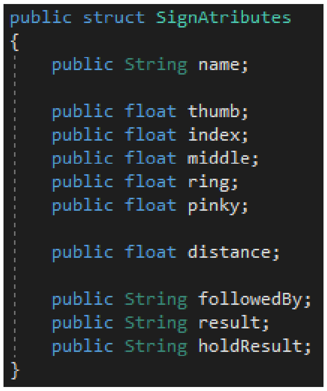
```
SignAtributes sign = signs[i];
float thumbDiff = Mathf.Abs(sign.thumb - hand.thumbCurl);
float indexDiff = Mathf.Abs(sign.index - hand.indexCurl);
float middleDiff = Mathf.Abs(sign.middle - hand.middleCurl);
float ringDiff = Mathf.Abs(sign.ring - hand.ringCurl);
float pinkyDiff = Mathf.Abs(sign.pinky - hand.pinkyCurl);

float handError = ((thumbDiff + indexDiff + middleDiff + ringDiff + pinkyDiff) / 5);
```

(Fig 7. Code that gets the error of the user's fingers for a given sign)

Once the system has a sign it waits for 0.2 seconds and checks if the user is still doing the same sign. This is to make sure the system has the correct sign. The reason to specifically wait 0.2 seconds is that any shorter I encountered issues where it was not accurate enough and any longer I could feel that the system was waiting.

Another important aspect of this system is to allow signs with multiple positions and finger layouts, for example, to say “thank you” place your hand at your chin then swing it down. This is done by having two separate signs. The first is the initial position which when detected is told to wait for the second sign. If, the next sign is the one it’s waiting for the result will be the outputted. However, if the user does not perform another sign the system will return either nothing or another word, for example, if the user performs the first sign of “thank you” and then does nothing it will say “five” because the sign for five is an open hand.

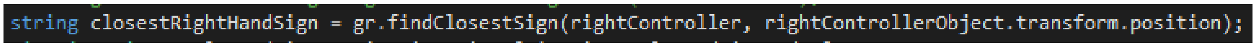
To store the information for each sign the system uses a custom data type called SignAtributes. This contains the sign name, finger positions, the distance from the user’s head, which sign it needs to wait for, and which sign to say if the user holds the position.



```
public struct SignAtributes
{
    public String name;

    public float thumb;
    public float index;
    public float middle;
    public float ring;
    public float pinky;

    public float distance;

    public String followedBy;
    public String result;
    public String holdResult;
}
```

(Fig 8. Custom data type SignAtributes)

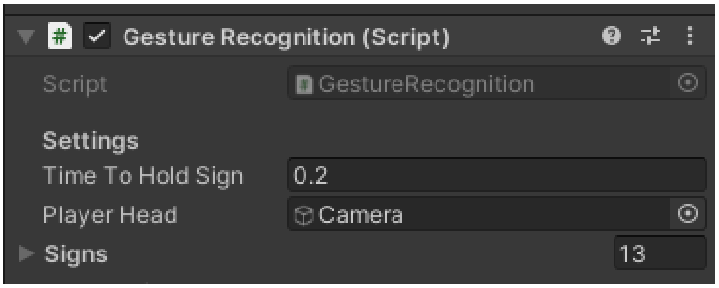
Similarly to everything else in this project, the gesture recognition system is modular and easy to expand. The method that gets the closest sign works with both hands and returns a simple string that can be used anywhere.



```
string closestRightHandSign = gr.findClosestSign(rightController, rightControllerObject.transform.position);
```

(Fig 9. Code that calls a method in the gesture recognition system)

All the settings can be easily changed in the Unity editor and there are not many hardcoded values which makes it easier to rapidly test.



☑ Gesture Recognition (Script)

Script: GestureRecognition

Settings

Time To Hold Sign: 0.2

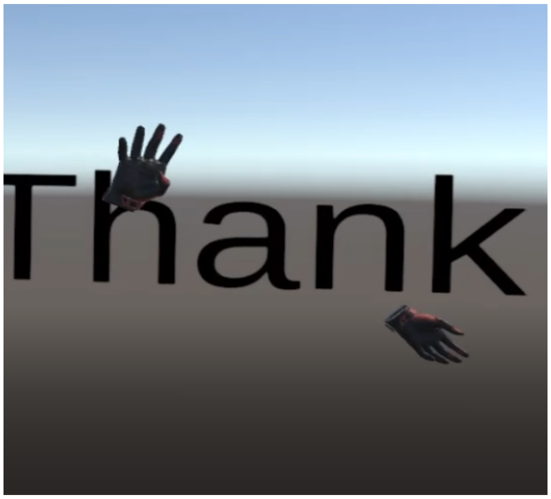
Player Head: Camera

Signs: 13

(Fig 10. Gesture Recognition script in the Unity editor)

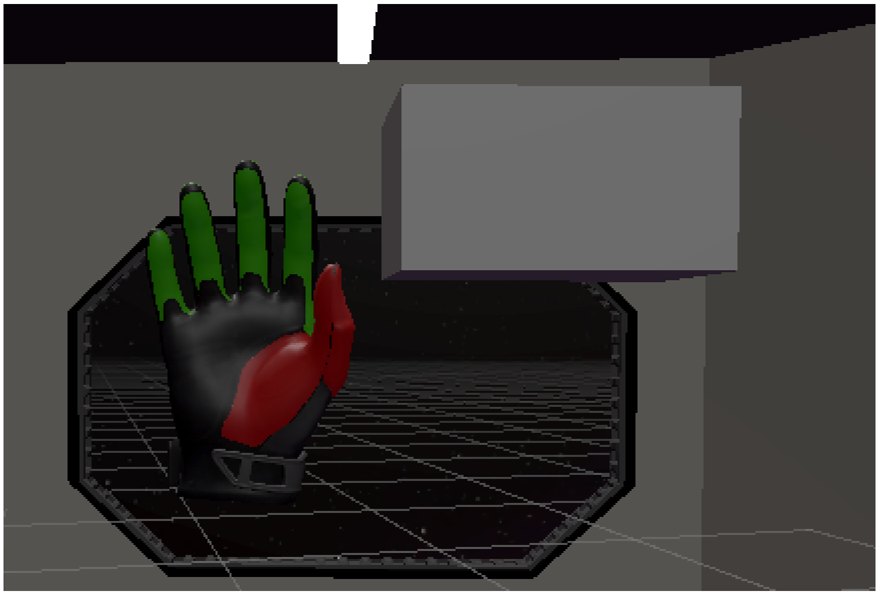
3.3) Feedback system

The feedback system went through two major different interactions. The first iteration showed a mirrored version of the user’s hands in front of them highlighting the parts of the hand which are positioned wrong. In theory, this seemed like a good idea because they can clearly see how they need to move their hand and fingers to perform a selected sign. However, in reality, it was very disorientating and made the experience worst.



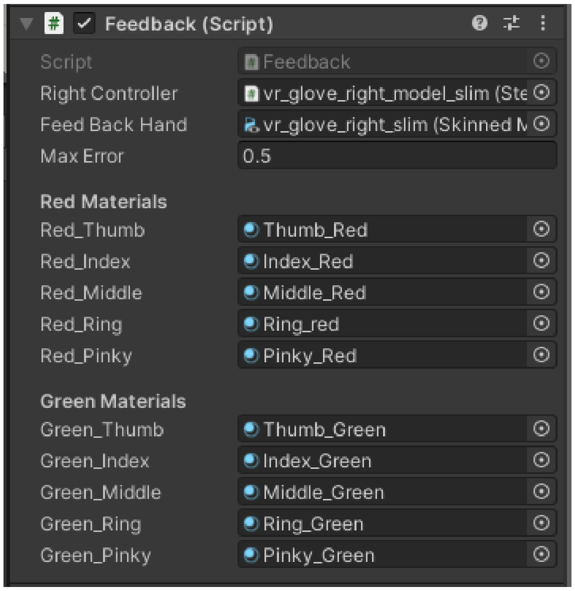
(Fig 11. Old feedback system showing mirrored hands)

The second iteration, and the one currently being used, is much simpler. The user is shown an animation of the sign they need to perform where the hand dynamically changes colours highlighting which fingers the user needs to extend or retract. This approach is much easier to understand which makes the user stay for longer and improves the learning experience.



(Fig 12. Current feedback system with highlighted fingers showing what the user needs to do)

To imploment this the system needs to talk to the gesture recognition and SignList controllers. This is because it needs information on the user's hand and what sign they are meant to be performing so it can display the correct information. In order to change the colour of the right finger, the system compares the value of the user's finger with the target sign, if it is correct the finger turns green, if it is wrong it turns red.



Feedback (Script)

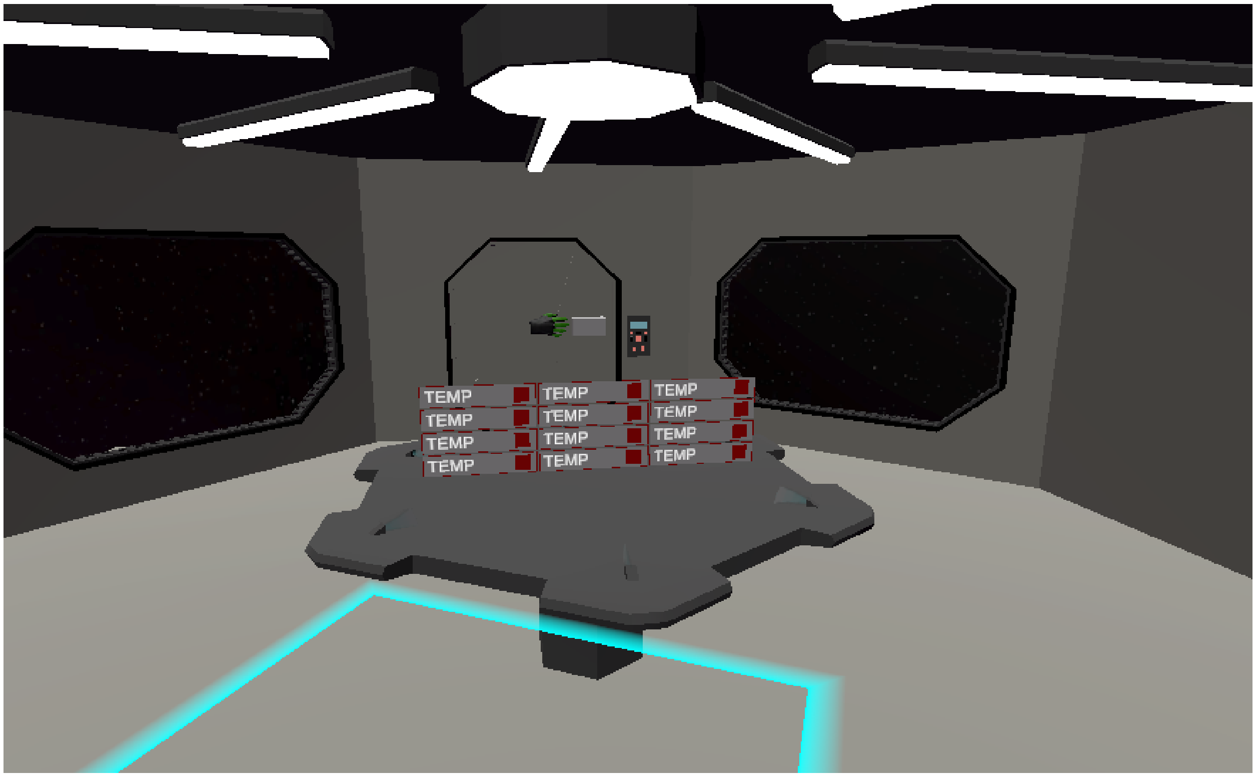
Script	Feedback
Right Controller	vr_glove_right_model_slim (Ste
Feed Back Hand	vr_glove_right_slim (Skinned M
Max Error	0.5
Red Materials	
Red_Thumb	Thumb_Red
Red_Index	Index_Red
Red_Middle	Middle_Red
Red_Ring	Ring_red
Red_Pinky	Pinky_Red
Green Materials	
Green_Thumb	Thumb_Green
Green_Index	Index_Green
Green_Middle	Middle_Green
Green_Ring	Ring_Green
Green_Pinky	Pinky_Green

(Fig 13. Feedback system script in the Unity editor)

3.4) Environment

It is important for the virtual environment to be visually pleasing and interesting so the user does not become bored or uninterested. The original idea was to have the application set in a cafe, however, this didn't feel right because a cafe does not have floating/holographic information and it is not a very good learning environment. Due to this, I wanted to create something futuristic that is also a good place to learn and fits well with floating holograms.

The design I went with is a space station classroom. This is perfect for this application because it works with the floating information and is interesting to look at.



(Fig 14. Space station classroom environment)

In the centre of the room is a table that the user interface sits on. Having the interface on the table works well with the theme that this is a classroom. The user is placed at the edge of the table facing the door and window because this is the most interesting view.

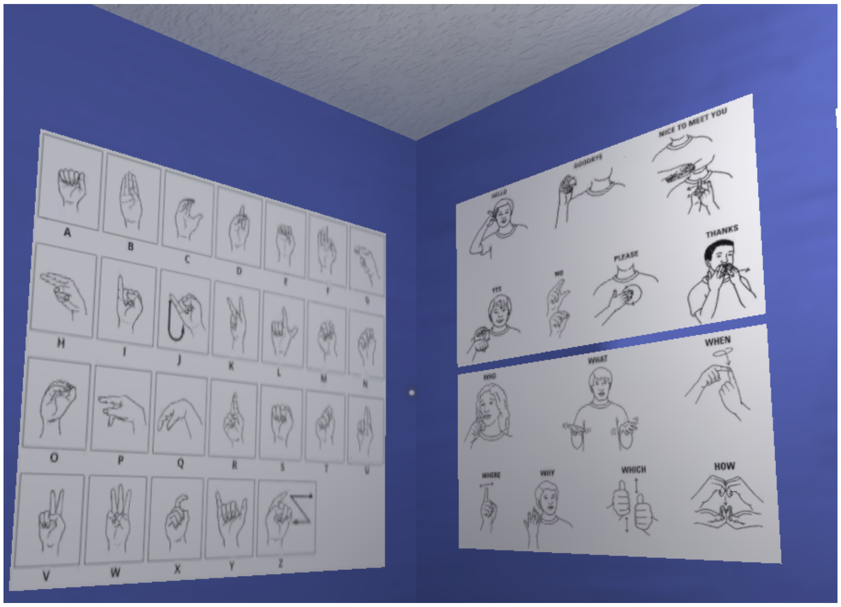
In an earlier version of the application, the interface was placed in front of the user and would stay in front of them if they turned their head. This was good while testing, however, when the environment was added it became confusing because the interface would interfere with the world. This was scrapped and replaced with the current implementation.

4) Research

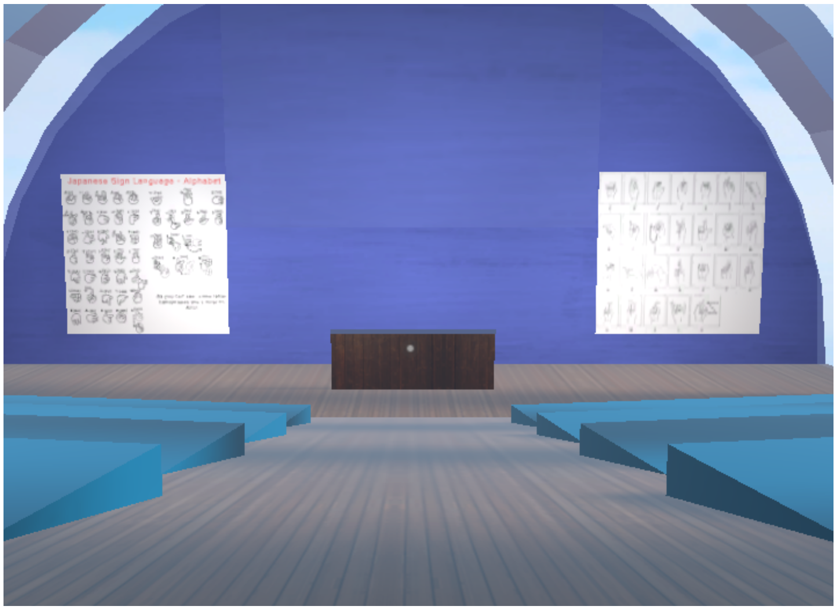
4.1) Existing resources

4.1.1) Zade’s ASL & JSL World

In VRChat there are already some user name resources for learning signs. One of the most popular is “Zade’s ASL & JSL World”. It has two rooms. The first is a simple box with posters on the walls with instructions on how to perform a sign. The second is a theatre with raised seating, a place for a speaker at the front and some signs behind them.



(Fig 15. Box room with posters)



(Fig 16. theatre room)

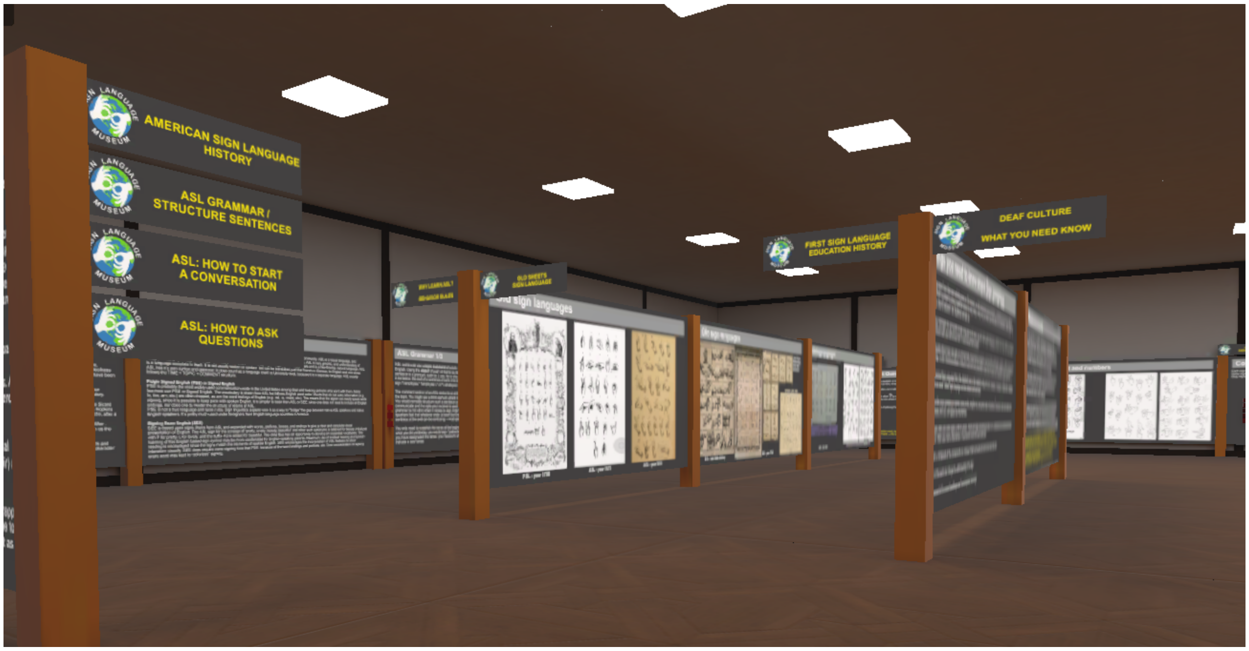
This world is good because they do weekly events hosted by a professional at midnight GMT. The lecturer stands at the front of the theatre and teaches the audience like someone would in real life. However, from my experience at one of these events, it is not very interactive and you are not given any feedback because there are simply too many people for the professional to realistically give advice to everyone.

Additionally, having the events at a set time every week is not ideal for all users because of time zones or other factors.

4.1.2) MrDummy_NL's Sign & Fun

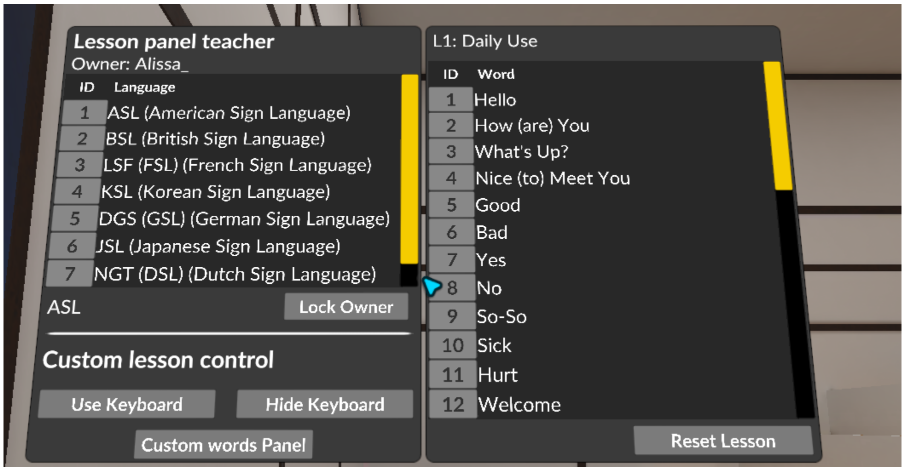
By far the most popular VRChat world for learning signs is "MrDummy_NL's Sign & Fun". This world is fundamentally the same as Zade's, however, it contains much more information and is more focused on the community aspect.

There are three main areas. The first is a museum which contains a vast amount of information on the history of sign language both in real life and in the virtual space.


AMERICAN SIGN LANGUAGE HISTORY
ASL GRAMMAR / STRUCTURE SENTENCES
ASL: HOW TO START A CONVERSATION
ASL: HOW TO ASK QUESTIONS
FIRST SIGN LANGUAGE EDUCATION HISTORY
DEAF CULTURE WHAT YOU NEED KNOW

(Fig. 17 virtual museum room)

The second area is the learning room. This area is similar to Zade's because it has a set of seats for an audience and a stage for a speaker. However, it is much more advanced. The speaker has a screen behind them that they can control with a panel to their side. It lets them pick which language and a topic, and then a set of words will display on the screen.



Lesson panel teacher
Owner: Alissa_

ID	Language
1	ASL (American Sign Language)
2	BSL (British Sign Language)
3	LSF (FSL) (French Sign Language)
4	KSL (Korean Sign Language)
5	DGS (GSL) (German Sign Language)
6	JSL (Japanese Sign Language)
7	NGT (DSL) (Dutch Sign Language)

ASL

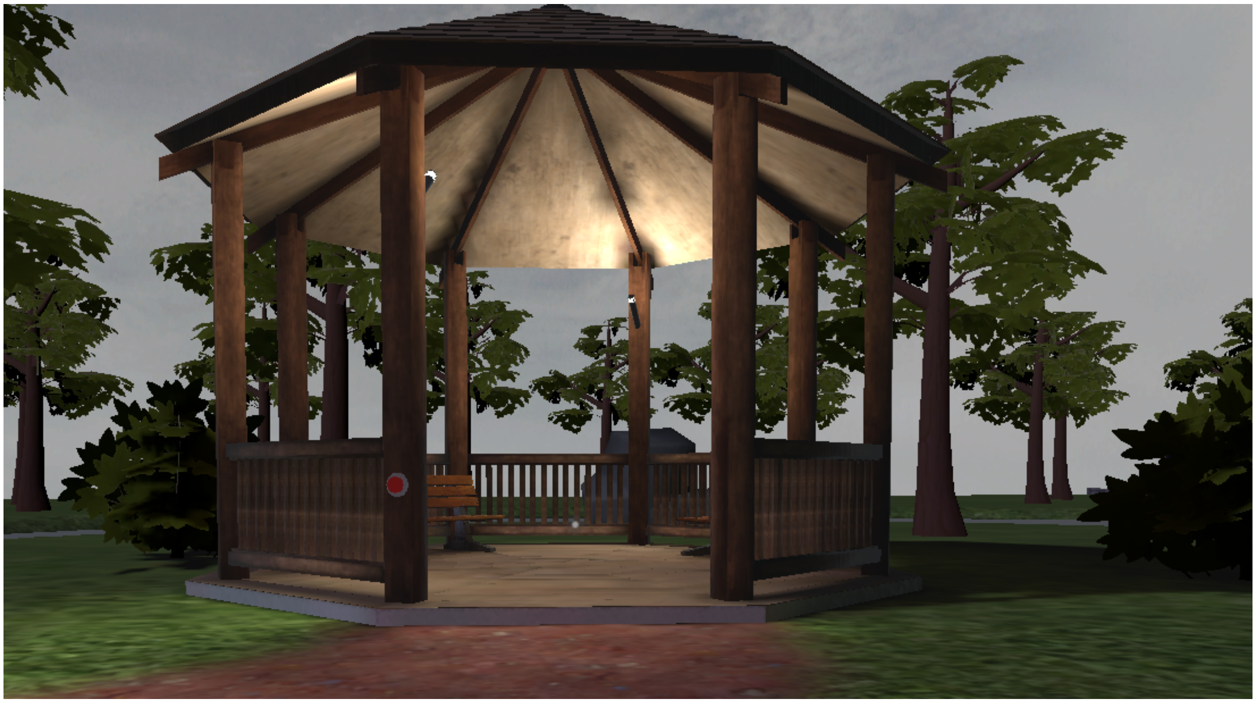
Custom lesson control

L1: Daily Use

ID	Word
1	Hello
2	How (are) You
3	What's Up?
4	Nice (to) Meet You
5	Good
6	Bad
7	Yes
8	No
9	So-So
10	Sick
11	Hurt
12	Welcome

(Fig 18. Stage control panel)

The third area is where the world focuses more on the community aspects, it is an outside seating area. This is used for putting the skills you just learnt into action in a real environment. This is a fantastic idea because it allows the users to try things in a safe space with other people that are learning the same thing as them. The setting is also perfect because the outside area is calming, visually pleasing, and very open.



(Fig 19. Outside seating area)

This implementation is much better than Zade’s because the people in the audience can see what they need to do on the screen while the lecturer is talking.

4.1.3) Overview

Both of these existing resources are great for learning sign language in VR, however, they both have the same problems. Firstly, you can learn by reading the posters placed around the world, this is good, but you don’t get any feedback so you do not know if you’re doing something wrong. Secondly, the only interactive way to learn is to go to one of the planned events and watch someone teach, which is great for some people, but most people would rather learn on their own time or aren’t confident enough to go to an event.

My project will set out to solve both of these problems by proving an environment to learn that is interactive, rewarding, and free to use whenever needed.

4.2) User surveys

To get a better understanding of what other users want from my VR learning application and to see how people feel about using sign language in VR so I asked some of my friends some questions. It is important to get a wide range of people from new users to very experienced so I picked 4 friends that I think would cover this range.

4.2.1) User information

Friend 1:
Username: BiggerChief
Age: 23
VR Experience: Limited. Only plays VRChat occasionally to join friends

Friend 2:
Username: Ratch
Age: 27
VR Experience: Average. Often joins events hosted by friends

Friend 3:
Username: AnxietyBee
Age: 25
VR Experience: Spends lots of time in VR with friends and exploring worlds on their own

Friend 3:
Username: Neon
Age: 28
VR Experience: Very experienced. Hosts her own public events and makes community content.

4.2.2) Questions and Answers

Q1) Do you know any sign language or do you know anyone in VR that does?

Answers:
Friend 1: “No, I don’t know any and I know no one else.”
Friend 2: “I know none, but I have a mute friend that does.”
Friend 3: “I’m a mute so I know some basic words, but nothing too much. Yes, I know lots of people that know it.”
Friend 4: “I don’t know sign language, but I see a lot of people come to events that talk to each other with it. It’s more common than people realise”

Q2) Do you think more people should learn sign language in VR?

Answers:
Friend 1: “I don’t see why not. It can be useful but it’s not for me”
Friend 2: “Everyone should know a second language and sign language is basically one, so yes”
Friend 3: “Absolutely, the mute and deaf community is very big”
Friend 4: “Yes it’s a good life skill to have while being inclusive”

Q3) Do you want to learn or learn more sign language?

Answers:
Friend 1: “No. I don’t meet people that use it enough so I don’t need to know it”
Friend 2: “Yes! I want to talk to my friend without them having to use text to speech or type to me”
Friend 3: “I would like to, but the amount I already know has been fine for now and I mostly use text to speech anyway.”
Friend 4: “I really want to learn it but don’t have the time. When I host an event I give a welcome speech at the start so it would be nice to use it then so if someone is deaf they know what I’m saying”

Q4) How would you prefer to learn in VR? Being taught by someone in a school setting or by yourself using a tool?

Answers:
Friend 1: “If I ever did it would be on my own time”
Friend 2: “I much prefer having someone I can talk to teach me so I can ask questions etc”
Friend 3: “I’m very shy so I don’t want someone else teaching me if I can avoid it”
Friend 4: “Having something that can teach it to me would be perfect because I don’t have time to sit down with someone and learn”

4.2.3) Q&A review

Question 1:
From these answers, it is clear that if you spend enough time in VRChat you will meet someone that knows sign language, for example, the friend that spends little time in VR doesn't know anyone, but the friend that spends the most time sees lots of people and it has become normal to them.

Question 2:
All the participants think more people should learn sign language. Two think it should be something everyone knows, even outside of VR.

Friend three said "the mute and deaf community is very big" which shows that there are lots of people that would benefit from more people knowing how to properly communicate with them.

Question 3:
Three out of the four friends said they want to learn or expand their knowledge so they can more easily talk to people they already know/indirectly know.

Question 4:
The majority said they would prefer to learn on their own. However, one friend said they want to have a teacher so they can ask questions which is a fantastic point because having a world in the game or a standalone application cannot account for everything someone would want to know.

Overview
From this short survey, I've learnt that there is a demand for a way to independently learn sign language in VR by users with medium experience to users with lots of experience.

Testing

Test Name	Description	Actual Result	Notes
Menu rotation	The menu in the centre of the room should look at the user and turn when they move.	This worked as expected. The menu perfectly faces the user at all times no matter their position.	
Menu buttons change colour when hovered over	When the user points the laser at a button it should turn green and when they	This works as expected 99% of the time. The 1% of the time this fails is when	The amount of time this fails is incredibly small so it is not too much of a concern.

	remove the laser it should go back to red.	the user moves the laser across the button extremely quickly and the button is green without being highlighted.	
Menu buttons perform an action when clicked	While the user is hovering over the button and they pull the trigger it will perform an action. In this case, it is the start button. It is expected to hide the main menu and show the learning mode.	This worked as expected with no issues.	
Sign animation	In the learning mode above the interface the correct animation should play, for example, if the word “Hello” is selected the “Hello” animation needs to play	All but one animation worked. When it needed to play the “Thank You” animation nothing happened, but everything else worked.	The “Thank You” animation was not playing because of case sensitivity issues. Once this was fixed the animation played with no issues.
Switching selected sign	Once the user completes the currently selected sign it will automatically move to the next one in the list and loop back to the start when finished.	This worked perfectly with no issues.	
Gesture recognition accurately recognises a sign with a single action	When performing a sign like “One” it should successfully get this sign without waiting for another action.	This worked as expected.	
Gesture recognition accurately recognises a sign with multiple actions	When performing a sign with multiple actions like “Hello” it should successfully wait for the second	This worked as expected.	

	action and output the result of both of them.		

5) Critical Review

5.1) Design

5.1.1) Interface

Overall I am happy with the interface. It is simple enough for the user to not become confused and everything is clear. The design fits well with the world and I am proud of how the menus sit in the centre of the room and follow the user because this went through many iterations and I think the one I chose is the best.

One aspect of the interface I would like to improve is the colours. They are very bland and the grey background blends slightly too much with the grey texture of the environment.

5.1.2) Environment

While working on the environment one of the biggest challenges was making sure everything was scaled properly. This was overcome by constantly exporting any changes from Maya to Unity in order to ensure everything looked correctly sized and the user does not feel claustrophobic because the world is fully enclosed.

This was a great learning experience because I normally only focus on the code so I learned how to create my own virtual environment. The skills from this can be easily transferred from virtual reality to normal 3D game development.

If I were to spend more time I would add extra details and textures to the world, for example, the floor and ceiling are simply a plane, and apart from one wall, they are all blank. However, I think it is enough and I am happy with the result considering my very limited 3D modelling experience before starting it.

5.2) Implementation

5.2.1) Gesture recognition

The gesture recognition system exceeds my expectations. When I set out to make it I was expecting the system to be much more complex, however, I came up with an efficient solution that is easy to expand, works with my different types of signs, and is very accurate.

I am particularly proud of how modular the system is. To add a new sign all I need to do is put the finger positions and distance from the user's head into the array and it will just work. It is also seamlessly integrated with the other game systems and can easily pass information from one to the other.

While working on this system I learned a lot about how to handle virtual reality controllers. They are handled in a very different way to traditional game controls, such as a keyboard and mouse, for example, the user's fingers are a float value between zero and one, whereas a mouse and keyboard would be a boolean so either true or false.

The aspect I would like to expand if I had more time would be to add more signs. I originally wanted to have a lot more signs, however, I was only able to get 4 words and numbers zero to eight. As stated previously it is easy to add more signs, however, it takes a very long time to create the animations for each one.

5.2.2) Feedback system

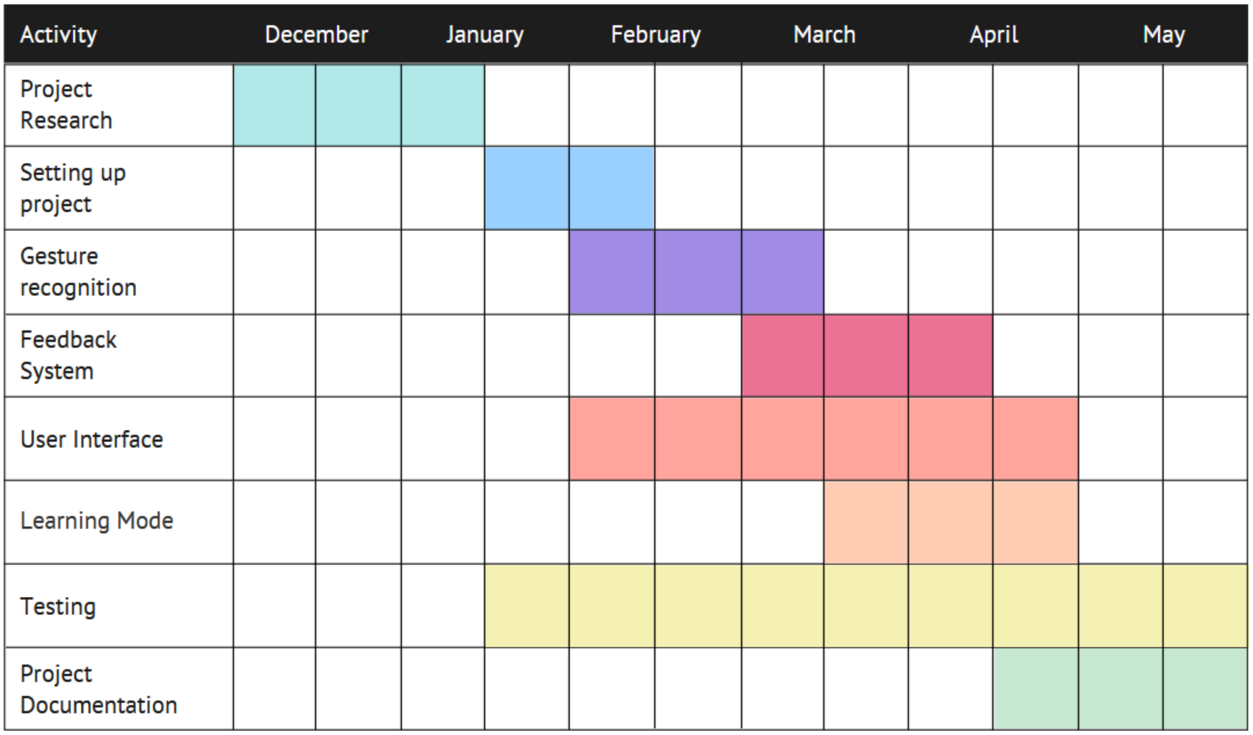
I am happy enough with the feedback system. It works as intended and I think it is a useful edition, however, I am disappointed with my original idea to have a mirror of the user's hands didn't work well enough.

Working on the feedback system taught me how important it is to make a system that feels good to use rather than giving the user as much information as possible. This is because if the system is confusing the user will become disinterested and not want to use it.

5.3) Planning/Project management

5.3.1) Planning/Time management

In order to plan my time I created a Gantt chart. This was incredibly useful because it lets me see when I was expecting a feature to be completed and I could see if I am behind or ahead of schedule.

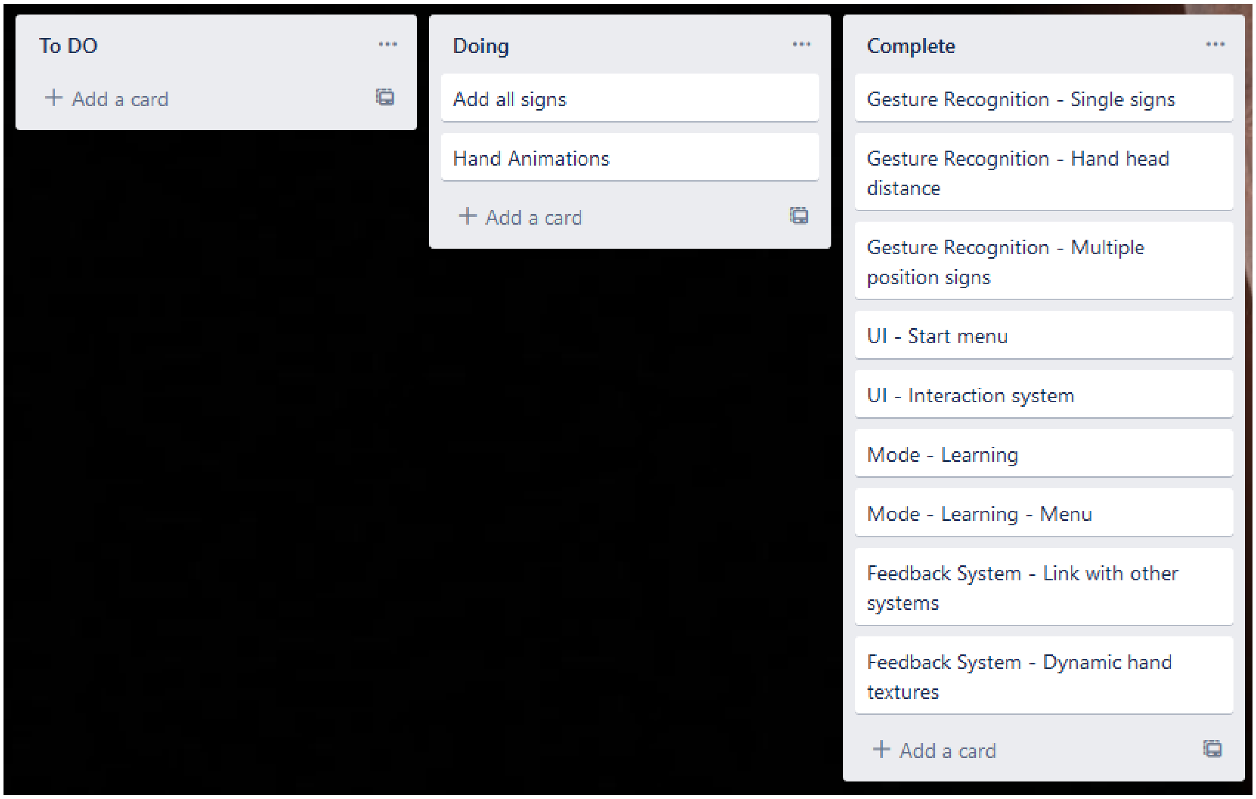


Activity	December		January		February		March		April		May	
Project Research	🟦	🟦	🟦									
Setting up project				🟦	🟦							
Gesture recognition					🟪	🟪	🟪					
Feedback System							🟥	🟥	🟥			
User Interface					🟥	🟥	🟥	🟥	🟥	🟥		
Learning Mode								🟧	🟧	🟧		
Testing				🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
Project Documentation										🟩	🟩	🟩

(Fig 20. Gantt chart)

For the majority of the project, I stuck to the schedule well and everything took around the same amount of time I was expecting it to, apart from the learning mode, which took far longer than I anticipated because of the time it took to create each animation. I allowed for testing throughout the entire schedule which fits with the agile methodology.

Additionally, I used Trello to keep track of individual tasks I needed to complete. This is an excellent tool because I can easily see what needs to be done, what I'm currently working on, and which tasks have already been finished.



To DO

Doing

- Add all signs
- Hand Animations

Complete

- Gesture Recognition - Single signs
- Gesture Recognition - Hand head distance
- Gesture Recognition - Multiple position signs
- UI - Start menu
- UI - Interaction system
- Mode - Learning
- Mode - Learning - Menu
- Feedback System - Link with other systems
- Feedback System - Dynamic hand textures

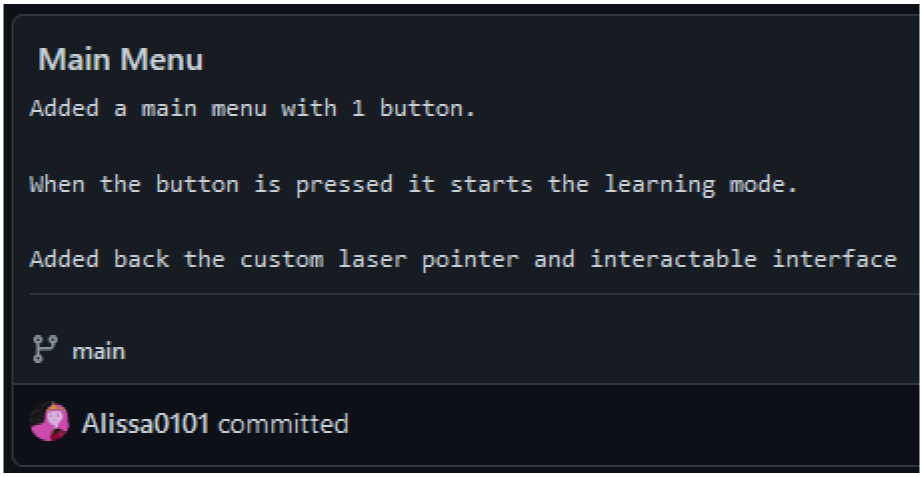
(Fig 21. Trello board showing the tasks I had to complete)

Overall I am happy with how I handled my time while working on this project. Everything was completed on time, and I could keep track of my work well.

5.3.2) File/Project management

In my opinion there is only one option for managing project files, which is to use Git. I decided to use GitHub because I have lots of experience with it. Git was used extensively throughout the creation and was an invaluable tool which saved me from losing all my work. Towards the end of the project I tried to update the Unity VR SDK which failed and corrupted my project. Without backing up the project this would have resulted in me losing all my progress, however, with Git I could quickly and easily revert to an older version.

Additionally, Git is a fantastic way to document changes. I used the description in a commit to describe what had changed since the last version.


Main Menu

```
Added a main menu with 1 button.

When the button is pressed it starts the learning mode.

Added back the custom laser pointer and interactable interface
```

(Fig 22. Github showing a description for a commit)

5.4) Conclusion

In conclusion I am very happy with the final product. I think it met the requirements I set out before starting this project. The experience is simple and easy to understand, the gesture recognition is accurate, and the feedback system displays enough information.

In retrospect, I would have lowered my expectations on how many signs I can add within the timeframe, but I think the ones I added are a good start.

I learned a lot about creating a virtual reality experience in Unity, specifically how to create environments and how to properly handle advanced input devices.

6) References

valvesoftware.github.io. (n.d.). *SteamVR Unity Plugin | SteamVR Unity Plugin*. [online] Available at: https://valvesoftware.github.io/steamvr_unity_plugin/index.html [Accessed 8 Jan 2022].

Vaitkevičius, A., Taroza, M., Blažauskas, T., Damaševičius, R., Maskeliūnas, R. and Woźniak, M., 2019. Recognition of American sign language gestures in a virtual reality using leap motion. *Applied Sciences*, *9*(3), p.445.

Shao, Q., Sniffen, A., Blanchet, J., Hillis, M.E., Shi, X., Haris, T.K., Liu, J., Lamberton, J., Malzkuhn, M., Quandt, L.C. and Mahoney, J., 2020. Teaching American Sign Language in Mixed Reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *4*(4), pp.1-27.

Riches, S., Elghany, S., Garety, P., Rus-Calafell, M. and Valmaggia, L., 2019. Factors affecting sense of presence in a virtual reality social environment: A qualitative study. *Cyberpsychology, Behavior, and Social Networking*, *22*(4), pp.288-292.

7) Appendices

A) Email History

14/10/2021

Summary: Discussed the initial idea for the project.

Suggested action: Research technologies to use for development

17/01/2022

Summary: Viva meeting

Suggested action: Start working more on the project, consider timeframes, research more

18/05/2022

Summary: Confirmed I can do a pre-recorded presentation

B) Consent forms

None of the participants felt comfortable with me using their real names so they included their usernames.

CI601 Participant Consent Form

Title of Project: Virtual Reality Application for Learning Sign Language

Name of Researcher: Bradley Wilson

I have read and understood the information sheet for the above study, and have had the opportunity to consider the information and ask questions.	X
The researcher has explained to my satisfaction the purpose, principles and procedures of the study and any possible risks involved.	X
I am aware that I will be required to take part in (tick relevant box):	
a focus group	X
a usability test	
I understand that my participation is voluntary and that I am free to withdraw from the study at any time without giving a reason and without incurring consequences from doing so.	X
I understand how the data collected will be used, and that any confidential information will normally be seen only by the researchers and will not be revealed to anyone else.	X
I agree to take part in the above study.	X

BiggerChief, 18/05/2022

………………………………………………………………………………………………………

Name of Participant, Date, Signature

Ratch, 18/05/2022

………………………………………………………………………………………………………

Name of Participant, Date, Signature

AnxietyBee, 18/05/2022

………………………………………………………………………………………………………

Name of Participant, Date, Signature

Neon, 18/05/2022

………………………………………………………………………………………………………

Name of Participant, Date, Signature

Bradley Wilson, 18/05/2022,

………………………………………………………………………………………………………

Name of Researcher, Date, Signature