GNG 2101

Design Project User and Product Manual

Project Empathy

Submitted by:

Empathy Factory, Group 4.3

Louis Choinière, 300264221

Jonathan Colasante, 300234532

Omar Abdul-Ghani, 300070139

Shavon Thadani, 300225889

<TEAM MEMBER 5, STUDENT NUMBER>

2022-12-10

University of Ottawa

Table of Contents

List of Figures

List of Tables

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
VR	Virtual Reality
SCM	Source Control Management
HMD	Head Mounted Display
XR	Extended Reality
APK	Android Application Package File
MR	Mixed Reality

Table 2. Glossary

Term	Acronym	Definition
Plastic SCM	Plastic	The Source Control Management software we used
Unity	Unity	The software used to design our VR application/prototype
Low-Polygon	Low-Poly	A term describing a digital 3D asset that utilizes a low number of polygons for its model. This can be for style, performance, both, or other reasons.

Extended Reality	XR	Refers to the open development standard for VR (and related applications)

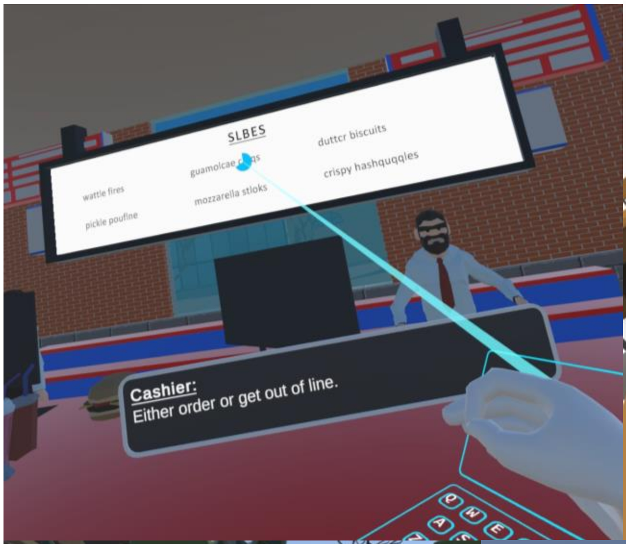
1 Introduction

The goal of this project was to instill empathy in our users towards the dyslexic community by walking in the shoes of someone with dyslexia. Through taking the user on a story of someone with dyslexia, we illustrate the struggles they face on a daily occurrence that most people overlook.

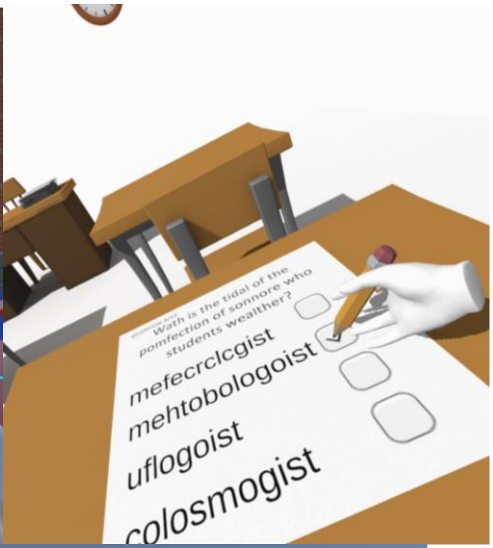
2 Overview

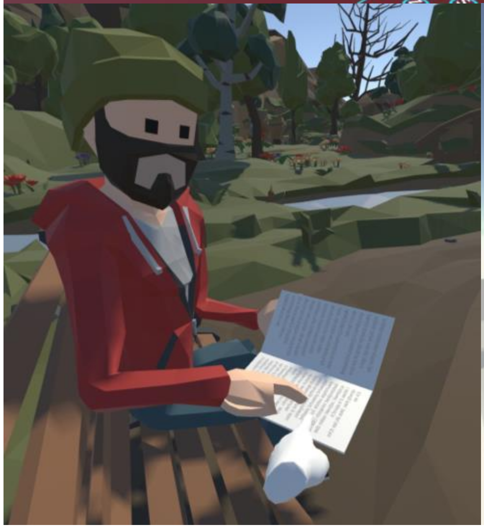
This semester, our project was to develop an immersive virtual environment that would utilise VR in order to increase empathy towards a chosen group of people. It is important to make sure that people with factors outside of their control are represented and that those who care, do what they can to reduce prejudices placed unjustly upon them. Our project is intended as a potential aid for this.

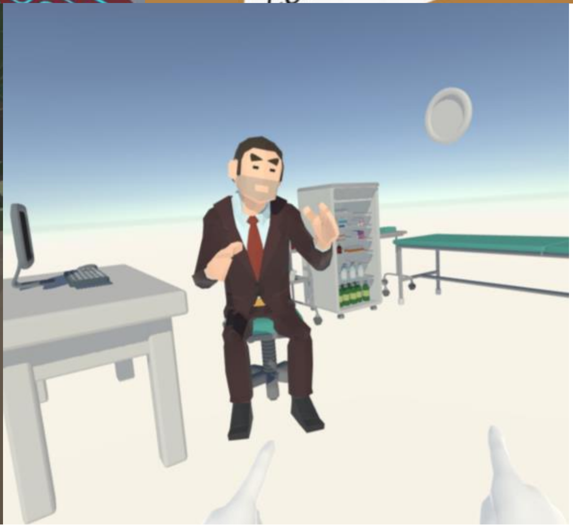
We were not the only group tasked with this project for this semester, and when comparing, there are a few key notable factors that differentiate us from other groups. First and foremost being that we utilised and had multiple scenes functional which no other group managed to achieve which gave us the most expansive and detailed world environment. Second, we also had custom hand and physics interactions as well as many unique and custom mechanics implemented. Third, we heavily focused on our story and its importance when attempting to evoke empathy regarding a very human subject. Below you will find attached screenshots of various elements of our prototype.



Cashier:
Either order or get out of line.


Wath is the tidal of the pomfection of sonnore who students wealther?
mefecrclcgist
mehtobologoist
uflogoist
colosmogist





To summarise, our prototype utilized the Unity game engine to create a virtual immersive experience that utilised virtual reality HMD technologies which would implement a script in order to tell an engaging story regarding the reading and learning disability Dyslexia and how those with Dyslexia experience and deal with this condition. Furthermore, a central part of our prototype is the “decipher” mechanic which is intended to represent (as best we could) what the process of reading with dyslexia can be like. We then utilized a three-act story with a hook, our setup, and our conclusion, for which we respectively had our restaurant scene, our park scenes and ‘branching’ storylines (with accompanying scenes) and our twist/conclusion. Specific elements we used include many low-poly asset packs from the Unity Store as well as the ‘built-in’ Unity XR Toolkit and many of the features provided by Unity such as the input, particle, and animation systems.

2.1 Conventions

We do not use any necessary command conventions in this manual.

2.2 Cautions & Warnings

Many new users of VR can find that many aspects can cause nausea due to a disconnect between the senses and the visuals seen by the user causing the brain to ‘panic’, resulting in motion sickness. Generally, this feeling only occurs during continuous movement or a *complete* lack thereof and goes away after an increased experience like a sailor’s “sea legs”.

3 Getting started

Take note of what VR platform you intend to use (Steam VR, Windows MR, Quest Native, etc.) and install Unity (only needed for development purposes).

3.1 Set-up Considerations

Unity can be a difficult program to work with so take caution when installing Unity itself regarding what version and features you are running. If using a standalone device take note of the battery life and hardware capabilities of the system.

3.2 Accessing/setting-up the System

1) For 'play' use only:
    a) For a Meta/Oculus Quest platform headset: Install the APK via a program such as sidequest and run it under the "unknown sources" tab from the applications menu.
    b) For others: Run the build files as necessary through the desired platform
2) For development:
    a) Download all necessary Unity project files
    b) Open the project in Unity

3.3 System Organization & Navigation

When opening the game for play, the user will be greeted with the start button. Pointing at this button (Or any interactable object) will reveal a blue line from the controller to the button. Pressing the trigger button on same the controller used for pointing will 'click' the button. This will start the game. From there, the 'primary' button on the right controller will skip dialogue interactions and (in future) the 'primary' button on the left controller will go back to previous dialogue and pause time until the present dialogue is returned to.

For development, the folder structure generally follows:

1. All asset pack assets are contained within their own folder/subfolders
2. All custom materials are in the materials folder
3. All custom assets are custom the assets folder
4. All custom scripts are within the scripts folder
5. All custom animation are within the animations folder

6. All sounds are within the sounds folder
7. All scenes are stored within the scenes folder
8. Etc.

3.4 Exiting the System

Close all open/running projects for best practices, save the project, exit.

4 Using the System

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of the <System Name and/or Acronym>.

4.1 <Given Function/Feature>

<Moving around in game/Movement>

First, the user must place the VR Headset on their head. Once launched into the game, they will be able to see and experience the first scene. In order to move around in the virtual world, all the user has to do is to use the thumb joy-stick attached to the controllers. They have a 360 degree of motion and can allow the user to move forward, backwards, side-to-side as well as up and down.

<Interaction in game/Using the controller given>

Once launched into the game, you will be required to place the VR Headset on your head, as well as to equip the two controllers provided. To interact with any object or feature in the games, all you simply need to do is aim the right controller at the object or feature you want to interact with. You will see a circle that goes from 0 degrees to 230 degrees over the object. This signifies that you want to interact with this, and the game is recognizing that. Once the circle is fully defined, you will be able to interact with anything that the feature provides.

4.1.1 <Given Sub-Function/Sub-Feature>

There are no sub-functions in this immersive experience.

5 Troubleshooting & Support

To troubleshoot, rollback changes using the Plastic SCM. In addition, leverage branching and only merge with the main branch once yours has been debugged. Moreover, you can contact us 613-519-9898.

5.1 Error Messages or Behaviors

Errors that you will get will be found in the Unity console. To debug them, leverage the Unity documentation and Unity’s community to find solutions.

5.2 Special Considerations

There are no special considerations.

5.3 Maintenance

No maintenance is required on the VR application.

5.4 Support

For support, reach out to 613-519-9898 or email us at empathyfactory@gmail.com.

6 Product Documentation

6.1 Prototype

We were tasked with a software project and as a result, this document as well as any code comments found is scripting files serves as the only form of documentation used.

When designing and constructing our prototype, our first decision was which application to use in order to develop it. There were a handful of choices at our disposal, but our choices were quickly narrowed down to either Unity or Unreal Engine. We ended up choosing Unity due to the general recommendation that Unity is easier to learn as well as the handful of existing tutorials we had already come across compared to none for Unreal Engine.

Our second choice was which topic to choose for our application. To decide on this, we had all members of the group brainstorm three separate topics with an accompanying storyline after which we reconvened and held an anonymous voting session in order to narrow down the candidates. After which, we were left with six, and we held a discussion and second voting session to narrow down our candidates to three. We then used a decision matrix to decide on our final design. However, after that, many people were skeptical about the scope of that design and suggested an alternative, which we then held a third vote and decided to move forward with the alternative.

	Story 1	Story 2	Story 3
Achievable Concept	2	2	3
Compelling Story	2	1	3
Immersive environment	1	3	2
Total Score	5	6	8

6.1.1 BOM (Bill of Materials)

Item Name	Description	Quantity	Unit cost	Extended cost

VR Headset	VR Headset	1	Free(Provided by the University)	Free
Unity	Game engine	6	Free	Free
Assets	3D and audio assets	As many as we used	Free	Free
Plastic SCM	Source Control Management	1	7$/month	21$ for 3 month
Computer	Computer able to run unity	6	Free(everyone in the group has one)	Free

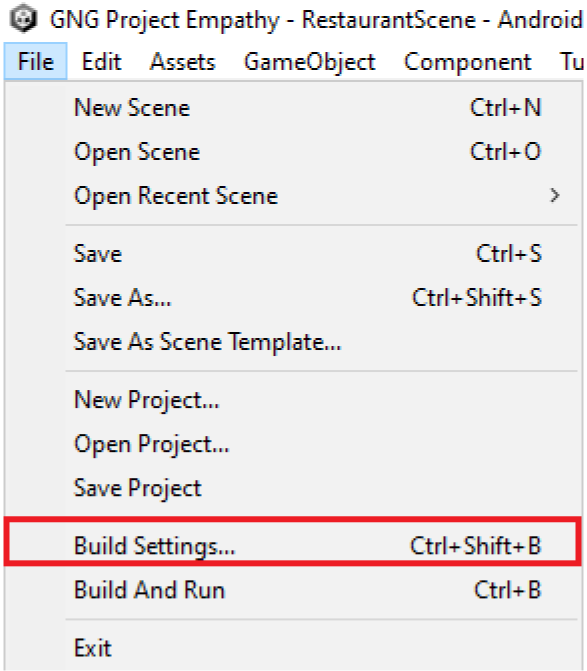
6.1.2 Equipment list

A computer capable of running Unity and a VR headset.

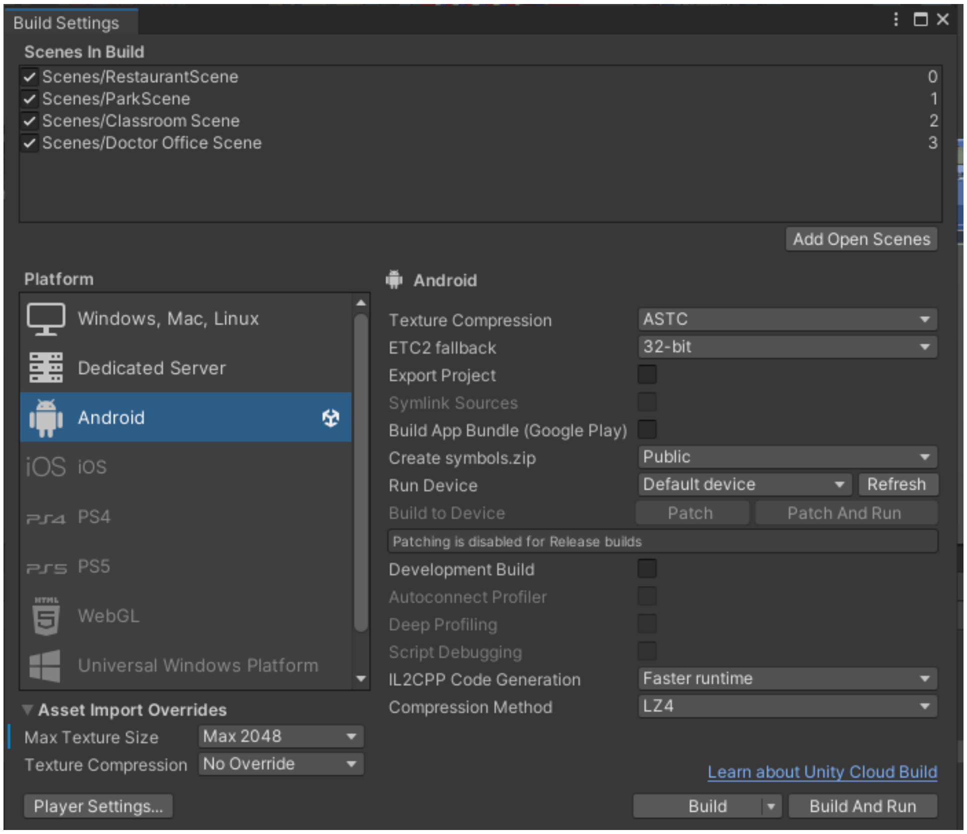
6.1.3 Instructions

To build this specific prototype, first install Unity and open the project.

Go to File ---> Build Settings



You will then be presented by a screen similar to below.



Scenes In Build

☑ Scenes/RestaurantScene 0
☑ Scenes/ParkScene 1
☑ Scenes/Classroom Scene 2
☑ Scenes/Doctor Office Scene 3

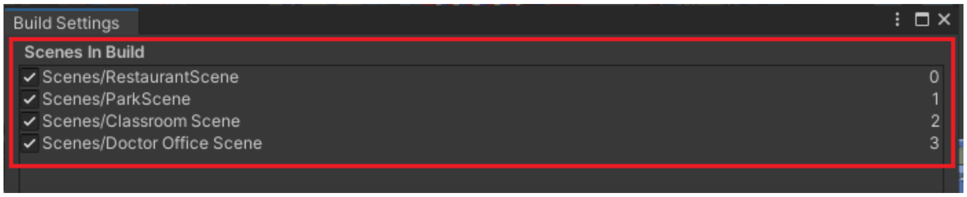
Android

Texture Compression: ASTC
ETC2 fallback: 32-bit
Export Project: ☐
Symlink Sources: ☐
Build App Bundle (Google Play): ☐
Create symbols.zip: Public
Run Device: Default device
Build to Device
Patching is disabled for Release builds
Development Build: ☐
Autoconnect Profiler: ☐
Deep Profiling: ☐
Script Debugging: ☐
IL2CPP Code Generation: Faster runtime
Compression Method: LZ4

Asset Import Overrides

Max Texture Size: Max 2048
Texture Compression: No Override

First, check that all desired scenes are added and checked in the order seen below.



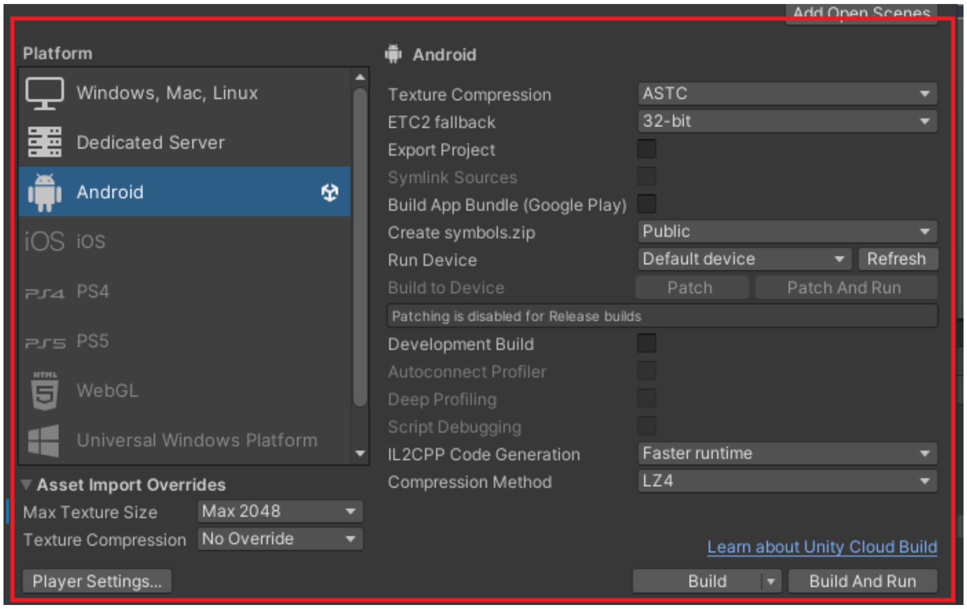
Scenes In Build

☑ Scenes/RestaurantScene 0
☑ Scenes/ParkScene 1
☑ Scenes/Classroom Scene 2
☑ Scenes/Doctor Office Scene 3

Depending on the platform you want to build to will determine the next steps.

For Android/Quest/Standalone:

Go to the ‘Android’ tab as seen highlighted in blue and verify that all of the settings are the same or similar. The “Build” and “Build And Run” buttons will be grayed out if you are using a version of Unity without the correct build tools installed.

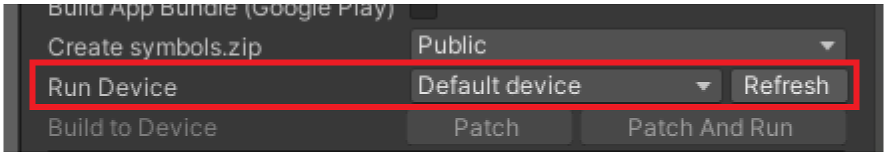


Texture Compression ASTC
ETC2 fallback 32-bit
Create symbols.zip Public
Run Device Default device
Patching is disabled for Release builds
IL2CPP Code Generation Faster runtime
Compression Method LZ4
Max Texture Size Max 2048
Texture Compression No Override

Next, decide whether you want to build and install manually or build and install automatically.

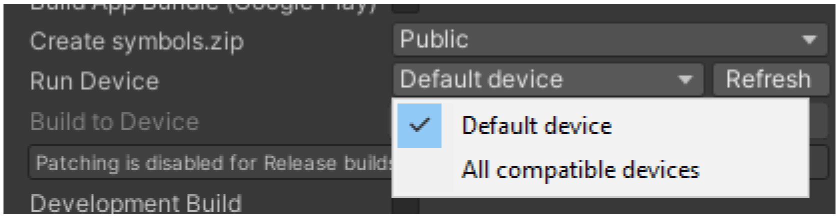
To build and install manually, hit the “Build” button, select a file location to store the generated APK file and use a program like SideQuest in order to install the APK to your headset.

The build and install automatically, the first step is to plug in your headset. Make sure developer mode is enabled on the headset and that your PC is allowed access to the headset’s file system. If you have set everything up correctly, hitting refresh on the below button will refresh all attached devices and populate the dropdown with your headset if it is found.



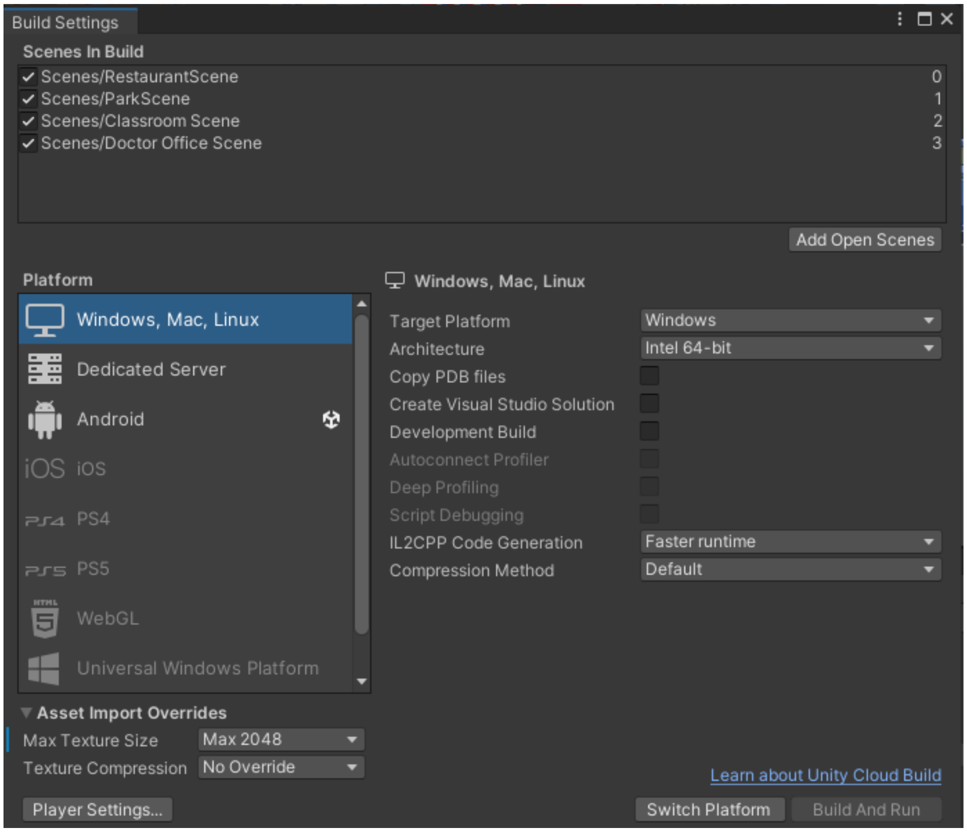
Create symbols.zip Public
Run Device Default device

Your headset will show up in this dropdown if it has worked. Select it and bit “Build and Run”. Let Unity build and install the project, after which point


Create symbols.zip: Public

Run Device: Default device

For non-android platforms, click on the tab below (highlighted in blue). Verify settings as mentioned and click the ‘switch platform’ button. Wait for Unity to complete the transfer and then click either ‘build and run’ if you have your headset set up already or ‘build’ if you just want to generate a complete build of the game for potential use later.


Scenes In Build

☑ Scenes/RestaurantScene 0
☑ Scenes/ParkScene 1
☑ Scenes/Classroom Scene 2
☑ Scenes/Doctor Office Scene 3

Windows, Mac, Linux

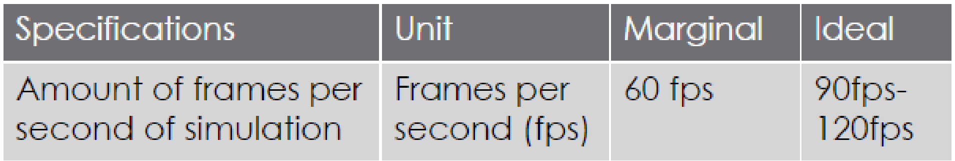
Target Platform: Windows
Architecture: Intel 64-bit
Copy PDB files: ☐
Create Visual Studio Solution: ☐
Development Build: ☐
Autoconnect Profiler: ☐
Deep Profiling: ☐
Script Debugging: ☐
IL2CPP Code Generation: Faster runtime
Compression Method: Default

Asset Import Overrides

Max Texture Size: Max 2048
Texture Compression: No Override

6.2 Testing & Validation

After completing our prototype in Unity, several tests were conducted to validate the final design. The first test was a user experience test, where users provided feedback on the overall experience of the game. We received positive feedback from all testers which validated that we had made a good prototype. A performance test was also conducted to measure the game's performance and stability to make sure it fell within our specified values, which it did by a mile (250fps+).



Specifications	Unit	Marginal	Ideal
Amount of frames per second of simulation	Frames per second (fps)	60 fps	90fps-120fps

In addition, compatibility tests were performed to ensure the game worked well with a variety of VR headsets and hardware configurations. Usability and functional tests were also carried out to ensure the game was easy to use and all features were working correctly. Other tests included a playability test to ensure the game was fun, a localization test to verify the translation of the game into different languages, and an accessibility test to evaluate the game's options for players with disabilities. Overall, these additional tests confirmed the design of the game was successful and ready for release.

7 Conclusions and Recommendations for Future Work

The lessons learned throughout this project include always staying organized and being on top of your assigned task. In relation to the prototype that our group had designed, these lessons are crucial to completing the task on time and efficiently. Our prototype went through a lot of iterations in the design stage since we had plenty of ideas and wanted to see how they'd look. However, the final prototype we presented best represented our clients' needs so as a group, we decided with that. For future work, the most productive avenues for other groups to continue and improve upon our work is to focus more on the mechanics side of Unity. Implementing more movement and interactivity within each scene would be ideal. However, everything else in terms of the scenes, assets and creations are perfectly good to go for future teams if they were to continue our project.

If our group had a few more months to work on this project, we would focus a lot more on asset quality and starting the incorporation of game mechanics early on and refine it as much as possible. This is because some of our plans for scene transitions and certain movement mechanics had to be abandoned due to the lack of time. These were important mechanics to add into our immersive experience because it enhances the user experience and allows for a better portrayal of feeling empathy for the user since it allows them to truly feel and be immersed in each scene.

8 Bibliography

Online Playlist of Unity VR development tutorials we used: How to make a VR game in UNITY in 2022

APPENDICES

9 APPENDIX I: Design Files

This user manual stands alone as the only form of documentation (excluding script comments) available. However, this playlist served as a great jumping-off point during the creation of our prototype.

MakerRepo project link: Project Empathy | MakerRepo (makerepo.com).

10 APPENDIX II: Other Appendices