

GNG1103
Design Project User and Product Manual

Invis-Ability - A VR Immersive Experience Tool

Submitted by:

Invis-Ability - Team F1.1

Nicole Bégin, 300261947

Honor Hinchey, 300167971

Anthony Talevi, 8833792

Yuteng Li, 300210312

April 8th, 2022

University of Ottawa

Table of Contents

Table of Contents.....	ii
List of Figures.....	iii
List of Tables.....	iv
List of Acronyms and Glossary.....	v
1 Introduction.....	1
2 Overview.....	2
2.1 Cautions & Warnings.....	3
3 Getting started.....	4
3.1 Configuration Considerations.....	4
3.2 User Access Considerations.....	4
3.3 Accessing/setting-up the System.....	4
3.4 System Organization & Navigation.....	6
3.5 Exiting the System.....	9
4 Using the System.....	10
4.1 Home Scene.....	10
4.2 Office Scene.....	12
4.3 Reflection Scene.....	17
5 Troubleshooting & Support.....	19
5.1 Error Messages.....	19
6 Product Documentation.....	20
6.1 Top Level View.....	20
6.2 General Scene Structure and Components.....	20
6.2.1 Models, Prefabs and Visual Construction of the Scene.....	20
6.2.2 XR Interaction Toolkit.....	20
6.2.3 XR Origin.....	20
6.2.4 Character Model and VR Constraints.....	21
6.2.5 Teleportation.....	22
6.2.6 Managers.....	22
6.2.7 User Interface.....	22
6.2.8 Interaction Scripts on Assets.....	23
6.2.9 NPC Interaction.....	26
6.2.10 Player’s Report.....	27
6.2.11 ADHD Distraction Mechanisms.....	27
6.3 Testing and Validation.....	28
6.4 Design Choices.....	28
6.4.1 Sound Design.....	28
6.4.2 Art Design.....	29
6.4.3 Gameplay.....	29
6.5 BOM (Bill of Materials).....	29
6.6 Equipment List.....	30
6.7 Testing & Validation.....	30
7 Conclusions and Recommendations for Future Work.....	31
8 APPENDIX I: Design Files.....	32

List of Figures

Figure 1: Block Diagram Outlining the Experience	3
Figure 2: Project Window	5
Figure 3: Open button in the inspector window.....	5
Figure 4: Location of the play button in the Unity Editor	6
Figure 5: Location of the Game Window in the Unity Editor	6
Figure 6: The three main scenes of the experience.....	7
Figure 7: Front door of the Home Scene	8
Figure 8: Task HUD displaying the user’s active task	8
Figure 9: Unity scene play button turns blue when active.....	9
Figure 10: User teleporting using a ray-caster bound to the right-hand controller.....	10
Figure 11: User picking up a coffee mug.....	11
Figure 12: User opening a door in the Home scene.....	11
Figure 13: Location of the front door of the Home scene.....	12
Figure 14: Location of the manager’s office with respect to the user’s starting position.....	12
Figure 15: Location of the desks with respect to the player’s starting position	13
Figure 16: Report on the user's desk.....	13
Figure 17: User’s view with when the report is picked up	14
Figure 18: Usable coffee mugs on the kitchen counter.....	14
Figure 19: Dirty dishes that can be put in the sink	15
Figure 20: More dirty dishes that can be put in the sink, along with the sink itself.....	15
Figure 21: Conversation with co-worker	16
Figure 22: Coffee cup on the player’s desk	16
Figure 23: User’s view while reading the report	17
Figure 24: Conversation with manager.....	17
Figure 25: Reflection scene	18
Figure 26: Ending text in the Reflection Scene	18
Figure 27: XR Origin game object hierarchy	20
Figure 28: VR constraints game object hierarchy.....	21
Figure 29: Heads Up Display displaying the user’s active task.....	23
Figure 30: HUD Canvas and child objects	23
Figure 31: Door animator.....	24
Figure 32: XR grab interactable and other components necessary for objects to be grabbable.....	25
Figure 33: Example of a collider used to detect task completion	26
Figure 34: Report to be read by the user as their main task.....	27
Figure 35: Distracting lights mechanism active when reading the report or talking to the co-worker	28

List of Tables

Table 1. Acronyms.....	v
Table 2. Glossary	v
Table 3. Bill of Materials	29
Table 4. Referenced Documents	32

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
ADHD	Attention Deficit Hyperactivity Disorder
AR	Augmented Reality
EDI	Equity, Diversity, and Inclusion
HR	Human Resources
VR	Virtual Reality

Table 2. Glossary

Term	Acronym	Definition
Extended Reality	XR	A term used to describe all real-or-virtual environments that are generated by wearable technologies. Includes virtual reality, augmented reality, mixed reality, and others.
Heads-up Display	HUD	A system that projects information directly to a user's point of vision.
User Interface	UI	The space where interactions and communications between humans and computers occur.

1 Introduction

This User and Product Manual (UPM) provides the information necessary for all users to effectively use the Invis-Ability immersive experience tool and for prototype documentation.

At the beginning of the Winter 2022 semester, the project team was presented with a need for a library of immersive experience tools for equity, diversity, and inclusion (EDI) training from Hanan Anis, the head of the School of Engineering Design and Teaching Innovation at the University of Ottawa. The assumption was made that this tool would use Virtual Reality (VR), which has the unique ability to immerse users and create more powerful experiences to allow them to experience situations as if they were a person that belongs to a marginalized group. To accomplish this, Invis-Ability decided to focus on creating an experience that demonstrates the daily struggles of a person with invisible disability, specifically attention deficit hyperactivity disorder (ADHD). The subsystems included in this product were sound design, art design, gameplay, and experience structure. The final prototype developed in Unity is accessible through the Makerepo repository for Invis-Ability and is included in Appendix I.

The purpose of this document is to provide detailed documentation of the work done on the prototype so that the client or other potential users can understand how to use it. First, this report will include an overview of the problem and its importance, the user's needs, and the key features of the product. Then, it will provide a general walkthrough of the final prototype, including considerations of configuration and user access, system access, navigation, and exit, and a detailed description of all features and functions included in the prototype. All recovery, testing, and error correction procedures, error messages, support resources, and product documentation, including a bill of materials (BOM) and equipment list, will also be provided. Finally, the last section of the manual will include lessons learned from the creation of this prototype and recommendations on the most productive avenues for future work.

The user is assumed to be a human resources (HR) manager looking to use Invis-Ability as an immersive EDI training tool to educate their employees on the challenges faced by people with invisible disabilities, specifically in the workplace environment. Due to the format of this UPM, this document ensures users' privacy information is secure and cannot be accessed. Moreover, since the prototype can only be accessed on one computer at a time, any device-specific information will be secured on the user's hardware only.

2 Overview

This EDI training tool focuses on invisible disabilities. We wanted to bring awareness to invisible disabilities because they are often overlooked and misunderstood since they are not obvious at first glance. In addition, invisible disabilities are very common and usually have large impacts on a person's work and behaviour. A disability is a condition of the body or mind that limits or prevents the person with the condition from engaging in certain activities and interacting with the world. Invisible disabilities tend to be a bit trickier to deal with since they are often hidden, and the level of the condition may vary and be more affected by different circumstances. For that reason, it is important to bring awareness and promote education about this problem.

Our solution will help solve this issue and foster inclusivity by allowing users to immerse themselves in a virtual reality where they experience some of the symptoms associated with a particular invisible disability. After using our tool, we hope that users will be more aware of the social stigma and difficulties people with invisible disabilities experience in their daily lives. Our experience offers an easy-to-use, interactive, immersive, and engaging solution that helps build empathy while showing another's POV, to foster diversity and inclusion and fulfill the educational needs of the user. The product helps the user reflect on what it's like to be different.

While creating our product, we did some research about existing products to get some ideas on what worked well and to see how we could make our product better. Based on existing products, interactive objects and shorter experiences made the experience more meaningful and engaging, which we made sure to add to our experience. Our solution also improves upon the benchmarked products because it includes more opportunities for users to interact with the virtual world, a voiceover of the character's inner monologue, and guided reflection, making it more immersive, interactive, and overall effective as an EDI training tool.

Our solution to help with invisible disability training is a VR experience in which a user must complete general everyday work tasks while being affected by the common symptoms of the invisible disability. The key features that make our product most efficient are the following:

- A character selection screen in which the user picks the character (and therefore, without knowing, the invisible disability) that they will be in the experience.
- A home scene that allows the user to get used to VR before they must start completing tasks.
- An office scene is the main experience and where the tasks would start to be affected by the invisible disability's symptoms
- A conclusion that includes a guided reflection and additional resources

This tool was designed for an Oculus Rift headset and HTC Vive VR system. Buttons on the controllers are used to move around the VR space and the headset allows the user to view the experience. Once the VR system has been turned on and connected to the Invis-Ability, Instructions are shown on screen for the rest of the experience. The user will encounter an introduction screen in which the click on a button allows the user to continue with the main experience. At each step, the user is prompted to complete different tasks and instructions are shown on the screen. When using VR, it is important to have space but always read the manual for the particular VR system you are using.

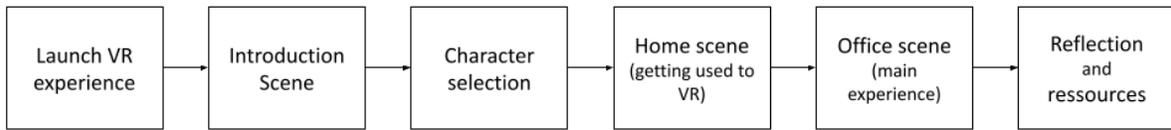


Figure 1: Block Diagram Outlining the Experience

2.1 Cautions & Warnings

Before using this tool, make sure to read the instructions for the VR system and headset you are using as some risks are associated with the use of VR systems such as personal injury, nausea, and discomfort.

3 Getting started

This section will provide a general walkthrough of the system from setup/initiation through exit. The logical arrangement of the information will allow users to understand the sequence and flow of the system.

3.1 Configuration Considerations

To run the experience users will need the following hardware and software:

- Oculus Rift, HTC Vive, or Valve Index VR headset and accessories, including two hand controllers
- Computer with a dedicated graphics card and hardware powerful enough to run VR devices (see VR hardware manufacturers for recommended computer hardware specifications).
- Unity3D

This manual assumes the user can prepare the VR hardware and does not provide instruction to do so. For help, see your VR hardware's installation and setup guide. Ensure that there is plenty of room for the user to move about while they are completing the experience.

Prior to launching the experience from the Unity editor, the VR hardware must be set up and connected to the computer, and the host software (e.g., Oculus) must be running.

3.2 User Access Considerations

The VR experience has been designed and developed to be as accessible as possible, however it will still not be accessible to everyone. Users prone to motion sickness should complete the experience with caution as virtual reality has been known to induce the affliction. Those with epilepsy should also proceed with caution. Virtual reality has been known to induce reactions in epileptics, and this experience has not been tested to determine its effect on epileptics. Two hands are required to complete the experience, so those lacking full dexterity with both of their hands may need assistance to complete the experience or may not be able to participate outright.

3.3 Accessing/setting-up the System

The VR experience can be accessed for free from <https://makerepo.com/hhinc076/1157.gng1103f-team-1-1-invisibility>. Download the package and unpack the folder to your preferred location on your PC.

Launch the Unity3D editor, and open the project contained in the unpacked folder. The project is large and will take a few minutes to load. The project should open to the 'Home' scene. If it does not, in the Project window, navigate to the Scenes folder and double click on the asset 'home_scene' to open the 'Home' scene. Alternatively, you can single click the asset, and click the "Open" button in the Inspector window.

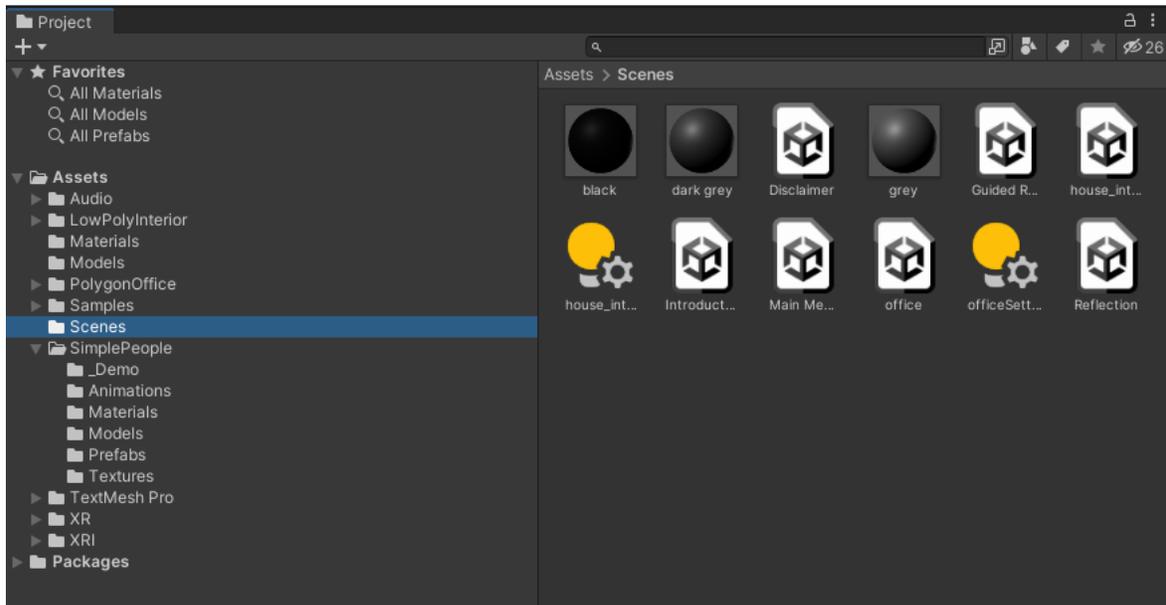


Figure 2: Project Window

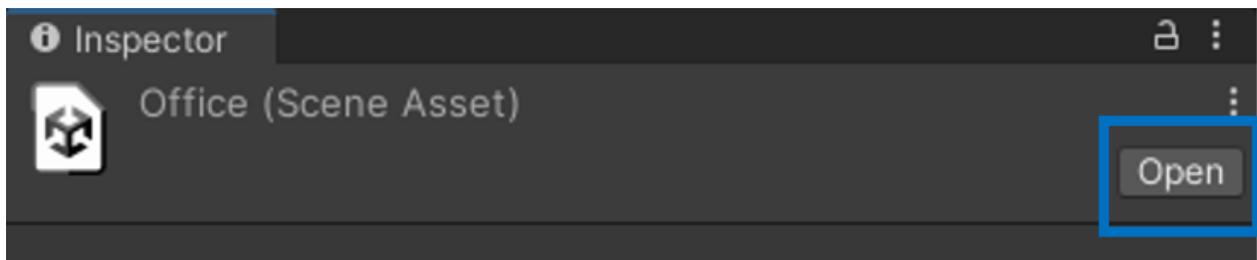


Figure 3: Open button in the inspector window

Once the 'Home' scene has loaded, you are ready to launch the experience.

Ensure that the user is donning the headset and has the controllers in their hand. Then, press the *Play* button at the center-top of the Unity window. The experience will launch, and the user will be placed into the virtual environment. This can be confirmed through the *Game* window of the Unity Editor.

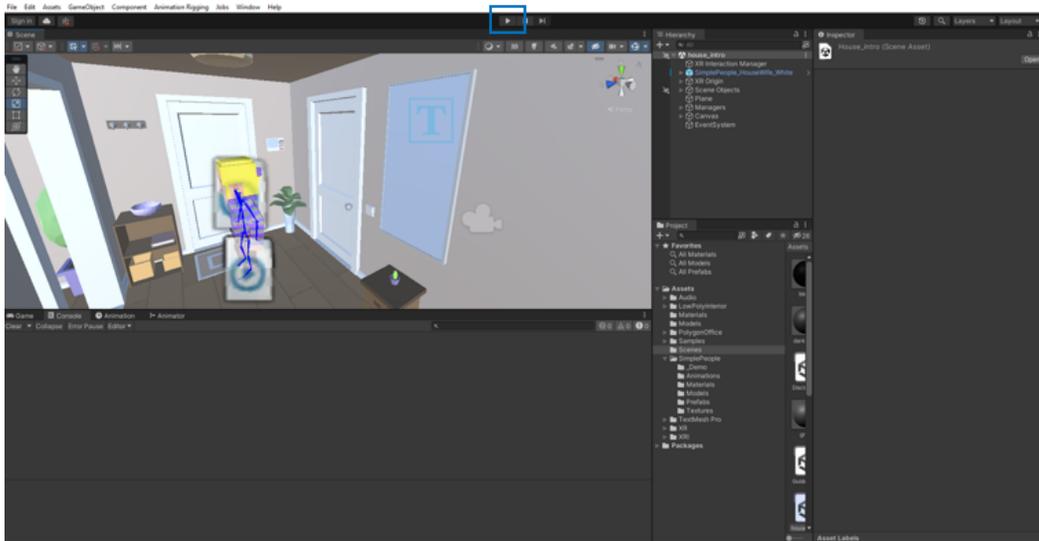


Figure 4: Location of the play button in the Unity Editor

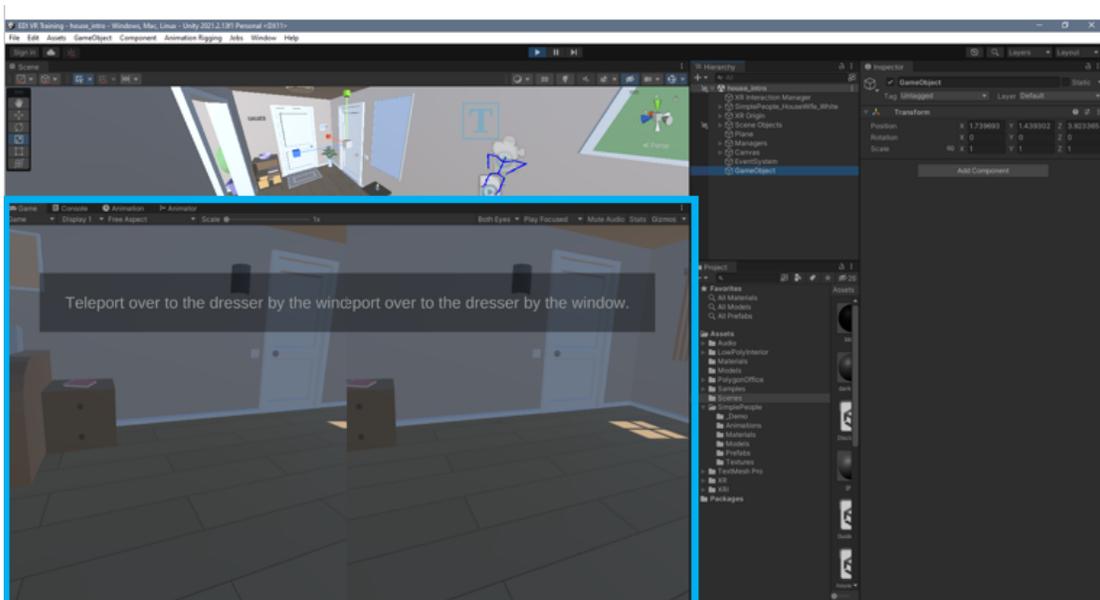


Figure 5: Location of the Game Window in the Unity Editor

3.4 System Organization & Navigation

The experience is separated into three scenes: the *Home* scene, the *Office* scene, and the *Reflection* scene.

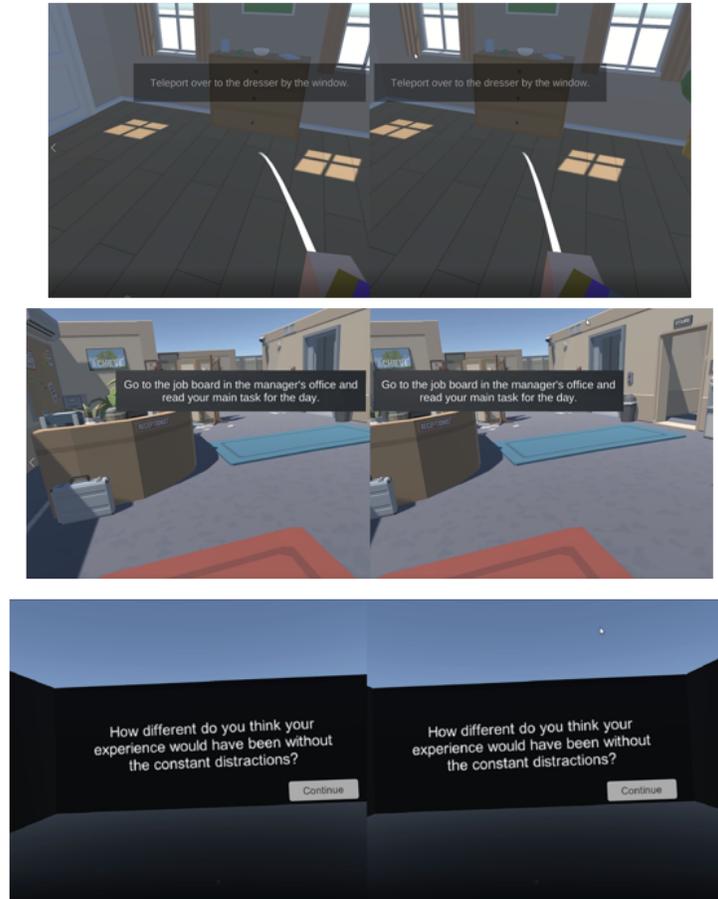


Figure 6: The three main scenes of the experience

Each scene serves a different purpose. The *Home* scene serves to introduce the user to the virtual environment and explains the mechanics of the world. The *Office* scene contains the core experience where the user will experience ADHD firsthand. Lastly, the *Reflection* scene serves to enhance the emotional education of the user by posing a series of reflective questions. The user experiences these scenes in the order that they have been described.

Each scene contains a link to the next. In the *Home* scene, the user can move to the *Office* scene by opening the front door.



Figure 7: Front door of the Home Scene

In the *Office* scene, the user moves to the *Reflection* scene by walking out of the manager's office after they've spoken to their manager at the end of the day.

The *Reflection* scene ends whenever the user is ready to exit the experience and removes their headset.

In the first two scenes, the user is assigned tasks that serve to move the experience forward. These tasks are assigned by the program, typically upon the completion of the previously assigned task. Whenever a new task is assigned, it will be shown in text at the top of the user's view for 5 seconds. Should the user wish to see the assigned task again, they can simply press the secondary button on either controller at any time to show the currently assigned task.

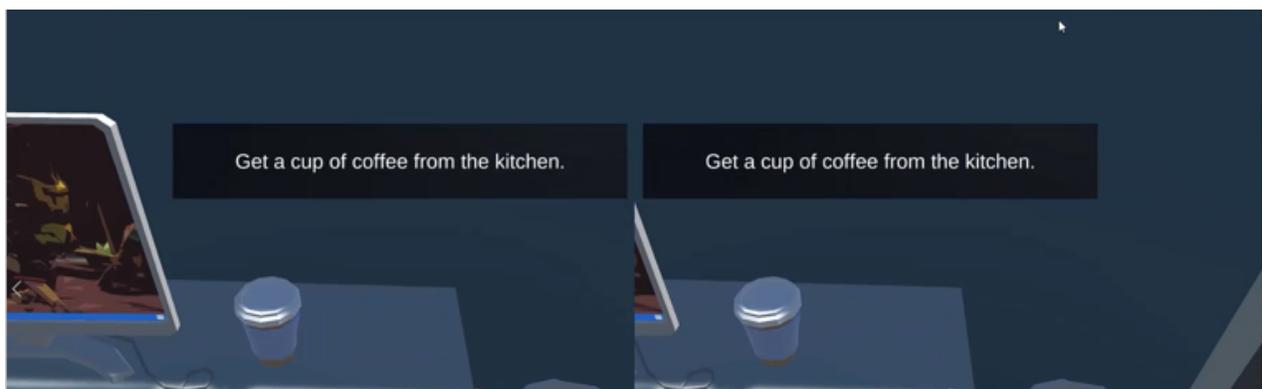


Figure 8: Task HUD displaying the user's active task

3.5 Exiting the System

The experience ends in the *Reflection* scene, with a parting message from the developers. Once this message is displayed, no more messages will display when the user presses the primary button (see *Using the System*). At this point, the user may remove the headset, and the experience may be stopped by clicking the *Play* button in Unity.

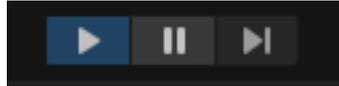


Figure 9: Unity scene play button turns blue when active

The user can exit the system at any time by following these steps, however it should be noted that the user's progress will not be saved, and they will have to start the experience over from the beginning.

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 Home, Office, and Reflection Scenes of the experience.

4.1 Home Scene

The *Home* scene is used to get the user used to interacting with objects in the virtual environment. It contains a series of instructions that the user must complete in order to proceed to the *Office* scene.

The first of these instructions is “**Teleport to the dresser by the window.**” To do this, the user must hold down the secondary button on the right-hand controller. This will show the teleportation ray. The user can teleport by pointing this ray at the desired location, and pressing the trigger button on the right-hand controller.



Figure 10: User teleporting using a ray-caster bound to the right-hand controller

Next, the user is directed to, “**Pick up the Rubik’s cube on the dresser**”. To pick up objects, the user must simply reach out a hand toward the desired object so that the hand is close to it, then press the grip button on the controller of the hand that is out. While the grip button is held down, the object will be grabbed by the user.

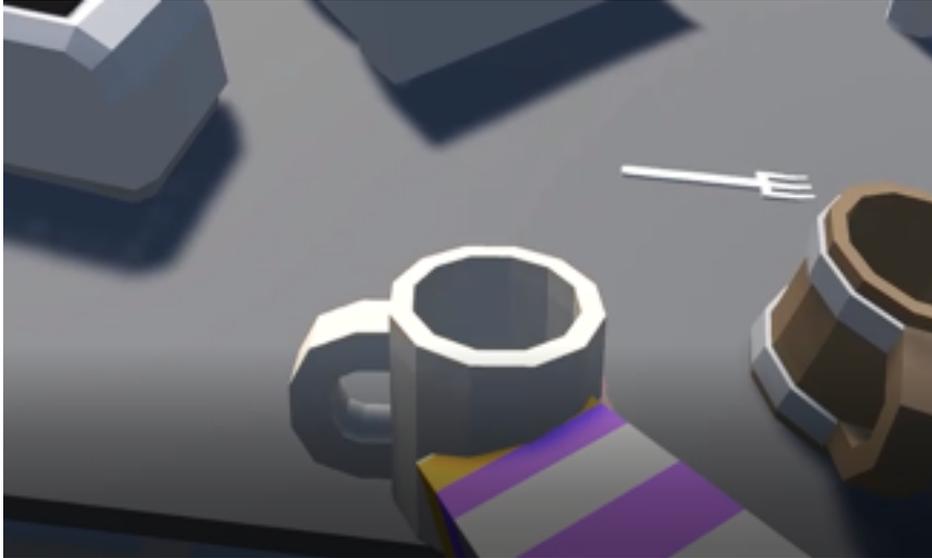


Figure 11: User picking up a coffee mug

Lastly, the user is asked to, “**Open the door and move into the hallway**”. To open doors, users must simply reach out a hand close to a door and press the primary button on the controller of the hand reached out. Once the door is open, the user simply must teleport into the hallway.

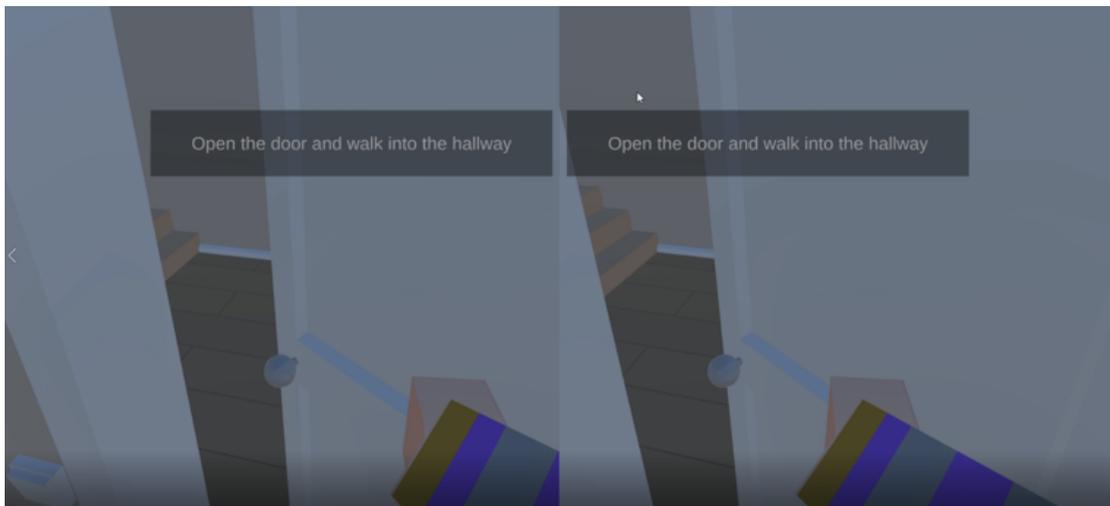


Figure 12: User opening a door in the Home scene

Once the above three instructions have been completed, the user will then be free to move to the *Office* scene by opening the front door of the house, located directly to the user’s right as they leave the first room. However, users are free to explore the house environment further if they would like more time getting used to the virtual environment and the mechanics of the experience.

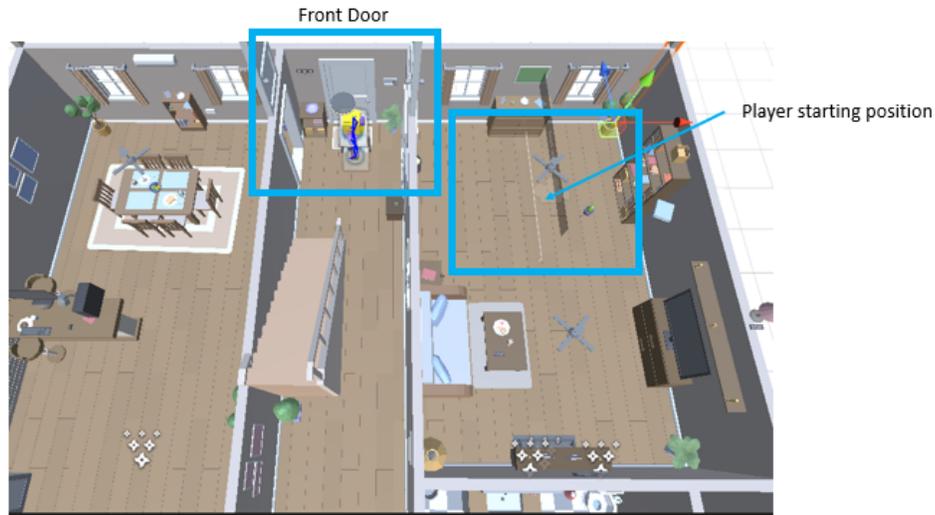


Figure 13: Location of the front door of the Home scene

4.2 Office Scene

The *Office* scene is the core of the experience. It is where the user will experience ADHD firsthand.

Similarly to the *Home* scene, the *Office* scene assigns instructions to the user which are to be followed to advance the experience.

The first instruction is for the user to, **“Walk into the manager’s office and read their task for the day from the job board”**. The manager’s office is directly forward from where the user is dropped into the environment.

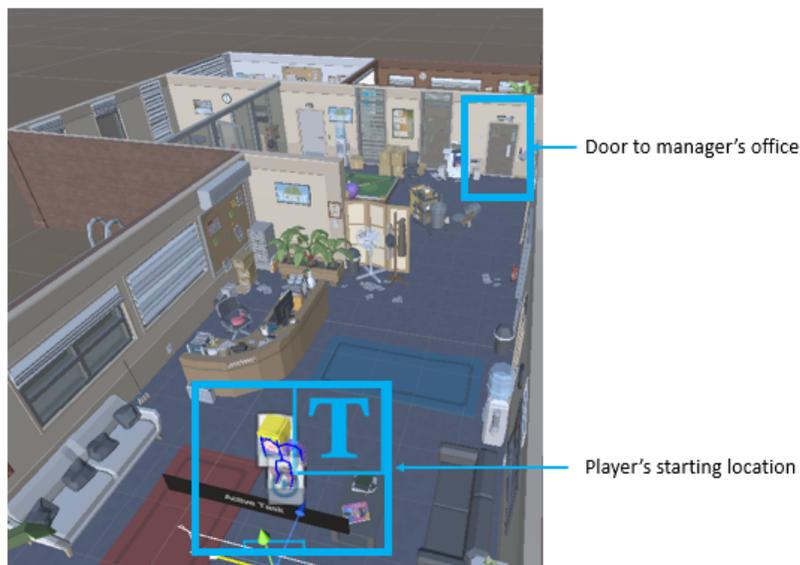


Figure 14: Location of the manager’s office with respect to the user’s starting position

The user must teleport to the office, open the door, and teleport inside. Once inside, the user must teleport closer to the job board behind the manager's desk and read the board to find their task for the day. The instructions are written for the names of the office workers, so the user must remember the name of their character. The name of the character in the ADHD experience is Alex.

Once the user has found their task, they must move to their desk. Their desk is located according to the following image:

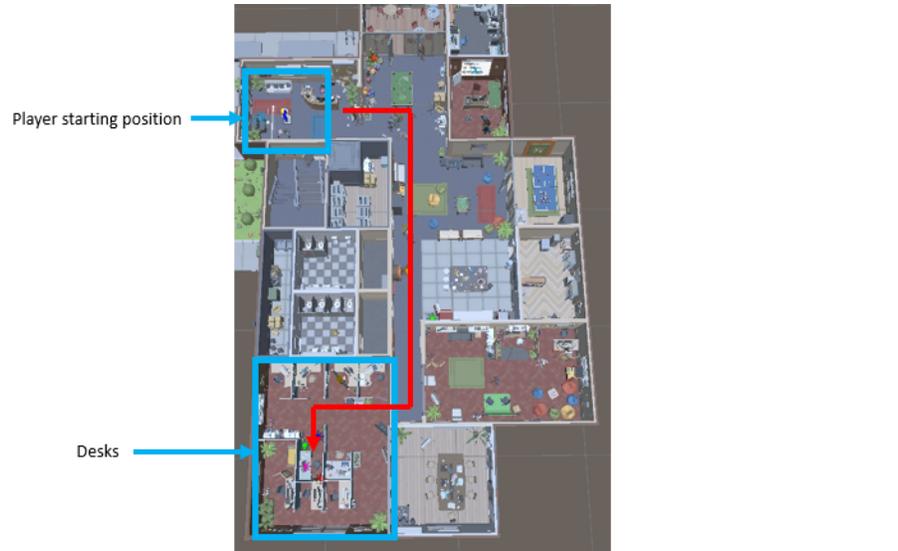


Figure 15: Location of the desks with respect to the player's starting position

The desk is identifiable by a disposable coffee cup and a file on the desk.



Figure 16: Report on the user's desk

If the user successfully reads the job board, then they will know to read the report. To do this, the user must grab the report. When they pick up the report, the text of the report will appear, along

with some distraction mechanics (lights and sounds) that are an attempt to simulate ADHD symptoms.

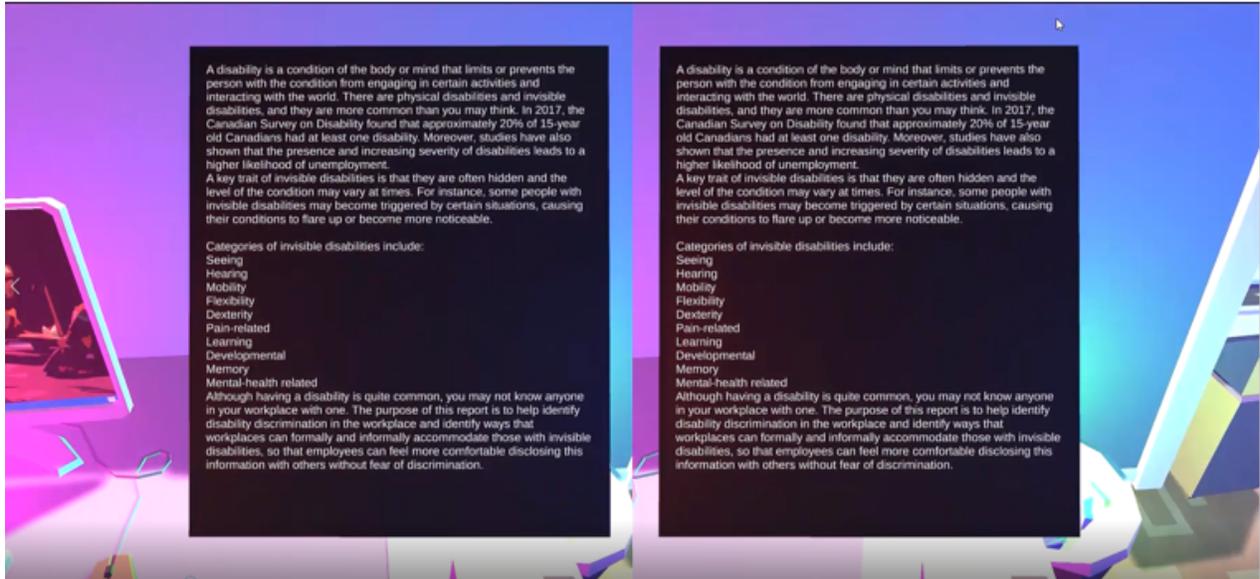


Figure 17: User's view with when the report is picked up

When the report is initially picked up, an inner-monologue audio clip will play. At the end of this clip, the next task is assigned, which is to “**Grab a cup of coffee from the kitchen.**” To complete this objective, the user should make their way to the kitchen (they will have passed it to get to their desk) and grab one of the three mugs on the counter.



Figure 18: Usable coffee mugs on the kitchen counter

Picking up on the three mugs will trigger the next task, which is to “**Clean up the kitchen by placing dirty dishes in the sink**”. To complete this task, the user must place 5 of the dishes found on the counters in the sink (excluding the three mugs). When an item has been successfully placed in the sink, it will disappear.



Figure 21: Conversation with co-worker

Because of the character’s ADHD they are unable to follow what the co-worker asked them to do, and they will return to their task of reading the report. The active task will then return to **“Continue reading the report”**.

Shortly after they begin reading their report, the character will remember that they’ve forgotten their coffee in the kitchen. The active task will then be **“Retrieve your cup of coffee from the kitchen.”** To complete this task, the user must return to the kitchen, grab one of the three mugs, and bring it back to their desk.

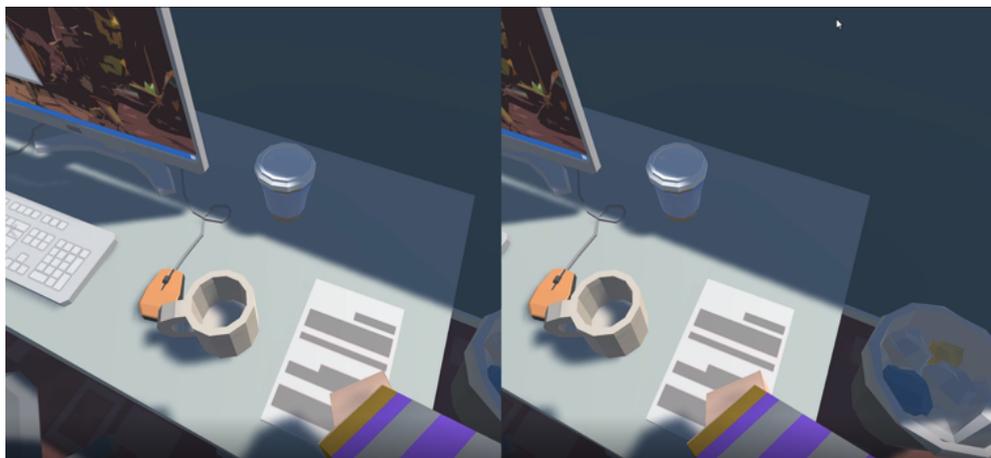


Figure 22: Coffee cup on the player’s desk

At this point, the user will be task with once again return to their report with the instruction, **“Finish reading the report.”** At this point, the user must pick up the report once more and do their best to make their way through it.

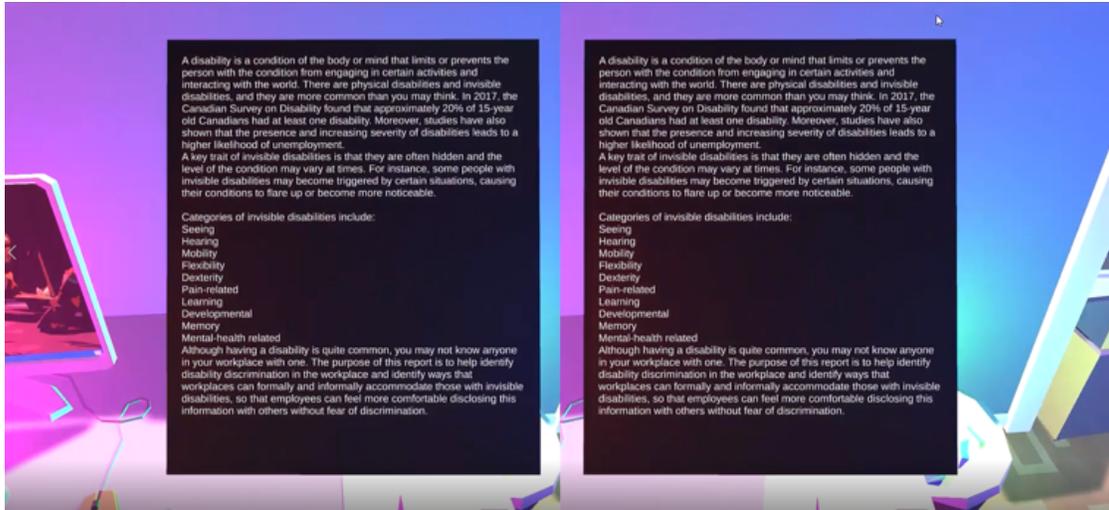


Figure 23: User's view while reading the report

After so much time, the character will notice the time and that it's the end of the day. The active task will change to the final, which is to **“Go report your progress to your manager in their office.”** The user must then put down the report and make their way back to the manager's office where they first found their task. Upon entry, they will notice a character in the office. Once close enough, the character (their manager) will engage them in conversation. The user is to sit and listen to the conversation.



Figure 24: Conversation with manager

When the conversation is finished, the user may leave the office to end the scene and move to the *Reflection* scene.

4.3 Reflection Scene

The reflection scene serves to enhance the emotional education of the user. It is very passive, requiring little input from the user.

Upon the end of the *Office* scene, the user will be brought to the *Reflection* scene. It is simply a room with four walls. The wall in front of the user displays text for the user to read. This text contains context and reflective questions for the user to ponder.

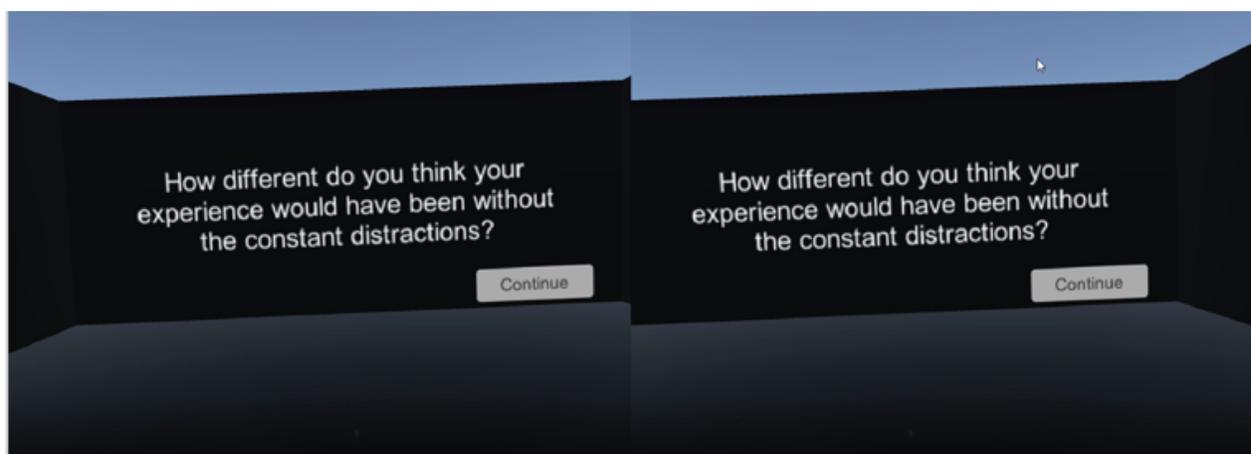


Figure 25: Reflection scene

Once the user is finished reading the wall of text, they must press the primary button on either controller to advance to the next chunk of text. When they have reached the end of the text and no more text appears upon pressing the primary button, they have completed the experience and can exit.

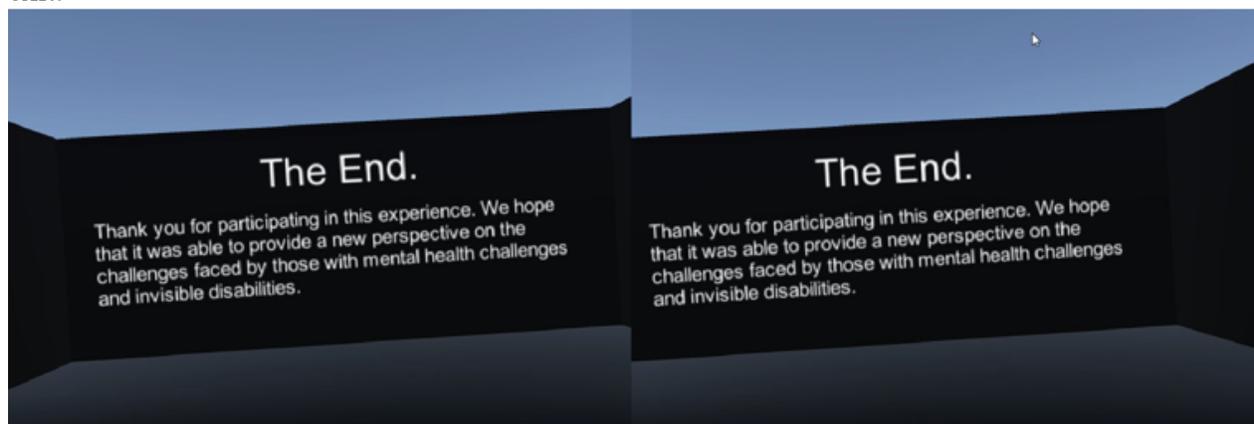


Figure 26: Ending text in the Reflection Scene

5 Troubleshooting & Support

This section will describe all recovery and error correction procedures, including error conditions that may be generated and corrective actions that may need to be taken.

5.1 Error Messages

Unity is a large and complex software and may produce error messages unknown to the developers. Upon the appearance of an error message in the console window of the Unity editor, it is best to look up the cause of the error on the internet. That said, there are a few warning messages that may appear in the console window that can be ignored, and there are a few experience-breaking behaviors that can arise through actions of the user.

Ignorable Warning Messages:

- “The referenced script on this Behavior ... is missing.” - This is a vestige of the assets purchased in asset packs. It is referring to behavior not utilized in this experience.
- “A Collider used by an Interactable object is already registered with another Interactable object” - Another consequence of using prefabs from an asset pack. The modifications made by the development team is overriding pre-packaged behavior from the asset pack which is no longer necessary.

Known Problematic Behaviour:

- The VR headset should not be removed during playthrough of the experience. Doing so may induce the warning message, “XRStats::SetStatFloat failed”, and will freeze the experience. The only way to fix this is to restart the experience.
- The transition from the Home scene to the Office scene takes some time. The experience will appear to be frozen. Do not remove the headset or close the experience. Wait for the scene to load, and the experience will once again behave normally.

6 Product Documentation

This section will explain in detail how the final prototype was built including design considerations and calculations.

6.1 Top Level View

The VR experience is logically divided into a series of different scenes. Each scene serves its own purpose to the experience. For example, the *Home* scene serves to introduce users to the mechanics of the experience. The scenes flow into each other, each containing a trigger that will move the user to the next scene, as shown in the project overview section of this manual.

Note that the *Introduction* and *Character Selection* scenes have not yet been fully implemented and are not part of the experience as of publishing of this manual.

6.2 General Scene Structure and Components

6.2.1 Models, Prefabs and Visual Construction of the Scene

Most of the scenes have been built using purchased visual and audio assets. Demo scenes from the purchased asset packs were modified to be used as the *Home* and *Office* scenes. Assets from these packs were added to the scenes to alter their designs to better match the desired theme. Character models were imported into these scenes from an asset pack.

The *Introduction*, *Character Selection* and *Reflection* scenes are built using Canvas elements from the Unity toolkit (the *Character Selection* scene would also use character assets from the purchased asset packs if it were finished).

6.2.2 XR Interaction Toolkit

The experience uses the *XR Interaction Toolkit* plug-in available from the Unity Package Manager. This plug-in aids in the development of virtual and augmented reality games and experiences by providing an abstract interface to VR and AR devices. It provides several functions for capturing input and output from the hardware devices, including button inputs and motion tracking. The plug-in also interfaces with several VR and AR devices, so that the experience input managers only need to be programmed once for compatibility with several devices. In practice, the plug-in greatly simplifies programming the experience for VR devices.

6.2.3 XR Origin

The base element of the user in all the scenes is the *XR Origin* game object. It is a standard element in the XR Interaction Toolkit and contains the elements for interfacing with VR/AR hardware. XR Origin contains the experience's main camera and controllers.

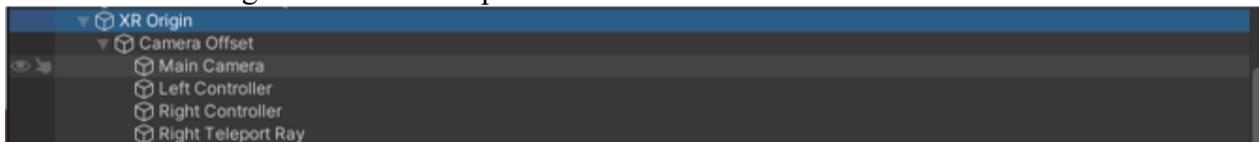


Figure 27: XR Origin game object hierarchy

6.2.3.1 Main Camera

The main camera is the user's view into the experience. The game object contains a script that binds the movements of the user's VR headset with the movements of the camera, so that as the user moves their head, the camera moves with it (see *TrackedPoseDriver* in Appendix_ - Scripts for more details).

6.2.3.2 Left and Right Controllers

The *Left Controller* and *Right Controller* game objects are the Unity representations of the user's hand controllers. They are copies of each other, with slight modifications to match their side. The controllers contain a script, *XRController* to track the movements of the controller and monitor their inputs (e.g., triggers, buttons). They also contain a script, *XRDirectInteractor* that implements the grab-functionality of the controllers (i.e., ability for users to pick objects up in the experience). The two scripts are standard XR Interaction Toolkit components. The controllers also contain a Sphere Collider, which is necessary for *XRDirectInteractor*. The sphere collider represents volumes around the hands in which objects can be grabbed.

There also exists a game object, *Right Teleport Ray*. This object is another controller, implemented through the right-hand controller (hardware). This controller is for a ray-interactor which is used for teleportation of the user around the environments. This is a standard component in XR Interaction Toolkit and contains several fields that can be tailored to the desires of the developer. See the *Teleportation* section for more details.

6.2.4 **Character Model and VR Constraints**

To enhance the experience and aid the user with interacting in the environment, a character model has been bound to the XR Origin. The character model used in this iteration of the experience is named *SimplePeople_HouseWife_White*. The *SimplePeople_HouseWife_White* game object contains a child game object entitled *VR Constraints*. This child is responsible for binding the movements of the headset and hand controllers worn by the user to the movements of the character model.

VR Constraints contains several different child objects. *Right Arm IK* and *Left Arm IK* contain *Two Bone Constraint IK* components, a standard component in the Unity library. This component aids in more naturally animating skeleton-based prefabs. For the purposes of this experience, the component ensures that when the hands of the character model move, the arms bend at the elbow and follow the movements of the hands, instead of the hands simply translating on their own. The same is true of the *Head Constraint* object. The *Target* and *Hint* child objects are respectively bound to the hand and forearm bones of the character model prefab, and are used by the script *VR Rig* attached to *SimplePeople_HouseWife_White*. It is this script that binds the movement of the controllers and headset with that of the character model.



Figure 28: VR constraints game object hierarchy

6.2.5 Teleportation

In this experience, locomotion is achieved through teleportation. It is a standard design choice used to minimize motion sickness in virtual reality. It is implemented with the game objects *Teleportation Area*, *Right Teleport Ray* (inside XR Origin), and the script *TeleportationProvider* (part of XR Origin).

6.2.5.1 Teleportation Area

This game object contains a list of all the objects which can be teleported to, which contains all of the floor, rug, carpet, stairs assets in the scenes. It is a standard component of the XR Interaction Toolkit.

6.2.5.2 Right Teleport Ray

This game object is a standard component of the XR Interaction Toolkit. It contains the visual ray wielded by the user to direct their teleportation, and components for modifying its properties.

6.2.5.3 Teleportation Provider

This script is responsible for the teleportation of the XR Origin (i.e., the user) and anything attached to it (i.e. the character model). It is a standard component of the XR Interaction Toolkit. See documentation of the XR Interaction Toolkit for more details.

6.2.6 Managers

Managers are high-level scripts that manage the flow of the experience. There are two managers present in each scene: the Game Manager, and the Player Manager.

6.2.6.1 Game Manager

The Game Manager controls the overall flow of the scene. It monitors the events that occur and responds accordingly. For example, the *OfficeSceneManager* of the *Office* scene contains the list of all tasks that must be done and assigns them when the *ADHDPlayerManager* reports to it that it has completed a task. The manager of the *Home* scene performs a similar function. There are currently three Game Managers in the experience: *Game Manager (Home scene)*, *OfficeSceneManager (Office scene)*, *ReflectionSceneManager (Reflection scene)*.

6.2.6.2 Player Manager

The Player Manager manages the state of the player, keeps track of important information regarding the player, and interacts with other managers to trigger events. There are only two player managers in this version of the experience: *PlayerManager (Home scene)*, and *ADHDPlayerManager (Office scene)*. In the *Home* and *Office* scenes these player managers keep track of all the player's outstanding tasks and notifies the scenes' game managers when a task is completed so that the game managers know to assign a new task to the player.

6.2.7 User Interface

The user interface acts as a heads-up-display (HUD) for the user so that they can always see what their active task is.



Figure 29: Heads Up Display displaying the user's active task

The interface is built with Canvas, a built-in UI object in Unity. The Canvas contains and controls all the elements that make up the HUD. The HUD Canvas contains a *Text* object which shows the desired text, and a *Panel* object which serves as a backdrop on which the text is displayed to assist with visibility.



Figure 30: HUD Canvas and child objects

The *Text* object contains a script, *UpdateTask* which updates the text displayed whenever a new task is assigned.

In the *Office* scene, the user can display the HUD whenever they like by pressing the primary button on either controller. This is controlled through a script entitled, *HUDControl* attached to the UI Canvas. The script simply listens for the press of the primary button and enables or disables the HUD accordingly.

6.2.8 Interaction Scripts on Assets

To make the experience more interactive, several assets within the environments contain scripts to either trigger the completion of tasks, trigger animations, or trigger other behaviors within the experience. See below for a complete list of behaviors triggered by scripts attached to objects within the environments.

6.2.8.1 Doors Opening

In the *Home* and *Office* environments the user can open and close doors by moving their hands near to the door and pressing the primary button on the hand controllers. This is achieved through a script attached to the door assets in the environments, entitled, *DoorHandleTrigger*. The script modifies the *OnTriggerEnter* function of the collider components of the doors to set a Boolean to true when the collider of the player's hands enters that of the doors. When this Boolean is true, the script will listen for the press of the primary button on the hand controller. If this button is pressed, then the script will trigger the door open animation of the doors. If the door is already open, it will trigger the door close animation.

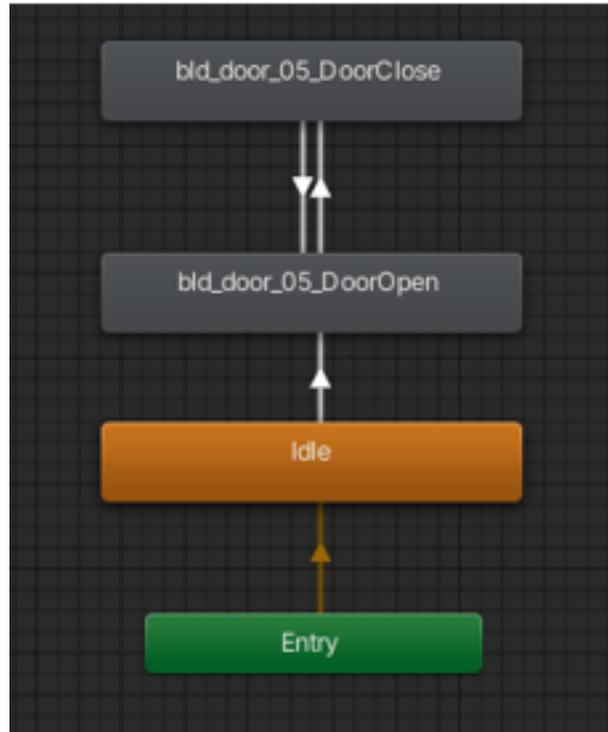


Figure 31: Door animator

6.2.8.2 Grab Interaction

Grab interaction is implemented with two components. The first is the *XRDirectInteractor* component on the hand controllers that allow for direct interaction with the controllers (see section *XR Origin* for more details). The second, is a standard component script from the XR Interaction Toolkit entitled, *XRGrabInteractable* which needs to be attached to all objects which should be grabbable. This script is what controls the translation and rotation of objects when they are picked up by the user.

Note, for *XRGrabInteractable* to work, objects must also contain a Collider component, and a Rigid Body component.

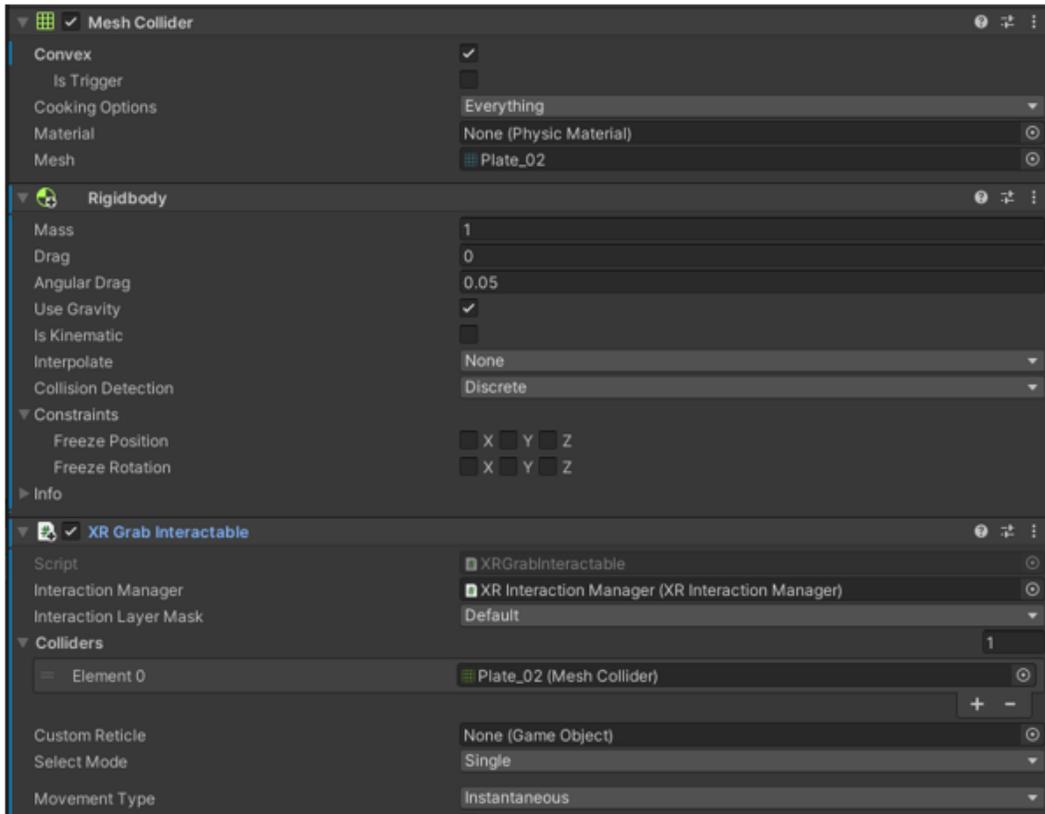


Figure 32: XR grab interactable and other components necessary for objects to be grabbable

6.2.8.3 Task Completion Triggers

Player completion of tasks is primarily detected by checking if the user is in the correct spot, or if they have picked up the right object. This can be achieved by detecting if the user’s hands are within the collider of the object being picked up, or in the case of location-based task completion, if the user’s hand colliders are within some collider box placed within the environment. In either case, the game object containing the collider has been given a script that modifies the object’s collider’s *OnTriggerEnter* function to detect when the user’s hand collider has entered the collider. From here, the script can go two ways. If the task completion requirement is location-based, then the user’s hand collider being within the location collider is enough to trigger the completion of the task. If some sort of interaction is required for task completion, then the script will contain a listener for some button on the controller, usually either the grab button or the primary button. When the user is within the location-collider, or when they interact with the object, then the script will check the active task. If the active task is the task specific to the collider in question, then the script will mark the task as completed to the Player Manager.



Figure 33: Example of a collider used to detect task completion

6.2.9 NPC Interaction

6.2.9.1 Co-worker Interaction

As part of the ADHD experience, a coworker approaches the user at their desk and engages them in conversation. A script attached to the coworker NPC game object's parent game object controls this behavior. The script, *MoveToPlayer*, waits for the completion of the task that returns the player to the desk. Once this occurs, the NPC translates through a path set by a series of predetermined points located in the office. These points are simply empty game objects. At the same time, the script triggers the NPC prefab's "walk" animation. Once the NPC reaches the last point, the script plays an audio source which contains the conversation.

The script also contains code to control some ADHD distraction mechanisms. See the *ADHD Distraction Mechanisms* section for more information.

6.2.9.2 Manager Interaction

At the end of the experience, the user must return to the manager's office and report their progress to their manager. However, at the beginning of the experience, the manager is not present in the office. Attached to the character model representing the manager is a script, *TalkToPlayer*, which controls this end of experience behavior. The script listens to the ADHDPlayerManager for the completion of the last work task. When the task is complete, it spawns the manager prefab in the office. The script then begins to listen to a dedicated task-completion-detection collider box that is in the manager's office. When the collider box is triggered by the player entering the object, the

MoveToPlayer script plays an audio source containing the player's conversation with the manager.

6.2.10 Player's Report

The player's main task of the experience is to read a report for their manager. The implementation of the report is a combination of a paper prefab asset, a canvas UI system, and a script. The paper prefab asset is a visual representation of the report that the user can interact with. It is placed on the user's desk. The canvas system is like the previous ones described. It contains the text of the report, laid out in a way which should be readable by the user.

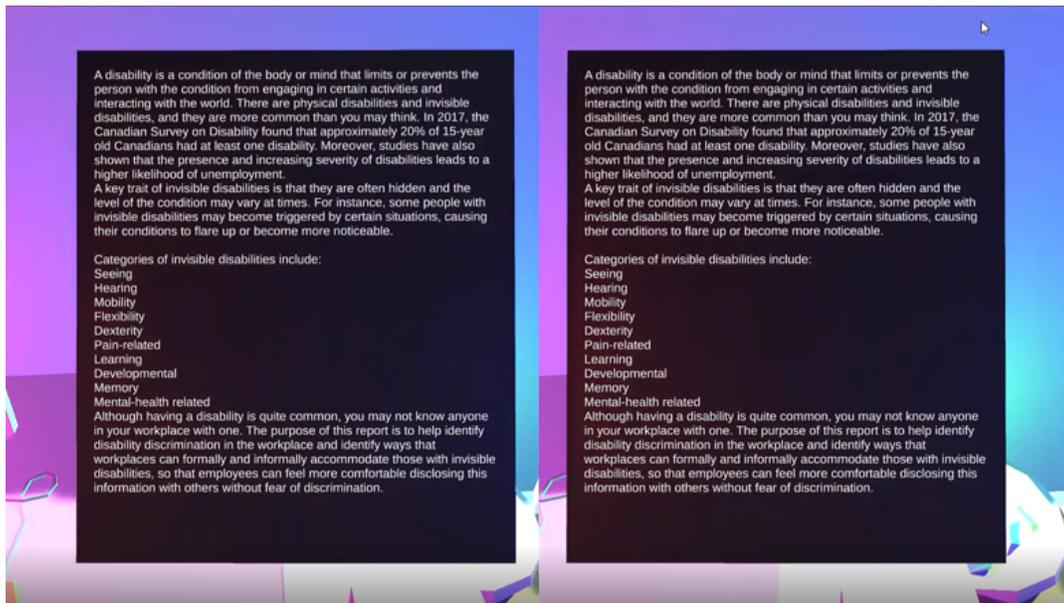


Figure 34: Report to be read by the user as their main task

The canvas follows the movements of the main camera, and so the text will move with the player. The script *ReadReport* - attached to the paper prefab - listens for the user picking up the report. When picked up, the script enables the canvas object containing the report. When the report is put down, the canvas object is disabled. *ReadReport* also contains functions for controlling distraction mechanisms, as described below.

6.2.11 ADHD Distraction Mechanisms

6.2.11.1 Distracting Lights

The distracting lights mechanism is made up of four light objects and code to control their behavior. The code is embedded in the scripts that dictate the behavior of game objects which should trigger the distracting behaviors, namely the Player's Report and the co-worker that engages the player in conversation. This code simply enables/disables the lights when appropriate, and modifies their colors on a sliding scale, producing a distracting rainbow-like effect.

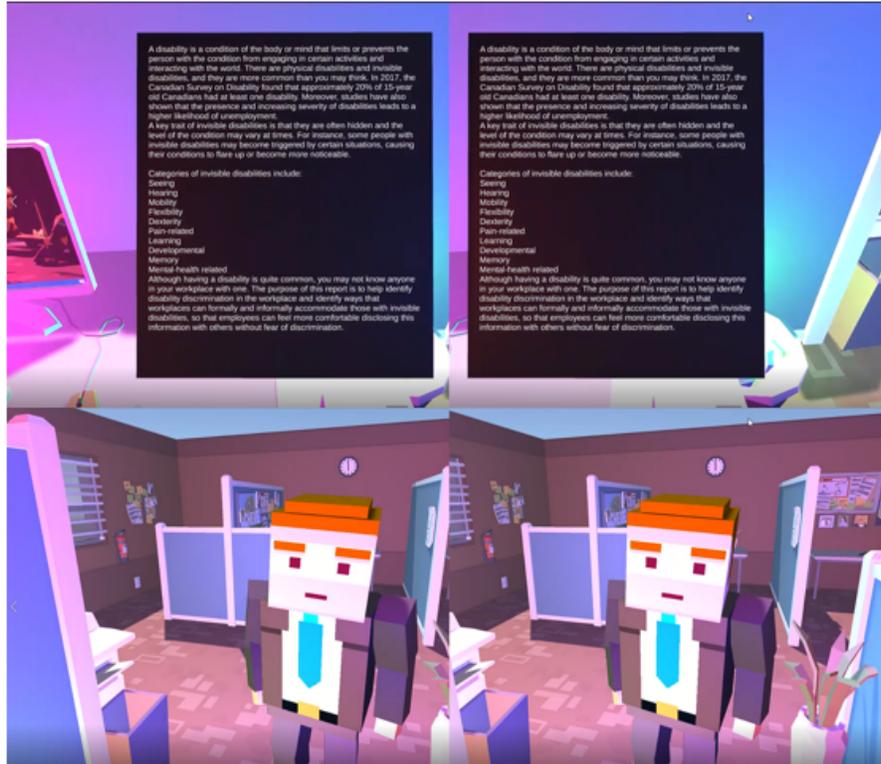


Figure 35: Distracting lights mechanism active when reading the report or talking to the co-worker

6.2.11.2 Distracting Audio

The distracting audio mechanism is like the distracting lights mechanism in that it is controlled by code embedded in the scripts that control the behavior of game objects which should trigger the behavior, again, namely the Player's Report and the co-worker. This script plays and stops an audio file as appropriate that contains a distracting sound.

6.3 Testing and Validation

All the above components and features have been tested through completion of the experience. They are functional as built, and when used according to the instructions laid out in this report. Deviation from these instructions may induce unwanted behavior, as the system has not been robustly tested and hardened against undescribed behavior.

6.4 Design Choices

6.4.1 Sound Design

In virtual reality (VR), audio accounts a significant amount of the experience. So, the right sound can provide a more realistic, immersive feel to VR applications. Audio is critical to completing a truly immersive experience in VR, not only conveying depth and emotion, building,

and enhancing interaction, but also guiding users to unfamiliar landscapes. Unity3D, a free 3D development engine for building games, simulations, and experiences, is the easiest way to make VR applications. Sound improves the user interface and user experience in many ways: conveying information to the user, input (confirm, cancel, return, previous, next, etc.), integrating sound, presenting a catalog of experiences, enabling interactions that were previously difficult to achieve, making the experience more familiar, etc. To guarantee these purposes, we designed to insert guiding sounds in different scenes and interfaces, as well as interrupting and advancing the plot with inner monologues and character dialogue interactions as the user's tasks progress.

6.4.2 Art Design

The art in VR is an important factor. The first is reflected in the first impression given, both in the interface as a whole and in the local art of the interface. Our interface is brightly colored so that it blends in naturally and doesn't feel the least bit abrupt, and the colors will reflect sudden changes used to enrich the transitions. In addition, we have applied some simple 3D technology, which relies heavily on Unity support. Of course, all our artwork is implemented in Unity.

6.4.3 Gameplay

The game subsystem includes all user movement and interaction with the world. The scenes in the game are mainly divided into home scenes and office scenes, where the user is assigned tasks that must be completed. In the development of the software, users will be able to control their movement in certain scenarios, interact with objects, and select dialogue options in conversations. Eventually, after all assigned tasks are completed, the user will receive the reflection interface, which we hope will enhance the empathic experience of the user by prompting guided feedback. Also, because the software was developed primarily for ADHD, we set it up to require the user to "sit and listen" to characters while being distracted.

6.5 BOM (Bill of Materials)

Table 3. Bill of Materials

Name	Description	Cost	Link
Low Poly Cartoon House Interiors	Low poly house themed asset pack including a demo house scene.	\$9.99 USD	https://assetstore.unity.com/packages/3d/props/interior/lowpoly-cartoon-house-interiors-167425
POLYGON Office	Low poly office themed asset pack including character models, objects, and office demo scene.	\$39.99 USD	https://assetstore.unity.com/packages/3d/props/interior/polygonoffice-low-poly-3d-art-by-synty-159492

Simple People -Cartoon Characters	Polygon character model assets	\$9.99 USD	https://assetstore.unity.com/packages/3d/characters/humanoids/simple-people-cartoon-characters-15126
Total Cost		\$59.97 USD	

6.6 Equipment List

This VR experience has been developed to support all major virtual reality hardware platforms (e.g. Oculus Rift, HTC Vive, Valve Index.)

6.7 Testing & Validation

All the above components and features have been tested through completion of the experience. They are functional as built, and when used according to the instructions laid out in this report. Deviation from these instructions may induce unwanted behavior, as the system has not been robustly tested and hardened against undescribed behavior.

7 Conclusions and Recommendations for Future Work

Invis-Ability is an immersive experience tool that is easy to use, test, and maintain. Already outlined in this manual is all the documentation needed to understand and use the implemented features in the final prototype. As it is understood that other users may wish to continue and improve upon this prototype, this section outlines some suggestions for future work and summarizes the lessons learned related to the work completed throughout the semester.

Due to time constraints, many desired features had to be simplified or removed to meet the deadlines for this project. These features included accessibility options such as subtitles, more distractions such as blurry text, and the ability to have dialogue choices when talking with a virtual character, such as the player's boss or co-worker.

In the future, it may also be desired to add another invisible disability, such as Generalized Anxiety Disorder (GAD) to the prototype. This was something that the project team had planned on adding; however, it was not implemented due to time constraints and feedback from the client. The basic storyline for this experience has already been developed and is included in Appendix II. If this was to be done, it would also require the creation of a character selection screen. A UI drawing of this concept is also provided in Appendix II.

Since the client stressed the importance of using VR for this project, the prototype was developed for an Oculus Rift headset. However, if future clients want to use this software on another medium, such as a desktop computer, mouse and keyboard controls would need to be implemented. This would make the product more accessible to a large variety of users.

Overall, the problem presented by the client and the creation of Invis-Ability allowed the project team to learn many important lessons. Firstly, the application of the different stages of the design thinking process, empathize, define, ideate, prototype, and test, was learned. Additionally, the team's problem-solving, time management, and conflict resolution skills improved significantly through collaborating for months on this project. The importance of giving and receiving high quality feedback was also learned as feedback was sought from potential users during the prototyping process. Since the project focused on empathy, it allowed the project team to gain a valuable insight into invisible disabilities and their impact on people's everyday lives. Finally, the use of the Unity software taught the team new coding skills.

The final prototype is available through the Makerepo repository for Invis-Ability and is also included in Appendix I.

8 APPENDIX I: Design Files

This section includes all documents referred to in this user manual and required for this project in table 3 below.

Table 4. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
Invis-Ability - MakerRepo	https://makerepo.com/hhinc076/1157.gng1103f-team-1-1-invisability	April 8 th , 2022
Project Proposal - Immersive Experience Tool	https://makerepo.com/project_proposals/268	March 9 th , 2022
Needs Identification and Problem Statement	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/se5ozbjhr8n79k3q4yxjd7wokpl?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20B%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192709Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=5168ec5bc5e727f1f1fcb81ec0f36c8a5fac2b1f1fe1675ee11db1af6448b38f	March 18 th , 2022
Design Criteria and Target Specifications	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/z3ssot1ci7pcrn9qig9pbdvzr48ua?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20C%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192737Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=c4ed31a61433e9f3f8ed9ab79633cde5ace1480eb319105a711401fa97bae9d	March 18 th , 2022
Conceptual Designs	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/un9nwpg0w0j2627c1g3eisenkfhm?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20D%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192752Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=b5dc29e7d6c0db0c3e30ba77a89d770f2566d5d7db045f0944aba753dda475bc	March 18 th , 2022
Project Plan and Cost Estimate	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/kf8u9vbyyzkhorlq4udm0gdf44?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20E%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192806Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=c122556750e153696ce50c34ea4e5af38a839c39da9a37addf7577520c39f20	March 18 th , 2022
Prototype I and Customer Feedback	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/hkxkf5khu25oz20uc2mcyw2kydyx?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20F%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192823Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=0336f34abefb8c0aa9a3fc8d92ec0347e274ed67462fcc20cc6b274ea6a1186	March 18 th , 2022
Prototype II and Customer Feedback	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/qutrdykp48i118hpdqwrq86snfc?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20G%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192837Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=71862601a2296abc6758af10b8ac34dbaca46bd06ab042bec0d787ba19ab97f7	March 18 th , 2022
Prototype III and Customer Feedback	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/k8gzaqv47seqlrbo8msr13ozh5k?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20H%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192848Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=0a505847d3620e72d9ba8931af83cabd6f207e1cbe1cb8431fb9bcd938508fc	March 27 th , 2022
Pitch Presentation	https://uottawa-makerspace.s3.us-west-2.amazonaws.com/izattschrsrvybtq6f2ifsg7id?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20I%20-%20Design%20Day%20Presentation%20-%20Team%201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fus%2Faws4_request&X-Amz-Date=20220408T192910Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=6f5f99e158de0e31bf9b92070a7eba7cdd0d36e832e0b77e0e9adcfbb03dad6	March 31 st , 2022

<p>Class Presentation</p>	<p>https://uottawa-makerspace.s3.us-west-2.amazonaws.com/o7pecq11cl3zhuvmmlh7ngdamfai?response-content-disposition=inline%3B%20filename%3D%22Deliverable%20%20-%20Class%20Presentation%20-%20Team%201.1.pdf%22%3B%20filename%2A%3DUTF-8%27%27Deliverable%2520P%2520-%2520Class%2520Presentation%2520-%2520Team%25201.1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA34PB766H365OOAG6%2F20220408%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20220408T192926Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=39d3f97520d81a247928b07575278c94d91d5af179b595f3cb30b3244c6039ea</p>	<p>March 31st, 2022</p>
--------------------------------------	--	---