Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Immersive Art With Natural Interaction In Mixed Reality

Xiangpeng Fu, BAI

August 2022

A Dissertation submitted in partial fulfilment

of the requirements for the degree of

Master of Science in Computer Science (Augmented and virtual reality)

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Xiangpeng Fu

August 18, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Xiangpeng Fu

August 18, 2022

Immersive Art With Natural Interaction In Mixed Reality

Xiangpeng Fu, Master of Science in Computer Science
University of Dublin, Trinity College, 2022

Supervisor: Mads Haahr

Abstract

This Master's Thesis intends to implement an immersive art prototype that allows users to be immersed in a mixed reality environment full of flowers. This prototype is based on Oculus quest 2 head-mounted display(HMD), Zed Mini Stereo Camera, and Leap Motion Controller. Leap Motion allows users to interact with virtual objects and interface freehand.

The three critical steps of this project are the setup of the Oculus quest 2 HMD, Zed Mini Camera, and the Leap Motion Controller, the design and implementation of the immersive art content and interaction method, and the discussion of user immersive experiment usability study of natural interaction.

Keywords

Immersive art, mixed reality, natural interaction, Oculus quest 2, Zed Mini Camera, Leap Motion

Acknowledgments

First of all, thanks to my supervisor Mads Haahr, for his responsible attitude. He read a lot of my revisions and helped to understand my confusion. Secondly, I would like to thank all the teachers I met during the year for their earnest teaching, which improved me academically and as a person. In addition, I would also like to thank my parents for supporting me, and allowing me to move forward carefree on the road of chasing my dreams. I just stepped on their shoulders and saw the prosperity of this world. Finally, I would like to thank this beautiful era for allowing me to pursue my dreams. There is a long way to go, and I will go up and down to search for it. There is no end to learning and a long way to go. I will be what I want to be.

XIANGPENG FU

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The first chapter introduces the concept of immersive art in mixed reality, as well as the main difference between different interaction methods. Then I discussed the main goals and sub-goals of this dissertation. Last, the dissertation structure is listed.

1.1 Motivation

We create new technological forms and media artworks in the digital age that connect people and the environment. Compared to traditional art, the newly popular "immersive art" has become increasingly popular due to its technology, interaction, and fun advantages.

Immersive art primarily referred to digital means of modern technology as well as various forms of multimedia display based on traditional art and used to display and disseminate new art forms. It is a combination of plastic art, visual art, and new media art. The resulting immersive exhibitions are a new and specialized exhibition genre that integrates audiences in a three-dimensional reconstructed space to enjoy artworks and feel the illusion of space, time, and environment. In these innovative spaces, viewers are encouraged to use their senses other than sight to appreciate the content of the artwork and be fully immersed in the environments that artists create. Today, with the development of technology, new types of exhibitions are emerging and being introduced to a variety of audiences.

Mixed reality is a popular technology in recent years. Mixed reality (MR)[5], a technology that combines augmented and virtual reality, allows the digital world to be extended into the user's physical world. In contrast to virtual reality, which immerses the user in an artificial world, mixed reality runs in the user's real world. It enables tangible interaction with 3D virtual objects by allowing one to move and interact with virtual objects as if they were real objects in our physical world by moving a physical object or marker. As a result, there is a true tangible interaction between the physical and digital worlds.

Based on the above characteristic of MR, this master's thesis has treated to explore a way for immersive art to use MR as a carrier, allowing users to visually immerse into augmented scenes through a head-mounted display, and interact with virtual objects or user interfaces(UI) without any wearable sensors. So that the user's interaction with the scene can be considered "natural". This means a truly immersive experience that improves on any interactive immersive art currently known.

1.2 Goals

The objective of this master thesis is to study and implement an immersive artwork that enables users to interact with virtual and physical objects in this environment in a natural way. To achieve this project, it should include the following sub-goals:

1. All concepts and techniques involved will be analyzed. In this way, the main literature on immersive art and natural interaction could be included and can be applied to the project.
2. Learn and use the tools and equipment for building mixed reality environments: Oculus quest2 HMD, Zed Mini Stereo Camera, Leap Motion Controller, Unity 3D game engine, and related plugins.
3. Integrate each of these individual devices and tools into one environment and make it work correctly.
4. Design and implement the specific content and interaction methods of immersive artworks in a mixed reality environment built with the above devices and tools.
5. Evaluate and analyze the sense of immersion felt in the realized immersive artwork.

1.3 Thesis organization

This master thesis is organized as follows:

Chapter 1: Introduction

The introduction of main concept and goals of this this thesis.

Chapter 2: State of the Art

All of the literature and related works of this thesis.

Chapter 3: Setup Integration

Introduction to the installation of devices used in this project.

Chapter 4: Design and implemented

The development process of a naturally interactive immersive art demo.

Chapter 5: Evaluation

By inviting participants to subjectively evaluate the implemented demos to evaluate the immersion enhancement brought by natural interaction to immersive art.

Chapter 6: Conclusions and future work

A summary of this thesis and an analysis of aspects that can be improved in the future.

Bibliography

Chapter 2

State of the Art

This chapter contains all of the literature on the project that was researched. The most recent findings have been looked for on each topic in order to keep the information current, mainly including:

1. The literature on immersive art was explored. As a background to the research, this paper mainly conducts an in-depth study of immersive art based on augmented and mixed reality.

2. The natural interaction in immersive technologies was analyzed, where users can freely interact with their own hands.

3. The introduction and research of the equipment and tools used in this project, including their features and functions.

2.1 Immersive Art

To better understand this paper's proposal, it is necessary to demonstrate immersive art's progress and current development's limitations.

Immersive art is a new form of artistic expression. Immersive art emphasizes the application of interactive design. Compared with regular art exhibitions, immersive art often makes full use of human senses through the integration of technology, focusing on seeing, listening, touching, smelling, and participating, and by adding interesting content to stimulate and invite participants to connect with the environment created by the artist. Therefore, immersive art has attracted a large number of audiences.

The proposal and realization of the idea of immersive art depend on the development of technology. After going through the early stage of installation art and digital art form, it has reached the mature stage of immersive art and exhibitions[4]. For example, with the maturity of projection technology and interactive multimedia, more and more artists are integrating them with immersive art exhibitions, such as the artworks in the "Borderless" Museum created by the Teamlab art team led by Inoko kotobuki: "Forest of Flowers and People: Lost, Immersed and Reborn." It is to use computer technologies and systems such as laser projection and infrared photography to combine images with interactive technology and dynamic recognition technology to generate various gorgeous special effects, capture recognition participation the user's actions to make the user feel at one with the work.

In recent years, with the development of immersive technologies such as augmented and virtual reality(AR)(VR), because these technologies are highly related to immersion, they have provided a new carrier for immersive art. For example, the "Van Gogh: The Immersive Experience"[1] art exhibition organized by the Exhibition Hub uses 60 projectors to bring 200 Van Gogh masterpieces to life on a surface of 1,000 square meters. Projections animate the floor and all walls around the visitor, creating a great immersive experience. Visitors can use VR headsets to experience a day in the life of Van Gogh, allowing visitors to enter deeper into Van Gogh's mind.

Meanwhile, immersive art based on AR has also been well applied. The traditional AR research literature divides AR into two types handheld and wearable devices. Due to the more mature development of mobile phones and tablet computers, handheld devices currently occupy the dominant position in AR system development platforms. Currently, its application mainly uses handheld AR technology to display gallery artwork or museum content in a novel way.

For example, in the AR art exhibition hosted by Artivive[2], artists superimpose images, text, or sound on the artwork that people can already see. The user stands in front of the scene and holds up their smartphone or tablet, using the app to change the existing picture and add a third dimension to the display, which brings the object or scene to life.

Although using handheld AR devices as an immersive art carrier is now a mature concept, there are still problems. One is that users can only connect with the virtual environment through a small screen. The other is that AR superimposes virtual objects in the real environment. Users can only interact with them through limited buttons on the screen, which significantly reduces the user's immersion.

In order to solve the above problems, this paper considers using wearable mixed reality devices to enhance the user's immersion. On the one hand, head-mounted displays (HMDs) have been used to display virtual environment content. These systems render 3D computer graphics by placing small screens close to the eyes. They can track the head by

---

[1] https://vangoghexpo.com/

[2] https://artivive.com/augmented-reality-for-artists

incorporating external cameras so that when the user moves their head in the real world, the virtual environment responds accordingly. This is critical in designing augmented and mixed reality(MR) systems. The computer needs to know precisely where virtual objects are on the live video stream as the user moves. On the other hand, using mixed reality rather than AR technology takes advantage of three characteristics of MR systems[7]:

1. Combining the real-world object and the virtual object.
2. Interacting in real-time.
3. Mapping between the virtual object and the real object to create interactions between them.

Therefore, HMD is chosen to provide a wider field of view to enhance immersion, and the characteristics of MR are used for more realistic interaction with virtual and real objects.

2.2 Natural interaction

In both basic and applied research, a great deal of effort has been devoted to highlighting the importance of human-computer interaction (HCI) for end-user experience in AR and MR environments[8]. Furthermore, the event of real-time interaction is described as the aesthetic manner in immersive art by Meng Qu[9], aesthetic experiences can only be created when an actor interacts in augmented space, implying that interaction is the only way to obtain those experiences.

The interaction of MR applications requires users to operate MR virtual content in a three-dimensional (3D) environment. Thus, the 3D user interface is introduced to cover the interaction in a 3D environment. The 3D user interface is a human-computer interaction where the user's task is directly performed in a real or virtual 3D space environment. But this often requires the user to operate an input device, which can be bulky and obtrusive, preventing the user from effectively interacting with the 3D environment, so Natural User Interfaces (NUIs) are gaining popularity.

NUIs allow users to interact with little or no training, based only on their existing knowledge and use it without any distractions[13]. With the emergence of NUIs, the goal of human-computer interaction is to pursue that interaction with the computer will be as natural as human interaction. For this reason, the integration of gesture recognition into Human-Computer Interaction(HCI) is an important research field. Gestures have long been considered an interaction technology that can provide a more natural, creative, and intuitive way to communicate with computers[10]. As a natural medium for communication between humans, the hand, relative to other parts of the body, is the best tool for HCI[3].

Furthermore, according to Gabbard[6]'s classification of different interaction levels of virtual environments, in a highly interactive and high-fidelity mixed reality environment, the user is active in the scene and can modify the scene's content, and virtual objects in the environment will be expressed according to the physical laws of reality (dynamic simulation, deformation, etc.), and also includes the response of virtual objects to the user (action response, force feedback, etc.). With the improvement of computing and graphics hardware performance and the advent of low-cost sensors and depth cameras such as Leap Motion, it is possible to track the spatial motion and positioning of the user's body and use this data for virtual object manipulation. This enables the user to use gestures to interact with the virtual environment in various complex environments. Furthermore, the techniques mentioned above present many advantages in 3D interaction, such as the need for additional bulky equipment.

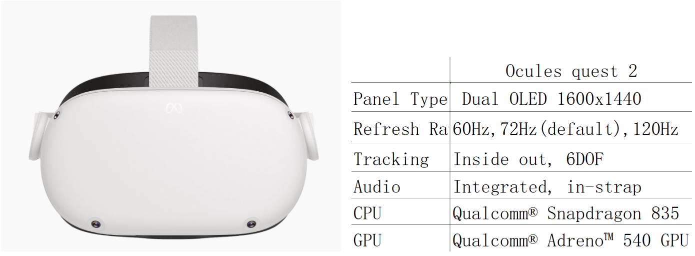
2.3 Devices and Tools

According to the analysis of the highly immersive mixed reality environment in the above two sections, the equipment required for this project is an HMD that can build an MR environment and another that can capture hand movements for interaction. For this, three devices were used: Oculus quest 2 HMD, Zed Mini Camera, and Leap Motion Controller. Each device is relatively independent, so they need to be merged together and work properly as a whole. The idea of using HMD + Zed Mini Camera + Leap Motion Controller to build a mixed reality environment has been applied in different fields. Sergio Serra[12] has applied it to explore the virtual interface in MR. Alberto Badías[2] used this system to calculate the difference between real and virtual objects' mechanical interaction. Next, these three devices are introduced separately.

2.3.1 Oculus quest 2 HMD

Oculus Quest 2 is a virtual reality (VR) headset developed by Facebook Reality Labs. What sets it apart from other VR headsets is that it is a breeze to set up, and it is a completely self-contained device that does not require a computer. With four camera sensors on the headset and six degrees of freedom, the user's location and body can be located and tracked without needing external sensors. It has 1832 x 1920 pixels and up to 120hz refresh rate for each eye, giving a high resolution. The equipment specifications are as shown in the figure 2.1.

In conclusion, its accurate localization and tracking and the characteristics of wireless PC connection make it suitable for the research purpose of this Thesis.



	Ocules quest 2
Panel Type	Dual OLED 1600x1440
Refresh Ra	60Hz,72Hz(default),120Hz
Tracking	Inside out, 6DOF
Audio	Integrated, in-strap
CPU	Qualcomm® Snapdragon 835
GPU	Qualcomm® Adreno™ 540 GPU

Figure 2.1: Oculus quest 2 device specifications

2.3.2 Zed Mini Camera

In mixed reality, there is a need for a device that captures information about the surrounding environment from the center of the user. Zed Mini is one such device.

ZED is a camera that reproduces the way human vision works. It features two high-quality 2K image sensors 63mm apart to reproduce the distance between the human eyes. Using these two "eyes" and through triangulation, it provides a 3D understanding of the scene it observes, allowing the user's application to perceive space and motion. Moreover, with high-speed HD video at 60 fps and a 110-degree field of view, the Zed Mini turns a VR headset into a high-end stereoscopic video pass-through HMD.

(a) (b) (c)

Figure 2.2: (a) Zed Mini Camera. (b) Zed Mini attached to the HMD. (c) the view returned by the Zed Mini.

2.3.3 Leap motion

For users to interact with mixed environments with their bare hands, this requires devices that can accurately identify hand joints and palms and calculate their movements.

Now mainly use leap motion controller. The Leap Motion Controller is an optical hand tracking module that captures the movements of hands with unparalleled accuracy.

It has two IR cameras and multiple LEDs. The LED illuminates a 140 x 120° hemispherical area with infrared light invisible to the human eye to observe the user's hands. The image sensor sends data to the computer via USB to track users' hands. It can recognize 27 different hand elements, including bones and joints, and track them even if other parts of the hand occlude them.

(a) (b) (c)

Figure 2.3: (a) Leap Motion Controller. (b) Leap Motion attached to the HMD. (c) The hand skeleton model recognized and simulated by Leap Motion.

2.3.4 Unity

The development tools of this project are mainly Unity.

Unity is a cross-platform game engine developed by Unity Technologies. It can be used to create three-dimensional (3D) and two-dimensional (2D) games, interactive simulations, and other experiences. The engine has not only supported the development of various desktop, mobile, and console applications but also has a good performance in developing augmented and virtual reality.

On the other hand, the above devices have provided developers with their plugins for Unity, so using Unity can better inherit and manage the devices mentioned earlier.

Chapter 3

Setup integration

In this chapter, the preparatory work before project design and development was introduced, including introducing the modules used by each device and integrating the devices.

3.1 Zed Mini Camera

Testing the Zed Mini camera's functions is the first step to building an MR environment. The test modules mainly include Depth Occlusion, Object Placement, and spatial mapping.

3.1.1 Depth Occlusion

Zed Mini's depth-sensing simulates how human binocular vision works. Its two eyes are separated by 6 to 12 cm, so it can estimate depth and motion by comparing the displacement of pixels between the left and right images so that virtual objects can be occluded by real environmental objects. As shown in Figure 3.1, when a white controller was put in front of the virtual sun, part of the virtual sun was obscured, and then put the controller behind the virtual sun, the real white controller was obscured by the virtual sun.

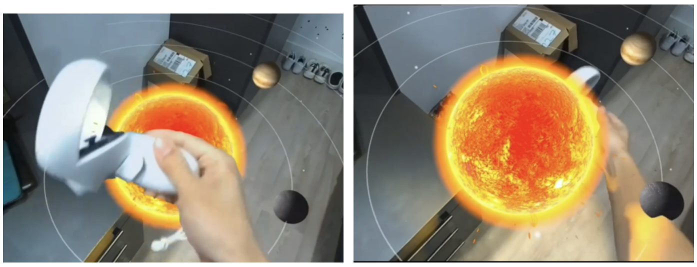


Figure 3.1: Zed Mini Camera depth occlusion

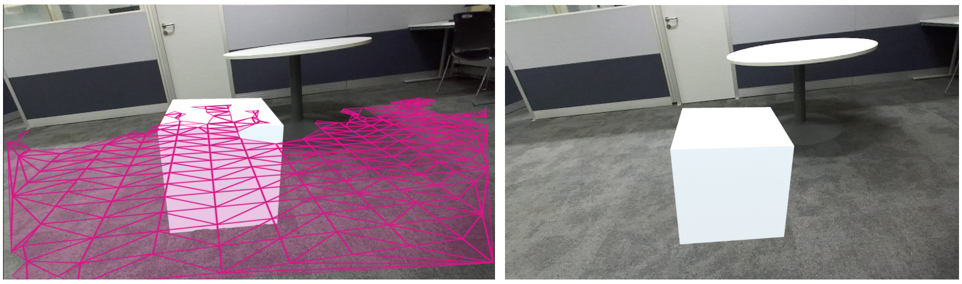
3.1.2 Object Placement

The Zed Mini camera not only captures the depth information of the surrounding environment in real-time but also captures the normal map of the environment. In order to create believable interactive AR/MR experiences, camera-captured depth and normal buffers are integrated into the Unity rendering pipeline. This allows us to place virtual objects in the real scene.

First, the Zed Mini camera extracts the plane in real-time by comparing the similarity of the surrounding environment normals and depth information. In a real test, clicking on any surface in the game view and the plugin will instantly create a game object that fits that surface. The result is shown in Figure 3.2.

Figure 3.2: Plane dection

The plugin will automatically generate a mesh representing the detected plane and attach the corresponding Collider component to it so the virtual objects can be placed on it. As shown in Figure 3.4, a cube with a RigidBody component fell on the mesh representing the plane after clicking on the corresponding plane. From the visual effect, it is like placing a virtual cube on a plane in the real environment.

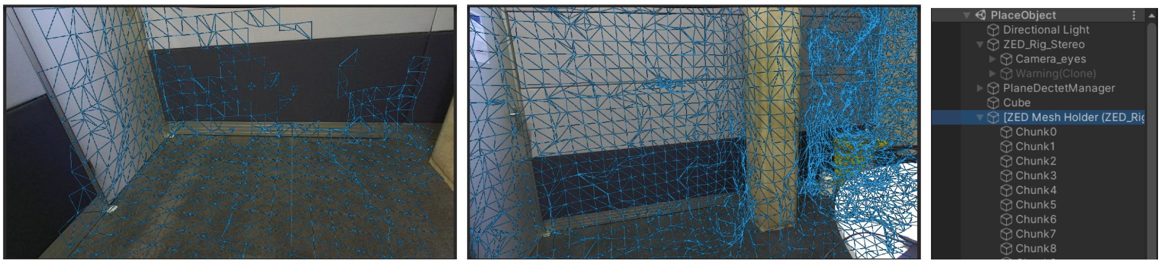


(a) (b)

Figure 3.3: (a) Place object on the plane (b) Hide the plane

3.1.3 Spatial Mapping

Although plane detection can be used to extract planes in the scene in real-time, this dramatically increases the program's cost. Using spatial mapping to scan and create a mesh of the entire scene in advance allows for physical interaction between the real and virtual worlds. The Zed Mini plugin provides a UI for configuring, starting, and stopping spatial mapping. Once launched, it will create a procedural mesh in Unity that can be used at runtime or saved for later use. As shown in Figure 3.4, after clicking the Start Spatial Mapping button to start the mapping, the Zed Mini Camera was moved to capture the environment's mesh, and the scanned area was represented by a blue wireframe. When done, click Stop Spatial Mapping to stop updating the mesh and start mesh filtering. At this point, the plugin will create a Zed Mesh Holder object in the environment that stores all the meshes extracted.



(a) (b) (c)

Figure 3.4: (a) Start spatial mapping (b) Stop spatial mapping (c) ZED Mesh Holder

3.2 Leap Motion controller

In this chapter, the main functions of the Leap Motion Controller were tested, including hand tracking, physical interaction, UI interaction.

3.2.1 Hand Tracking

The Leap Motion plugin can be used for hand tracking directly. We only need to use the Service Provider prefab and place the device body on the corresponding desktop. We can see the recognized and simulated hand model in the scene. As can be seen from Figure 3.5, its recognition is very accurate, and the hand model can follow our hand to move in real-time.

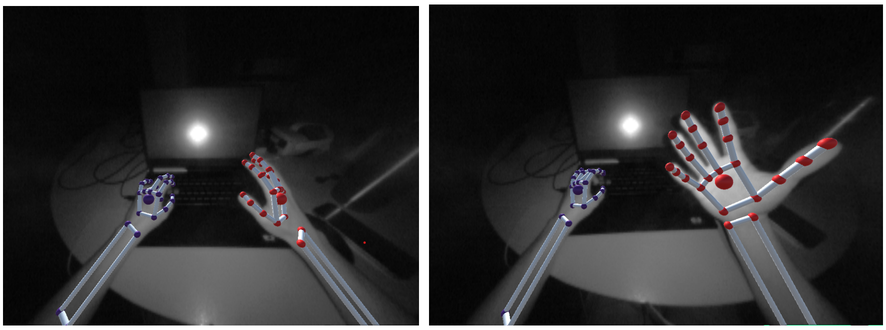


Figure 3.5: Hand tracking

3.2.2 physical interaction

Leap Motion's interaction engine provides the tools to naturally interact with virtual objects using a physical representation of our own hands. With it, we can manipulate arbitrary virtual objects. Its performance was tested through several use cases, as shown in Figure 3.6.

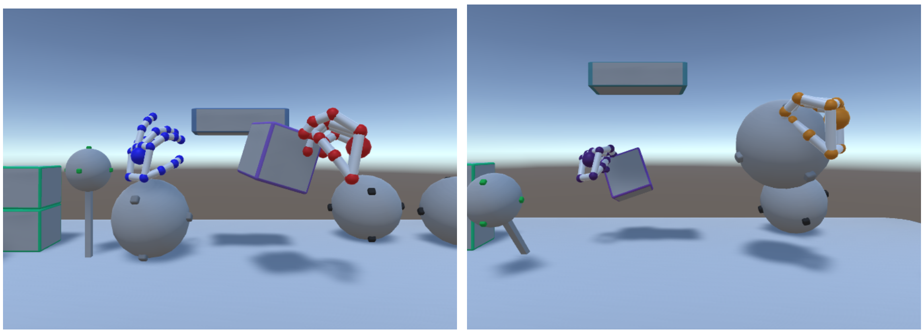


Figure 3.6: Physical interaction

In order to interact with any virtual object, we need to attach a Rigidbody and at least one Collider component. Then we only need to attach the InteractionBehaviour component in the plugin to the corresponding object. We can touch, rag, or throw this object.

3.2.3 UI Interaction

Leap Motion provides us with the InteractionButton and InteractionSlider components for UI operations. The InteractionButton component is a button capable of physical interaction, which can be activated by physically pressing the button and generating different events when pressed and released.

InteractionSlider is a physical slider component that can trigger events by physically pressing and sliding them to the corresponding position. Furthermore, it can be allowed to act as a 1D or 2D slider by increasing its horizontal and vertical sliding limits.

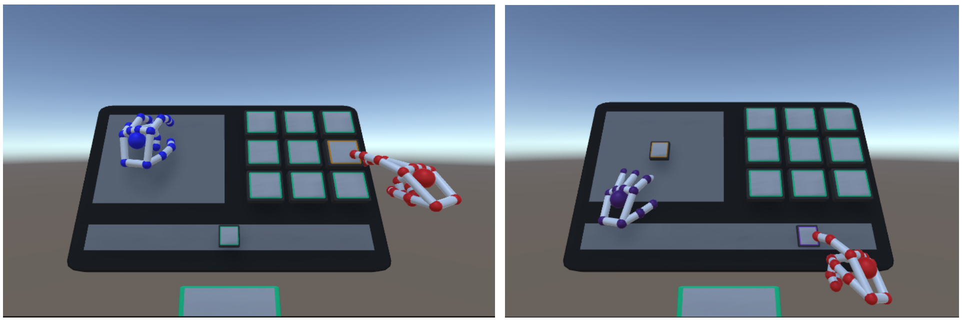

(a) (b)

Figure 3.7: (a) Press button (b) Slide button interaction

Button interaction works well in most cases, but to avoid unnecessary interaction and unnecessary occlusion of the UI interface with objects other than the hand in the environment, the developers of Leap Motion have come up with a solution that utilizes the Leap Motion tracking accuracy and allows multiple virtual objects to be attached to any of the 21 possible data points so that a UI panel can be attached to the hand. The hand can be tracked in real-time as it moves. Then use the SimpleFacingCameraCallbacks script to check if a particular hand is facing the same direction as the camera. It raises 2 Unity events so the logic can be driven from the Unity Inspector. When the palm is facing up, the hand UI is displayed. When the palm is facing down, the hand UI will be hidden, and the test results are shown in Figure. 3.8.

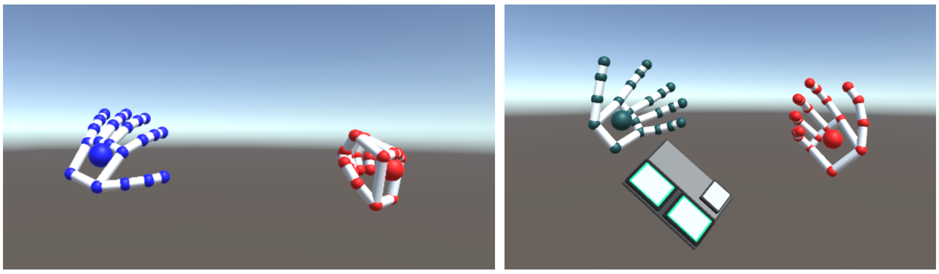

(a) (b)

Figure 3.8: (a) When palm is facing up, the hand UI is displayed (b) When palm is facing down, the hand UI will be hidden

3.3 Setup integration

After the above-mentioned corresponding tests for the functional modules of each required device, the next step is to integrate each device so that each required module's functions cooperate to build an MR environment that can interact naturally.

3.3.1 HMD integration with Zed Mini Camera

As introduced in section 3.1, the Zed Mini Camera can obtain the depth and normal information of the real environment to realize the interaction between the real environment and the virtual environment. However, the current output image will only be displayed in Unity's game view. Therefore, the HMD should be integrated with the Zed Mini Camera first so users can be better immersed in the MR environment through the HMD.

Zed Mini's depth-sensing simulates how human binocular vision works, so the position of the Zed Mini Camera on the VR glasses must be in the center of the HMD to simulate the field of view of the eyes. To this end, the Zed Mini Camera developers provide a mount to connect it with the HMD (see Figure 2.2 (b)) and use the Zed Camera prefab in Unity to process Zed's image, tracking, and headset output. We can observe in Figure 3.9 that the Zed Mini Camera capture the real environment in a stereoscopic manner and presents it in front of our eyes in the HMD, and its depth occlusion and spatial mapping test results are still accurate.

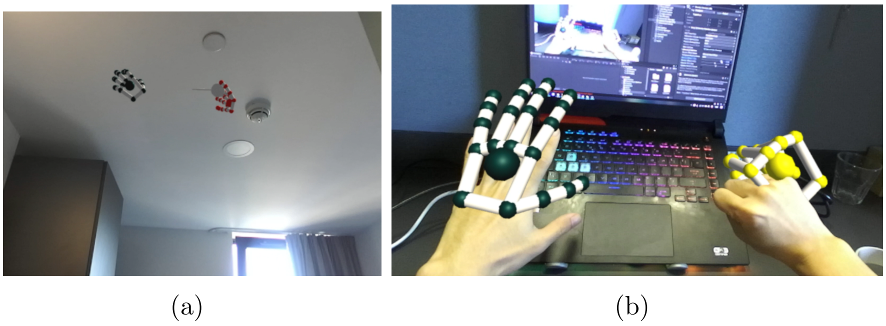
Figure 3.9: HMD integration with Zed Mini Camera

3.3.2 Leap Motion Integration

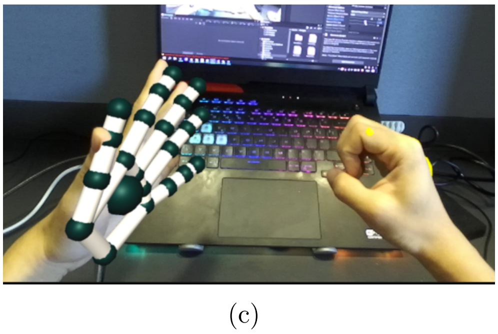
A method of integrating Leap Motion was proposed by StereoLabs (the developer of the Zed camera), which is to put the Leap Motion below the Zed Mini Camera.

Because Leap Motion needs to bind a Unity camera to its prefab to track the HMD, the hand model it recognizes and simulates will always appear in front of the camera. Suppose we do a VR project with only a Leap Motion. In that case, the final image output to the HMD is the image processed by the camera bound to the Leap Motion prefab. However, in this project, the camera prefab of Zed Mini mentioned above is responsible for headset tracking and image output so that the image finally seen in the headset is the image processed by Zed Mini, which requires that the two-hand model should always appear in front of the Zed Mini Camera and moves with it, so a script attached to it to the camera bound by Leap Motion to keep the position and angle of its transform the same as the Zed Mini. So that the hands model always appears in front of the Zed Mini. Finally, adjusts the offset slightly to make the virtual hands coincide with the real hands.

As shown in Figure 3.10(a), if the Leap Motion does not follow the Zed Camera, the hand model will always appear in the wrong place and cannot interact correctly. Figure 3.10(b) shows that the attached script on Leap Motion makes it move with the Zed Mini Camera so that the hand model is always in a relatively correct position. However, there is still a deviation between the identified bones and the real hand. Only needs to adjust the offset parameter through the x-y-z axis on the prefab of Leap Motion, and the recognized virtual hands will be displaced. After repeatedly adjusting the offset parameters, as shown in Figure 3.10(c), the virtual hand can coincide with the real hand to the greatest extent.



(a) (b)



(c)

Figure 3.10: Adjust virtual hands' offset

After completing the setup of Leap Motion, Leap Motion's functions were re-tested to determine whether it can normally work in the MR environment:

1. As in section 3.3.1, first spatially map the environment, place a cube with Rigidbody and Collider components in the environment, and uncheck the gravity option of its Rigidbody component.

2. As shown in Figure 3.11(a), when the cube is pushed and dragged, the cube can respond correctly.

3. As shown in Figure 3.11(b), after the cube is pushed away, the cube will interact with the real environment that has been spatially mapped, specifically, it will bounce and rotate after collision


(a) (b)

Figure 3.11: Drag object in MR

This final combination inherits all the properties of MR that we have discussed for natural interaction. As shown in the figure 3.12, users only need to put on the headset to be immersed in the MR environment. They can interact with the virtual environment with bare hands, providing a solid foundation for designing and implementing subsequent immersive artworks.

Figure 3.12: Devices setup

Chapter 4

Design and Implementation

This chapter describes the development process of a naturally interactive immersive art demo. After the previous research and analysis on mixed reality immersive art, natural interaction, and device setup, developing this demo turned theoretical research into a practical project. Through this demo, users will truly experience the sense of immersion it brings.

The whole development process is mainly divided into content design and specific implementation.

4.1 Design

4.1.1 Overall design

The basis of the design is first to determine what the demo should include since this project focuses on exploring natural interaction behavior in mixed reality environments. According to Bachmann et al. [1], mixed reality environments contain four basic interaction tasks: selection, manipulation, system control, and navigation.

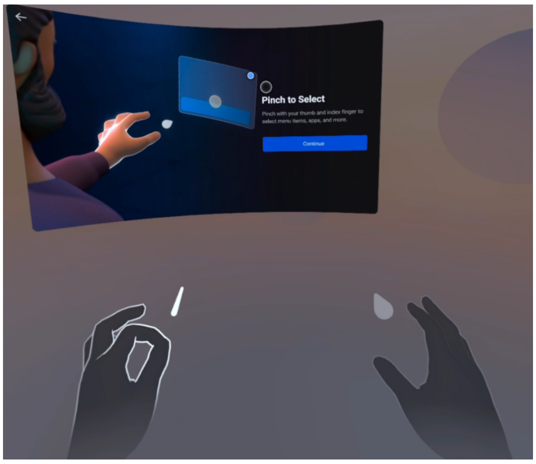
1. Selection: The user is able to select an object to perform an action.
2. Manipulation: The user is able to change any object properties, such as scaling, displacement and other operations on the object.
3. Navigation: The user is able to change position or orientation in an immersive environment.
4. System Control: The user can make changes to the system state, such as menu-based change operations.

4.1.2 Detailed design

Based on the analysis of the basic interaction tasks that should be included above, the next step should be to determine the specific content of each task. This demo is inspired by Teamlab's artwork: "Forest of Flowers and People: Lost, Immersed and Reborn"[1]. This team built a large number of curtains in the pavilion, and then used laser projection technology to continuously project countless computer-rendered flower animations on the curtains, and used dynamic recognition technology to enable visitors to interact with the flowers, such as if a person stands still, then the flowers around them will grow more lush. Inspired by this installation art, the specific content in this demo should be:

1. Animation: In the mixed reality environment, flowers should continuously grow on the floor and walls, and each flower should include animations of growth and withering, and some flowers would have an animation of falling petals after growing.

2. Selection: According to the traditional device operation habits, the computer controls the position of the cursor by controlling the movement of the mouse and then selects it by clicking. Because all interactions in the demo are performed freehand, the hand movement will be detected by Leap Motion to move the cursor in the MR environment, and the cursor will be represented by a sphere. This is inspired by the hand tracking interaction of Oculus quest 2(as shown in Figure 4.1), which shoots a ray from the middle of the thumb and index finger to control the cursor and then pinch the thumb and index finger to represent a click.



Pinch to Select

Pinch with your thumb and index finger to select menu items, apps, and more.

Figure 4.1: Oculus quest 2 hands tracking

3. Manipulation: Users should be able to touch, drag, and throw flowers within reach. For unreachable distant flowers, users can choose whether to move them into the

---

[1] https://www.teamlab.art/w/flowerforest/

reachable range or directly perform remote movement operations through different gestures. The specific remote operation method is:

- When the thumb only touches the index finger as Figure 4.2(a), users can move the distant flower to another location on the ground or wall.
- When the user keeps the index finger upright and touches the thumb with the other three fingers except for the index finger as Figure 4.2(b), the selected flower in the distance will move directly to the user's hand.

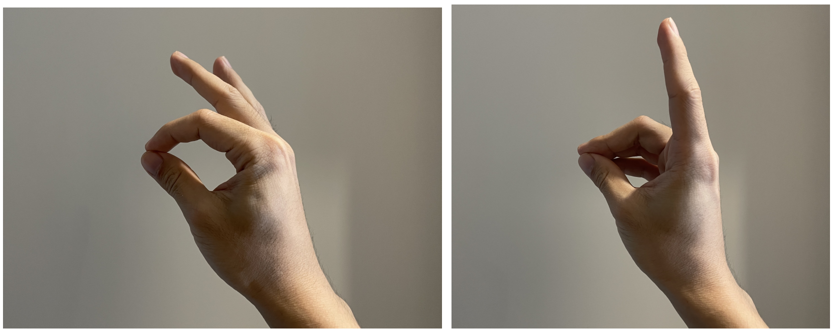

(a) (b)

Figure 4.2: (a) Gesture for direct remote control of objects (b) Gesture for getting objects from remote

4. Navigation: As previously introduced in section3.1, the Zed Mini Camera will continuously track the user's position in the environment, so in the demo, users are free to move around the environment.
5. System Control: The user should be able to use the menu in the demo to control the display and hiding of the 3D model of the hand, the start and end of the spatial mapping, the display and hiding of the environment mesh, and the generation and clearing of flowers.

4.2 Implementation

According to the content of the detailed design, the demo can be divided into the following modules: animation and generation of flowers, hand interaction, and system control.

4.2.1 Animation and generation of flowers

Flower generator

First, a flower generator should be implemented, which can continuously randomly obtain the position where flowers can grow on the ground or the wall in the environment, and then generate a random flower model on the obtained position.

In order to obtain a suitable growth position, first, get the current Zed camera position through the Zed camera plugin and place a generator in front of its line of sight. Use the Random.insideUnitSphere() function to randomly generate a vector (Vector3) within a sphere of radius one around the generator. Then use Physics.Raycast() function to cast a ray from the center of the sphere in the direction of the generated vector. If the ray hits the terrain mesh, it returns a hit point object of type RaycastHit, which includes the location information of the hit point and normals and other information, and then traverse all the returned hit point objects, and use Instantiate() to randomly generate a flower prefab at the corresp, which is convenient for the removal operation in the subsequent section 4.2.3.

Flower animation

After getting the spawn location of the flower, a flower model needs to be generated at this position and play its animation. First, consider the model of the flower. The Unity Store provides many models of flowers. A few ones were chosen for demo use, as shown in Figure 4.3[2]. However, these model assets provided by the Unity store are inseparable, so they can be animated as a whole. However, the animation of their floating petals cannot be achieved.

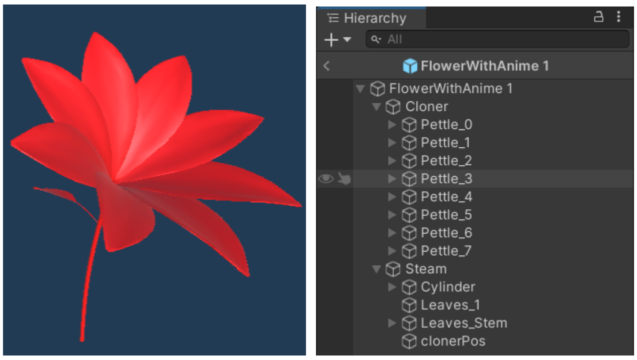
In order to solve the above problems, the modeling software Cinema 4D was used to make a flower model realize the effect of petal floating. In Unity, each part of this flower model can be split. Its structure is shown in Figure 4.4, So I can control each petal individually to achieve the desired effect.

For the overall animation of the flower, when it first spawns, set its size to 0. It will randomly increase to the corresponding size over time. At this time, set the ”is kinematic” property of its Rigidbody component to true so that it will be fixed on the ground. In order to avoid unnecessary collisions with other flowers during the growth process, the Collider component will be disabled first. After that, each flower will have a 20% chance of floating into the air. When it decides to do this, the script will set the ”is kinematic” property to false and move towards the normal direction of the growth position, apply a random direction rotation to it and enable Collider.

---

[2] https://assetstore.unity.com/packages/3d/vegetation/flowers/low-poly-trees-pack-flowers-178576

Figure 4.3: Flower models from Unity store



(a) (b)

Figure 4.4: (a) splittable flower (b) model structure

Each flower will have a random health value when it is generated, and it will decrease over time after it is generated. When it decreases to zero, the flower begins to wilt. Similar to the growth process, the wilting process is that its size decreases over time. When it becomes zero, use the Destroy() method to clear.

An additional Animation component was added to the flower model for the petal floating effect. Through this component, the petals of the flower can be controlled to drift to different positions over time by adding keyframes, as Figure 4.5.

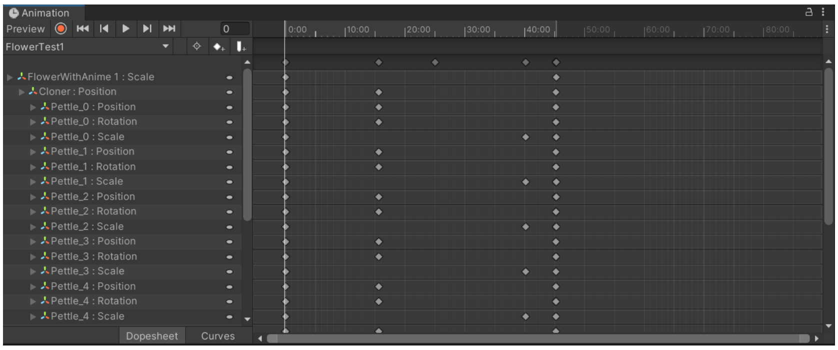


Figure 4.5: Flower Animation component

4.2.2 Hand interaction

There are two cases of hand interaction operation, one is the interaction with the flowers that can be reached, and the other is the interaction with the flowers that cannot be reached from a distance.

Interact with flowers within reach

As tested in Chapter 3.2, Leap Motion's interaction engine enables users to create physical representations of hands and VR controllers. The interaction engine includes grasping, throwing, stable collision feedback, and hand proximity detection. It provides a full-featured interaction API as well as samples and prefabs. The desired interaction behavior can be achieved with these prefabs:

- First, use the InteractionManager script to receive from Unity and process all the internal logic that makes the interaction possible, including updating hand data and interaction object data.
- Second, add the InteractionController script to the interaction hands under the InteractionManager. This completes all interactions with interacting objects, whether by picking them up, touching them, hitting them, or just approaching them.
- Finally, attach an InteractionBehaviour script to any object with a Rigidbody and at least one collider to poke, poke, slap, grab, or throw it.

Interact with distant flowers

According to the previous detailed design, the operation of distant objects must implement the selection function and the manipulation.

As for the selection function, the position of the cursor ball controlled by hands should be calculated first. In the development document of Leap Motion, it is found that the Hands.GetPredictedPinchPosition() method can be used to obtain the pinch position of the right thumb and index finger or an approximate value of the predicted position to be pinched. Also, use Hand.PalmNormal() method to get the direction the palm is facing.

Moreover, the Physics.Raycast() method is used to cast ray from the returned position to the direction of the palm of the right hand, and a small sphere is placed at the position where it hits to represent the cursor. Figure 4.6 shows that the cursor has moved to the target flower.

As analyzed in detailed design, manipulation consists of directly moving distant flowers and dragging flowers into hands. These two different operations are distinguished by different gestures representing clicks. The implementation process is as follows:

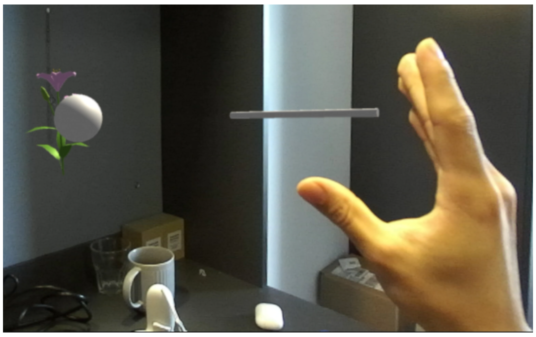


Figure 4.6: Move the cursor onto the plower

Directly moving distant flowers: After moving the cursor to the target flower, the selected object can be returned through RaycastHit.transform.gameObject. In the click operation, to avoid the confusion of the two operations of pinch and grab, first use GrabStrength() to return the currently grasped degree. If it is less than the threshold, there is no object to be grasped at present. Then use Hand.ispinching() to judge whether the right hand is pinching and if so, use Hand.PinchDistance() to return the distance between the thumb and index finger of the right hand. When the returned distance is less than the threshold, it is considered that I have clicked or selected, and then set the object hit by the cursor to move with the cursor until the distance between the index finger and the thumb is greater than the threshold. Moreover, set the "is kinematic" property of the flower to true. Figure 4.7 shows directly moving the flower to other terrain places.

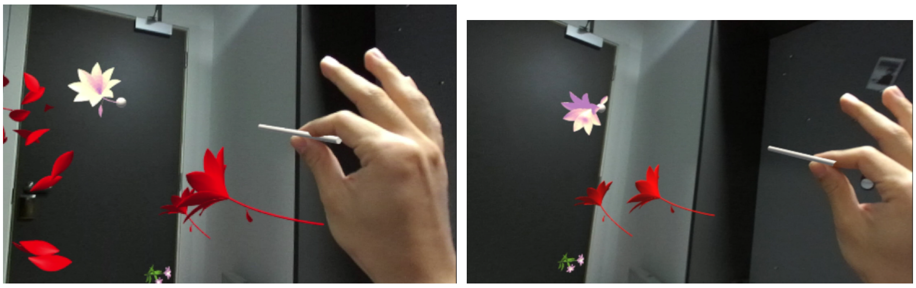


Figure 4.7: Directly moving distant flowers

moving flower into hand: Similarly, if the hand is not currently gripping and the distance between the thumb and index finger is greater than a threshold. Put the position of the cursor at the position returned by Hands.GetPredictedPinchPosition() (the predicted position where the right thumb and index finger touch), and when the cursor moves to this position, the hit flower will also follow the cursor back to the hand. Finally, set the flower's "is kinematic" property to false. This effect is shown in Figure 4.8. When the index finger is held vertically, and the thumb touches the other three fingers, the target flower will return to the right hand.

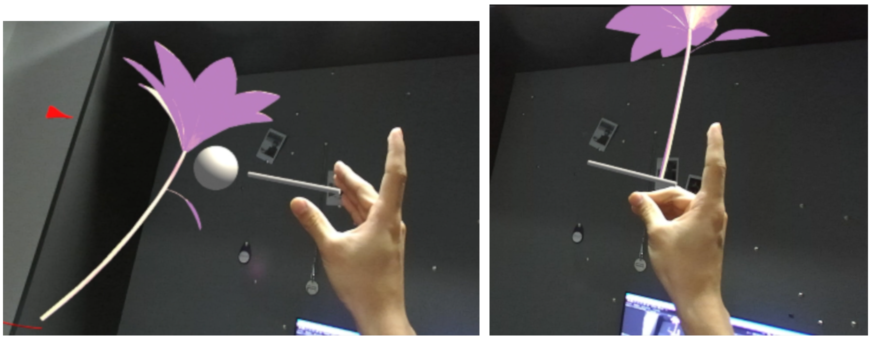


Figure 4.8: Moving flower into hand

4.2.3 system control

In order to realize the user's manipulation of the system, it is necessary to provide the user with an interactive virtual panel, which relies on the Leap Motion UI module tested in chapter 3.2.3. First, attach the entire control panel to the left hand, and check the orientation of the palm. When the palm is facing up, the panel will be displayed, and when the palm is facing down, the panel will be hidden. Place the button prefab provided by the developers of Leap Motion on the panel that can bind functions, and then find the functions corresponding to the start and end of the control space mapping in the Zed camera and Leap Motion source codes, respectively bind them to a button prefab. Use a similar method to bind other control functions to the control panel, as shown in Figure 4.9.

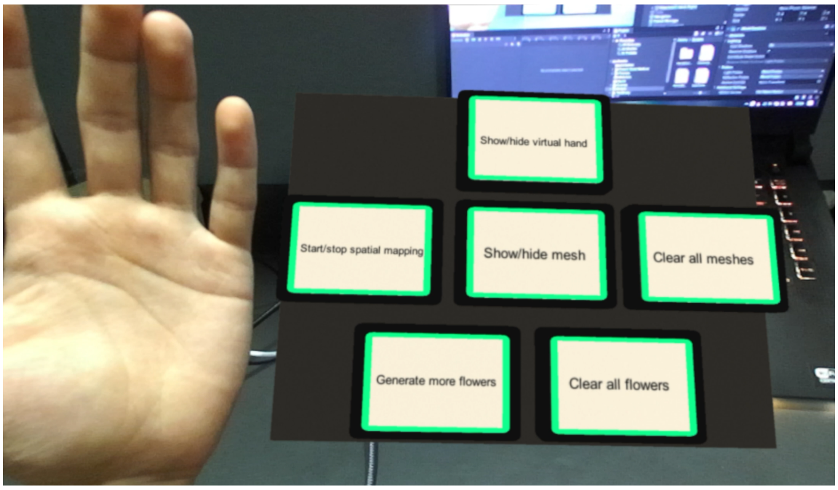


Figure 4.9: Control panel

With this demo, it is possible to evaluate whether immersive art with natural interaction in MR really brings a better immersive experience to users.

Chapter 5

Evaluation

A questionnaire was used to evaluate whether providing natural interaction in immersive art helps to enhance the user's sense of immersion.

There are 7 participants in this assessment. They are master's students from Trinity College Dublin's computer science augmented and virtual reality strand. All of them have experience using AR applications on mobile phones and tablet devices, So they can better perceive the contribution of natural interaction to immersive AR art.

5.1 Experiments

Here, the User Experience Questionnaire(UEQ) was used to assess the user's immersion in the demo. UEQ is a quick evaluation to measure user experience in an interaction task with any physical or virtual product described in [11]. This assessment is based on six dimensions (shown in Table 5.1): attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. As shown in the table 5.1, these features are descriptions of the immersive and interactive experience of the product, including all possible opinions that users might have in the demo we implemented.

Dimension	Description
Attractiveness	Overall impression of the demo
Perspicuity	Is it easy to get familiar with the natural interaction method?
Efficiency	Does the product react to user's interactive behavior quickly?
Dependability	Does the user feel in control of the interaction?
Stimulation	Is it exciting to use this natural interaction method in an immersive artwork?
Novelty	Is this natural interaction method creative in an immersive artwork?

Table 5.1: UEQ dimensions and corresponding description

These dimensions are calculated from the 26 questionnaire items in 5.1 in which the user must assess several aspects of the natural interaction being tested. Table 5.2 shows some examples of forms that participants will fill out after experiencing the demo, where each item must be scored on a 7-point Likert scale based on how they felt about the demo. Answers are scaled from -3 (completely agree with negative items) to +3 (completely agree with positive items).

annoying	-3 -2 -1 0 1 2 3	enjoyable
not understandable	-3 -2 -1 0 1 2 3	understandable
dull	-3 -2 -1 0 1 2 3	creative
complicated	-3 -2 -1 0 1 2 3	easy
……	-3 -2 -1 0 1 2 3	……
confusing	-3 -2 -1 0 1 2 3	clear

Table 5.2: Part of UEQ items

Before the participants start the demo, they will be told to pay more attention to the difference between the interaction method in the demo and the traditional AR interaction method and its impact on the sense of immersion.

After the demo, let the participants complete the questionnaire quickly instead of doing a long analysis of each item.

Finally, analyzing the questionnaire results will allow us to understand whether the project has achieved its original purpose and what areas can be improved.

5.2 Results

Each user was required to quickly complete the UEQ form after experiencing the demo and was repeatedly required to focus on the impact of the interaction method in the demo on the sense of immersion. After collecting all the forms, calculate the above six dimensions according to the method in [11], as shown in the figure 5.1.

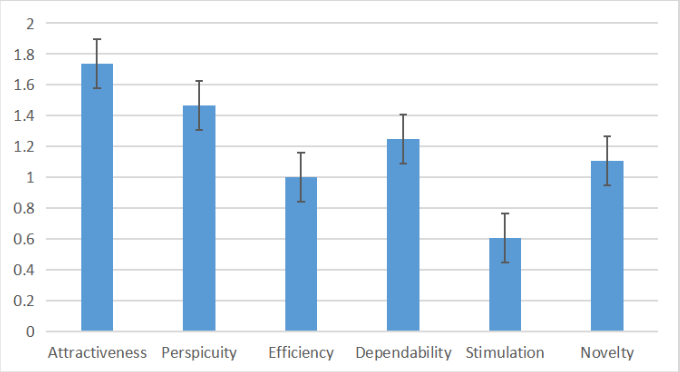


Figure 5.1: Evaluation result

The above result plot shows two descriptive parameters for each dimension: the mean and the score range for each dimension. As can be seen from the figure, MR immersive art that can interact naturally is very attractive. Compared with their previous AR experience, this is an innovative idea. This interaction is based on people's everyday behavior, so it is easy to learn how to operate. In addition, the user feels this way of interaction is controllable. However, the specific interaction and its fluency should be improved. The gesture control method implemented in the demo does not stimulate the user very well.

Chapter 6

Conclusions & Future Work

6.1 Conclusions

This thesis starts with the introduction of immersive art, introduces the current application of AR technology in the field of immersive art, and then analyzes the importance of Natural User Interfaces for natural human-computer interaction according to the limitations of these applications in immersion and human-computer interaction. Then proposed using head-mounted displays and hand tracking devices to enhance the immersion of immersive art. Finally, a demo was designed and implemented for the proposed device, and subjective tests were conducted on multiple users to evaluate the immersion improvement brought by natural interaction with immersive art.

The real challenge in this project was the integration of Oculus quest 2, Zed Mini, and Leap Motion, needing to ensure the availability of the required modules for each device and to work with each other after integration. The implementation of the hand interaction function is another challenge. Leap Motion provides APIs available to developers, and it is necessary to find the required functions in a large number of documents to implement each designed complex logic.

For users to better appreciate the benefits of natural interaction in immersive artworks, a visually appealing demo was designed and used subjective evaluation to evaluate user satisfaction. The results showed that this interaction method is novel and attractive enough. It is helpful to enhance the sense of immersion. However, if it is more fluent in specific interaction tasks, it will bring a better immersive experience.

Through this dissertation, my most significant gain is the improvement in analysis ability. Although there were only a few general ideas at the beginning of this project, after a step-by-step analysis, the different requirements of each stage were gradually determined, and for each requirement, investigate a reasonable solution. Another gain is that I have a preliminary practice for designing and constructing the MR environment, which has laid a good foundation for my future research on MR immersive art.

6.2 Future Work

Based on the evaluation results in the previous chapter, some improvements are proposed to solve the problems found in the project. For example, the implementation logic needs to be improved for specific interactive tasks. When the user is grasping the virtual object, the current logic may misjudge the selection of distant objects, causing objects that should not be manipulated to appear in the wrong place. Another problem is that currently, users can only simply drag and drop virtual objects, and some more attractive interactive tasks should be designed in the future.

There are also some issues with the devices. For example, there is a wide border around the actual screen input from the Zed Mini to the HMD so that the field of view seen by the user is much smaller than the maximum field of view that the HMD can provide. Another problem is that the interaction engine provided by Leap Motion is not very precise in some operations, such as when the user throws the object in one direction, the object may fly in another strange direction.

Furthermore, I believe that with the development of advanced technologies such as VR, AR, and MR, the expression of immersive art will never be satisfied with using traditional projection and other devices as a carrier. From the perspective of the current application of VR technology in immersive art, the combination of traditional installation art and AR and MR technology would have better development prospects. If traditional installation art can be well utilized with the real-time interaction between real and virtual objects in MR would stimulate people's senses more, but too much or too little content in the virtual environment will affect the harmony of the entire work, so considering the proportion of real environment content and virtual environment content in artworks will require artists to think hard.

Bibliography

[1] Daniel Bachmann, Frank Weichert, and Gerhard Rinkenauer. Review of three-dimensional human-computer interaction with focus on the leap motion controller. *Sensors*, 18(7), 2018.

[2] Alberto Badías, David González, Icíar Alfaro, Francisco Chinesta, and Elías Cueto. Real-time interaction of virtual and physical objects in mixed reality applications. *International Journal for Numerical Methods in Engineering*, 121(17):3849–3868, 2020.

[3] Biplab Ketan Chakraborty, Debajit Sarma, M.K. Bhuyan, and Karl F MacDorman. Review of constraints on vision-based gesture recognition for human–computer interaction. *IET Computer Vision*, 12(1):3–15, 2018.

[4] Yunyan Chen. *Immersive Exhibition.* PhD thesis, 2021.

[5] Adrian David Cheok, Michael Haller, Owen Noel Newton Fernando, and Janaka Prasad Wijesena. Mixed reality entertainment and art. *International Journal of Virtual Reality*, 8(2):83–90, Jan. 2009.

[6] Joseph L. Gabbard. A taxonomy of usability characteristics in virtual environments. 1997.

[7] Wolfgang Hönig, Christina Milanes, Lisa Scaria, Thai-Binh Phan, Mark T. Bolas, and Nora Ayanian. Mixed reality for robotics. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5382–5387, 2015.

[8] Theofilos Papadopoulos, Konstantinos Evangelidis, Theodore H. Kaskalis, Georgios Evangelidis, and Stella Sylaiou. Interactions in augmented and mixed reality: An overview. *Applied Sciences*, 11(18), 2021.

[9] Meng Qu. The aesthetic experience of augmented reality art. pages 79–86, 06 2017.

[10] Siddharth Swarup Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43:1–54, 2012.

[11] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. Applying the user experience questionnaire (ueq) in different evaluation scenarios. In *HCI*, 2014.

[12] Sergio Serra, Redouane Kachach, Ester Gonzalez-Sosa, and Alvaro Villegas. Natural user interfaces for mixed reality: Controlling virtual objects with your real hands. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 712–713, 2020.

[13] Gideon Steinberg. Natural user interfaces. In *ACM SIGCHI conference on human factors in computing systems*, 2012.