北京郵電大學
Beijing University of Posts and Telecommunications

Queen Mary
University of London

Undergraduate Project Report
2025/26

Development of a Digital Twin of a Morse Telegraph

Name:	Canxi YANG
School:	International School
Class:	2022215112
QMUL Student No.:	221168420
BUPT Student No.:	2022213370
Programme:	**Internet of Things Engineering**

Date: 26-04-2026

Table of Contents

Abstract *3*

Keywords *3*

Chapter 1: Introduction *5*

1.1 Project Background and Motivation **5**

1.2 Project Objectives **6**

1.3 Methodology and Development Workflow **6**

1.4 Structure of This Report **7**

Chapter 2: Background *9*

2.1 Morse Telegraph: History and Technical Principles **9**

2.1.1 Historical Context and Development 9

2.1.2 Technical Principles 10

2.2 Digital Twin Technology in Cultural Heritage **12**

2.2.1 Concept and Current Applications 12

2.2.2 Research Gap: Mechanical Cultural Relics 13

2.3 Key Technologies Used in This Project **13**

2.3.1 3D Scanning with LiDAR 13

2.3.2 3D Modelling with Blender 13

2.3.3 Interactive Development with Unity 14

2.3.4 Meta Quest VR glasses 14

Chapter 3: Design and Implementation *15*

3.1 3D Data Acquisition **15**

3.2 Model Refinement in Blender **17**

3.2.1 Key Model Refinement 17

3.2.2 Receiver Model Refinement 19

3.3 Historical Scene Construction **21**

3.3.1 Sending Room 21

3.3.2 Receiving Room 23

3.4 Interaction System: Sending Station **24**

3.4.1 System Architecture Overview 25

3.4.2 From Physical Press to Morse Signal: the Core Logic 26

3.4.3 Training Mode 26

3.4.4 Operating Mode 28

3.5 Interaction System: Receiving Station **29**

3.5.1 System Architecture 30

3.5.2 Receiver Animation 31

3.5.3 Morse Signal Reception and Display 33

3.5.4 Morse Decoding Mechanism 33

3.6 Integration and Cross-Scene Connection **35**

3.7 VR Integration and Deployment **36**

Chapter 4: Results and Discussion *40*

4.1 Sending Room: Walkthrough of Operation **40**

4.2 Receiving Room: Walkthrough of Reception **43**

4.3 Discussion and Reflection **45**
4.3.1 The Value of a Reproducible Workflow 45
4.3.2 Limitations 46
Chapter 5: Conclusion and Further Work ***48***
5.1 Conclusion **48**
5.2 Reflection **48**
5.2.1 Ethical Considerations 48
5.2.2 Social Impact 48
5.2.3 Legal Considerations 49
5.2.4 Environmental Considerations 49
5.2.5 Personal Reflection 49
5.3 Further work **50**
References ***51***
Acknowledgement ***53***
Appendices ***54***
Disclaimer **54**
Acknowledgement of using Generative AI **55**
Project specification **56**
Early-term progress report **62**
Mid-term progress report **67**
Supervision log **72**
Risk and Environmental Impact Assessment ***75***

Abstract

The Morse telegraph is one of the most important inventions of the 19th-century communication revolution. It combines a precise mechanical structure with a complete operating procedure, holding significant historical and educational value. However, due to the need for preservation, telegraph machines in museums are usually displayed in a static manner, and visitors are unable to experience the full process of sending and receiving messages. This project aims to develop an interactive digital twin system for the Morse telegraph held in the Vicente Miralles Segarra Museum of Telecommunications at UPV, enabling users to experience the complete process of encoding, sending, receiving and decoding telegraph messages in a virtual reality environment.

The project follows a step-by-step development workflow. First, an iPhone equipped with a LiDAR sensor was used to perform 3D scanning of the physical telegraph to obtain the raw geometric data. Then, the scanned models were refined, optimised and placed into reconstructed historical scenes in Blender. After that, a complete interaction system was developed in Unity, including a Training Mode for beginners and an Operating Mode that supports free-form message input. Finally, the system was deployed to a Meta Quest 3 VR headset with hand-tracking interaction.

The final system has achieved all the objectives set out in the project specification. In addition, this project has also validated a complete and reusable development workflow from physical scanning to VR deployment, which relies only on consumer-grade hardware and free software, and may serve as a methodological reference for the digital reconstruction of other mechanical cultural artefacts.

Keywords

Digital Twin, Morse Telegraph, Virtual Reality, Cultural Heritage, Unity,

摘要

莫尔斯电报机是 19 世纪通信革命的核心发明之一，兼具精密的机械结构与完整的操作流程，具有重要的历史和教育价值。然而，出于文物保护的需要，博物馆中的电报机通常只能静态陈列，观众无法体验完整的收发报过程。本项目旨在为 UPV Vicente Miralles Segarra 电信博物馆馆藏莫尔斯电报机开发一套交互式数字孪生系统，使用户能够在虚拟现实环境中体验电报的编码、发送、接收与解码全过程。
项目采用分阶段递进的开发流程：首先使用搭载 LiDAR 传感器的 iPhone 对实物电报机进行三维扫描，获取原始几何数据；随后在 Blender 中对模型进行修复、优化与场景搭建；之后在 Unity 中开发了完整的交互系统，包括面向初学者的训练模式和支持自由编辑的操作模式；最终将系统部署至 Meta Quest 3 头显，支持手部追踪交互。
最终系统实现了项目规范中设定的各项目标。此外，本项目还验证了一套从实物扫描到 VR 部署的完整、可复用的开发工作流，该流程仅依赖消费级硬件与免费软件，可为其他机械类文物的数字化重建提供方法参考。

关键词

数字孪生，摩尔斯电报，虚拟现实，文化遗产，Unity

Chapter 1: Introduction

1.1 Project Background and Motivation

With the continuous improvement of material living standards, people are paying increasing attention to their spiritual and cultural lives, and fields such as cultural heritage and the arts have become popular topics in recent years. As more and more ordinary visitors walk into museums, a growing number of emerging technologies are also being introduced into the display and educational work of cultural relics. Beyond the most traditional exhibit labels and audio guides, many museums have started to adopt technologies such as Augmented Reality (AR) and Virtual Reality (VR) to support the public education of their collections.

On the other hand, mechanical cultural relics have their own particular characteristics — their value lies not only in "what they look like", but also in "how they work". Yet for preservation reasons, most museums can only display them inside glass cases, where they can be observed but never touched. This contradiction makes mechanical cultural relics naturally well suited to interactive VR digital twins, but research on digital twins for this particular category of cultural objects is still relatively uncommon (see Section 2.2.2).

The Morse telegraph is a typical example of such a mechanical cultural relic. It is not only one of the core inventions of the 19th-century communication revolution and a symbol of humanity's entry into the era of "instant communication", but also a device that combines a delicate mechanical structure with a complete operational procedure, and both of these aspects are difficult to convey through traditional static display. In museums, Morse telegraphs are usually placed behind glass, where visitors can neither understand how the lever mechanism of the key actually works, nor experience the full process through which a message is tapped, encoded, transmitted, received and decoded. Based on this observation, the present project attempts to develop an interactive digital twin system for the Morse telegraph held in museums, allowing users to experience the complete process of sending and receiving telegraph messages in a VR environment. This approach responds to two seemingly contradictory needs at the same time: the preservation of the physical artefact, and the provision of an interactive experience, since users can operate the virtual telegraph repeatedly without causing any damage to the original object. In addition, it is hoped that this project may serve as a methodological exploration that provides a useful reference for the development of digital twins for other mechanical cultural relics in the future.

1.2 Project Objectives

The general objective of this project is to develop an interactive digital twin of a 19th-century Morse telegraph. The system aims to faithfully reproduce the original artefact in terms of visual appearance, mechanical structure and operational procedure, and to allow users to experience the process of sending and receiving telegraph messages within a VR environment. At the same time, the project also aims to validate a complete, reusable workflow that may serve as a methodological reference for the digital reconstruction of other mechanical cultural artefacts.

To achieve this general objective, the following specific objectives have been set:

- Digitisation of the physical artefact: Using LiDAR scanning to acquire high-precision three-dimensional raw data of the Morse telegraph key and receiver.
- Model refinement: To clean, repair and reorganise the meshes obtained from the scans in Blender, and to rebuild small mechanical components whose scan quality was insufficient.
- Reconstruction of historical scenes: To build, in Blender, two virtual indoor environments with a 19th-century atmosphere — a Sending Room and a Receiving Room — so that the user experience is not limited to operating the device itself, but also immerses the user in the historical context in which the telegraph was used.
- Mechanical animation: To develop animations for the key moving parts of the telegraph, and to manage the overall interaction flow through a finite state machine and a set of C# scripts.
- Two-mode interaction system: To develop two progressive interaction modes. A Training Mode, which guides beginners through the standard procedure of sending Morse code with on-screen instructions, and an Operating Mode, which allows users to send messages freely and observe the decoded result in real time at the receiving end.
- VR integration and deployment: To integrate and deploy the complete system on Meta Quest series VR headsets using Unity's XR Interaction Toolkit, allowing users to perform all interactions naturally through VR controllers.

1.3 Methodology and Development Workflow

The methodology adopted in this project can be summarised as a progressive workflow that

moves from historical and contextual research towards technical implementation.

Although the final outcome of the project is a digital twin system, research into the historical background of the artefact is equally indispensable. This kind of research acts as a kind of "hidden gene" that runs through the entire development process, subtly influencing many of the details of the reconstruction and the design decisions involved. For example, the choice of furniture, decorations and spatial layout within the virtual scenes all need to be grounded in an understanding of the cultural context of that era. Only when the developer has a sufficient understanding of the historical background of the artefact can a VR environment be built that truly immerses the audience.

The technical implementation follows a clearly phased structure. The first is three-dimensional data acquisition, capturing the geometry and surface texture of the original artefact. The second is model processing and scene construction in Blender. The third is the development of the interaction system in Unity. The final step is VR integration and deployment.

This combination of "historical and humanistic research" with "phased technical development" is itself one of the outcomes of this project. Together, these stages form a relatively complete and reusable workflow, which may serve as a reference framework for the development of digital twins of other mechanical cultural relics in the future.

1.4 Structure of This Report

The remainder of this report is organised as follows.

Chapter 2 provides the background to the project. It reviews the historical development and technical principles of the Morse telegraph, discusses the current state of digital twin technology in cultural heritage, and introduces the key technologies used in this project.

Chapter 3 presents the design and implementation of the project. It describes the complete development process, covering 3D data acquisition, model refinement and scene construction in Blender, the development of the interaction systems for both the sending and receiving stations, cross-scene integration, and VR deployment on Meta Quest 3.

Chapter 4 reports on the results and discussion. It presents the final outcomes of the project through screenshots and explanations, illustrating the complete user operation procedures in both the Sending Room and the Receiving Room, and discusses the value, limitations and ethical considerations of the project.

Chapter 5 concludes the report. It summarises the overall outcomes, reflects on the experience

and lessons gained throughout the development process, and outlines directions for further work.

Chapter 2: Background

2.1 Morse Telegraph: History and Technical Principles

2.1.1 Historical Context and Development

In 1832, the American Samuel Morse was inspired by a demonstration of an electromagnet during a voyage, and began to devote himself to the study of telegraphy. By 1837, he had built a working telegraph machine capable of transmitting coded messages. In 1844, with the support of the United States government, a 64-kilometre telegraph line was completed between Washington and Baltimore, and Morse sent the world's first telegram — "What hath God wrought" — marking the beginning of the electrical telecommunications era [1].

Telegraph technology was also introduced to China in the second half of the nineteenth century. In 1870, the Great Northern Telegraph Company of Denmark, backed by Tsarist Russia, extended a submarine telegraph cable to Daji Island off the Wusong estuary without the permission of the Chinese government, becoming the first point of entry for Western telecommunications into China [2]. The Qing government gradually recognised the importance of telegraph communication and began to establish its own telegraph service. On 28 December 1881, the Tianjin–Shanghai telegraph line was opened for public use (see **错误!未找到引用源。**). This was the first long-distance public telegraph line built and operated by China itself, and marked the beginning of China's self-run telecommunications industry [2].



Figure 1: The route map of the Tianjin–Shanghai telegraph line (1881). [2]

In the early twentieth century, wireless telegraphy emerged, and the use of Morse code expanded from wired to wireless communication. With the development of the telephone and modern communication technologies, the telegraph gradually faded from practical use. Nevertheless, the significant historical and educational value of the Morse telegraph has made it an important exhibit in telecommunications museums around the world.

2.1.2 Technical Principles

The Morse telegraph system consists of two main components: a key (also called a manipulator) on the transmitting end and a receiver on the receiving end. The two are connected by wires and work together to encode, transmit and record messages.

Mechanical Structure of the Key

The core of the key is a metal first-class lever, whose fulcrum is connected to one of the wires on the line, as shown in 错误!未找到引用源。. The lever can swing between two contacts: one is connected through a DC battery to the other wire on the line, and the other is connected to the receiver at the local station. In its resting state, a spring keeps the lever handle in the raised position, so that the receiver circuit remains closed. When the operator presses the lever down, the local receiver circuit is disconnected, while the sending circuit is closed through the other contact, sending an electrical pulse to the remote station[4].

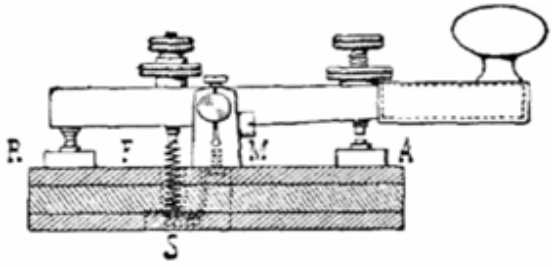


Figure 2: Diagram of the Morse telegraph key. [3]

Mechanical Structure of the Receiver

The receiver involved in this project is a tape receiver. One end of its lever arm serves as the armature of a dual-coil electromagnet, while the other end carries a stylus. In the resting state, the stylus is held by tension springs very close to, but not touching, the surface of the paper tape. The tape continuously unrolls from a drum, driven by a clockwork mechanism and a flywheel, which the operator winds up using a handle. When the electromagnet receives an electrical pulse, it attracts the armature, causing the stylus to tilt and press against the paper tape through an inked roller, leaving a mark. The length of the mark depends on the duration of the

pulse, thus clearly distinguishing dots from dashes on the tape [4].

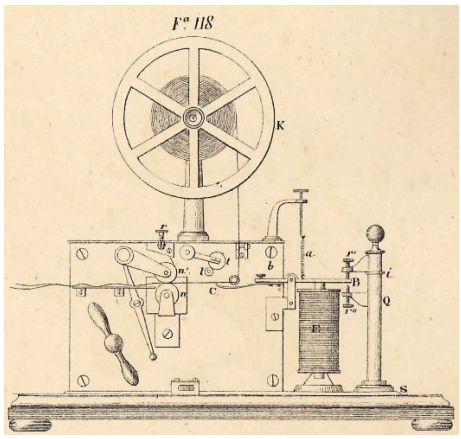


Figure 3: Technical illustration of a Morse tape receiver, from Plate 7 of Telegrafía Práctica. [4]

Encoding Rules and Timing Conventions of Morse Code

Morse code represents letters, numbers and punctuation marks through combinations of two basic signals: the dot ("·") and the dash ("-"). International Morse code covers 26 English letters (A–Z) and 10 digits (0–9), with each character corresponding to a unique dot-and-dash sequence (see 错误!未找到引用源。).

International Morse Code

Character	Morse Code	Character	Morse Code
Letters (A - Z)			
A	· –	N	– ·
B	– · · ·	O	– – –
C	– · – ·	P	· – – ·
D	– · ·	Q	– – · –
E	·	R	· – ·
F	· · – ·	S	· · ·
G	– – ·	T	–
H	· · · ·	U	· · –
I	· ·	V	· · · –
J	· – – –	W	· – –
K	– · –	X	– · · –
L	· – · ·	Y	– · – –
M	– –	Z	– – · ·
Digits (0 - 9)			
0	– – – – –	5	· · · · ·
1	· – – – –	6	– · · · ·
2	· · – – –	7	– – · · ·
3	· · · – –	8	– – – · ·
4	· · · · –	9	– – – – ·

Figure 4: International Morse Code for letters (A–Z) and digits (0–9).

In terms of timing, Morse code uses the duration of a dot as the basic time unit (denoted as 1t). A dash lasts 3t, the gap between signals (dots and dashes) within a character is 1t, the gap between characters is 3t, and the gap between words is 7t (see 错误!未找到引用源。). In the

international standard, the sending speed is measured in WPM (Words Per Minute), and the actual duration of one time unit equals 1200/WPM milliseconds.

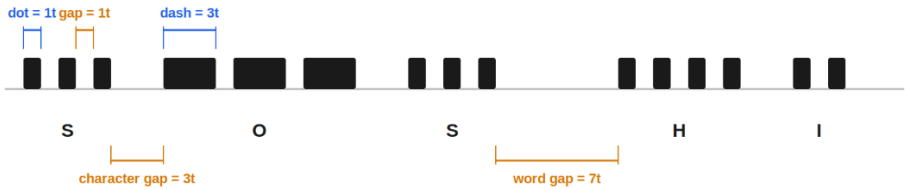


Figure 5: Timing structure of International Morse Code, illustrated with the message "SOS HI".

2.2 Digital Twin Technology in Cultural Heritage

2.2.1 Concept and Current Applications

The concept of Digital Twin was first introduced by Grieves in the field of product lifecycle management. The basic idea is to create a high-fidelity virtual replica of a physical entity through digital means, so that real-time information synchronisation and two-way mapping can be achieved between the physical object and its digital counterpart [5]. This concept was initially applied mainly in aerospace and industrial manufacturing, and gradually became widespread with the development of Industry 4.0. In recent years, it has begun to extend into the field of cultural heritage preservation.

In the cultural heritage domain, the application of digital twin technology has mainly focused on several areas. The first is the high-precision digital recording and preventive conservation of architectural heritage. For example, in the restoration of Notre-Dame de Paris, over one billion data points were captured through 3D laser scanning, and the digital twin model was used to carry out structural analysis of the spire and to optimise the fire protection system [6]. The second area is the virtual display and online service of museum exhibits. Taking the Palace Museum (in Beijing) as an example, researchers have proposed a multi-granularity information organisation approach to provide personalised exhibit recommendations and guided tours [7]. In addition, VR/AR technologies are being increasingly adopted in museum settings. A study by Hao and Wang (2024) [8] showed that more than 80% of visitors were satisfied with the virtual exhibition spaces driven by digital twin technology. In China, institutions such as the Nanjing Museum have started to apply mixed AR-VR technology to exhibition interaction, using methods like virtual scene reconstruction and puzzle-solving games to enhance visitor engagement [9].

2.2.2 Research Gap: Mechanical Cultural Relics

Mechanical cultural relics, such as telegraph machines, antique clocks and gramophones, differ from ordinary artefacts in important ways. As Yang (2024) [10] pointed out, apart from the material and visual properties shared by all cultural objects, mechanical relics also possess working attributes such as power, movement and operation. Their value lies not only in "what they look like", but more importantly in "how they work". However, this functional nature also creates a particular dilemma for their preservation: frequent operation accelerates the wear of mechanical components, while keeping them idle means their core value cannot be demonstrated. As a result, most museums can only place them behind glass for static display [10].

Current research on digital twins has not yet paid sufficient attention to this category of cultural relics, and this gap is precisely the starting point of the present project.

2.3 Key Technologies Used in This Project

2.3.1 3D Scanning with LiDAR

LiDAR (Light Detection and Ranging) is a technology that obtains three-dimensional spatial data of an object's surface by emitting laser pulses and measuring their reflection time. Compared with traditional photogrammetry, LiDAR can capture high-precision point cloud data more quickly and is better at handling complex geometric structures. In recent years, with LiDAR sensors being integrated into consumer devices such as the Apple iPhone Pro series, the barrier to using this technology has been significantly lowered, making it more feasible for the 3D digitisation of small-scale cultural artefacts. In this project, an iPhone equipped with a LiDAR sensor was used together with Apple's Reality Composer application to perform 3D scanning of the physical Morse telegraph in the museum, producing the raw data needed for subsequent modelling.

2.3.2 3D Modelling with Blender

Blender is an open-source 3D modelling software that supports mesh editing, material and texture configuration, animation, and rendering. In this project, Blender was mainly used to clean up, repair and optimise the raw model obtained from LiDAR scanning. This included fixing mesh defects, reducing the polygon count to meet real-time rendering performance requirements, and applying textures and materials to make the model visually match the appearance of the real object as closely as possible.

2.3.3 Interactive Development with Unity

Unity is a widely used cross-platform game engine that supports real-time 3D rendering, physics simulation and user interaction, and is currently one of the most popular platforms for VR application development. In this project, Unity was used as the development platform for the entire interaction system, within which the operating logic of the telegraph key and receiver, the Morse code input and decoding mechanism, and the final VR deployment were all implemented.

2.3.4 Meta Quest VR glasses

Meta Quest 3 is a mixed-reality headset released by Meta in 2023, capable of running VR applications independently without the need for an external PC. Equipped with the Qualcomm Snapdragon XR2 Gen 2 processor, it offers strong graphics rendering performance and can smoothly run VR applications developed in Unity. In this project, Meta Quest 3 was used as the final VR deployment and experience terminal, where users wear the headset and interact with the digital twin scene through its hand-tracking functionality.

Chapter 3: Design and Implementation

3.1 3D Data Acquisition

The starting point of the digital twin workflow is the three-dimensional digitisation of the physical artefact. For this project, the scanning was carried out using the LiDAR sensor built into an iPhone, in combination with Apple's Reality Composer application. This choice was based on two main considerations. First, the project supervisor had previously used the same workflow in several similar digital twin studies with satisfactory results. Second, using an ordinary iPhone made it possible to carry out the scanning without purchasing additional professional scanning equipment, which also made the workflow more accessible and easier to replicate.

Equipment and software

The scanning device used in this project was an Apple iPhone from the Pro series, provided by the supervisor, since the author's own iPhone 13 does not include a LiDAR sensor. The scanning itself was performed through the Reality Composer application running on the iPhone, specifically using its Object Capture feature.

Setting up the scanning environment

The scanning was conducted in a cultural heritage restoration workshop at UPV, as shown in **错误!未找到引用源。**. The room was equipped with a white backdrop, a rotating circular platform, and a set of professional lights, which together provided conditions comparable to those of a photography studio. Before the scanning began, the intensity and angle of the lights were carefully adjusted so that the surface of the artefact was illuminated as evenly as possible, in order to avoid local reflections or shadowed areas that could affect the quality of the captured depth data.



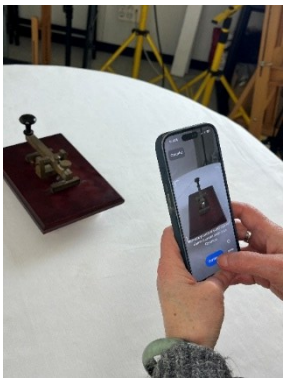


Figure 6: The 3D scanning setup: (a) the studio environment with professional lights and a white backdrop; (b) scanning of the telegraph key using an iPhone with LiDAR.

Scanning the key

The first object to be scanned was the key, which has a relatively simple structure and a smaller size (approximately 20×15×10 cm). In the initial attempt, we kept the iPhone stationary and tried to change the viewing angle by rotating the circular platform on which the key was placed. However, this approach did not produce a valid scan. Only after we changed the method — having the operator hold the iPhone and slowly walk around the stationary key — did the scanning complete successfully.

The scanning procedure itself was as follows. The application first guided the user to enclose the target object within a white semi-transparent cubic bounding box and to set a spatial anchor point around it. The operator then had to move slowly around the object, once at eye level and once from a higher angle, while the application automatically took photographs and captured depth data in real time. Each full scan took roughly 10 to 15 minutes to complete.

Scanning the receiver

After becoming familiar with the full procedure, we moved on to scanning the receiver. Compared with the key, the receiver has a noticeably more complex mechanical structure and a larger body (estimated at around 30×20×35 cm), and is also considerably heavier. Since the receiver is normally displayed inside a glass case in the museum, prior permission had to be requested, and a museum staff member assisted in taking it out of the case for us. This arrangement is itself a reminder of one of the motivations discussed in Chapter 1: mechanical cultural relics in museums are usually not accessible to the public for direct handling and operation, which is precisely the gap that a digital twin system is intended to fill.

Thanks to the experience gained from scanning the key, the scanning of the receiver was carried out by the author alone and proceeded smoothly.

Scan output

At the end of each scan, Reality Composer packaged the result into a compressed file of approximately 200 MB, which contained around seventy high-resolution photographs taken automatically during the scanning process, together with the final three-dimensional model in .usdz format. The raw model produced in this way already captured the general shape and surface appearance of the original artefact, but still contained noticeable topological errors and missing data in certain areas. The next section describes in detail how these problems were

diagnosed and addressed in Blender.

3.2 Model Refinement in Blender

3.2.1 Key Model Refinement

The raw key model exported from ObjectCapture contained approximately 12,494 vertices and 25,000 faces. It preserved the overall geometry of the object and the basic texture of its components, such as the wooden base and the brass lever. However, the model still showed clear defects in geometric accuracy, structural completeness and fine details, and could not be used directly for scene construction or interaction development (see **错误!未找到引用源。** for an overall comparison of the model before and after refinement). The purpose of the refinement stage, though, was not to fix every scanning defect one by one. With limited time and workload, the refinement had to be guided by how the model would later be used in the final VR scene, so that decisions could be made about which parts deserved effort, which parts could be left as they were, and how far each modification should go.

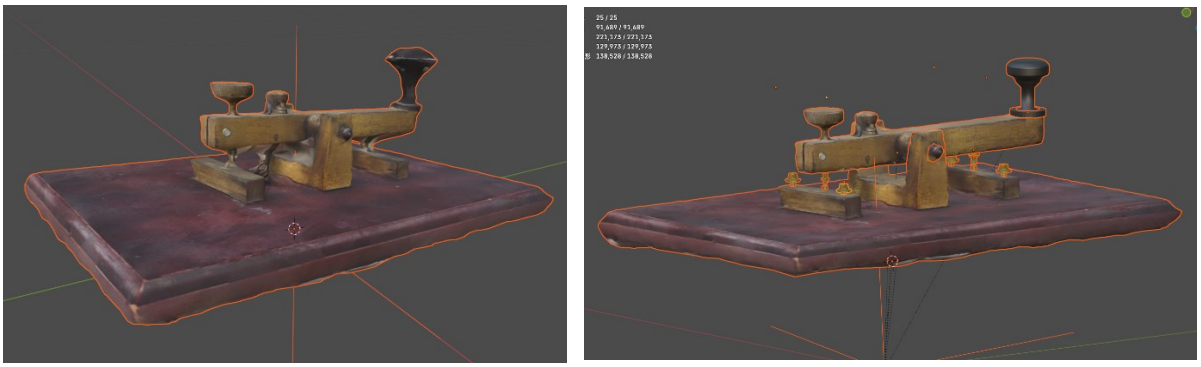


Figure 7: Overall comparison of the key model before and after refinement: (a) the raw scan output from ObjectCapture; (b) the refined model.

In this project, the key model has three key properties, each corresponding to a different level of refinement judgement.

First, the key is the main object placed on the desk in the sending room scene, and it is also the visual focus of the user. At a normal viewing distance, the appearance of the model therefore had to be close enough to the real object that no obvious geometric distortions or scanning artefacts would be visible to the user. At the same time, the key itself is not a large object, and the parts that the user would rarely observe in the VR scene — such as the underside, the interior of the components, or regions hidden by other structures — did not deserve much refinement effort. Based on this judgement, the wave-like deformation on the front of the base

was cleaned up, while the underside and internal details of the model were left untouched.

One specific design decision is worth noting here. On the real object, the connectors that link the key to the external wires are located on the underside, where they would be completely invisible from any VR viewpoint. After discussing this with my supervisor, I decided to move the connectors onto the top surface of the base, adding four small cylinders as visible connection points. This change was not a departure from the original artefact — many Morse keys from the same period place the connectors on the top surface — but rather a deliberate design choice made to serve visibility in the VR scene The newly added connectors, together with the cleaned-up connection area between the lever and the U-shaped bracket, are shown in **错误!未找到引用源。**.



Figure 8: Close-up comparison of the lever–bracket area: (a) the raw scan; (b) the refined version.

Second, the key is not only an object to be looked at, but also an interactive device through which the user sends Morse code. This means that the lever of the key must be able to tilt up and down in response to the user's pressing action, just like a real telegraph key. Based on this requirement, the lever was first separated from the original model, and was then further subdivided and refined. During the separation, the hollow region between the lever and the U-shaped bracket, which had been incorrectly filled in as solid geometry by the LiDAR scan, was also restored: the wrongly filled faces were first deleted, then the two sides were reconnected using Bridge Edge Loops, and the newly generated faces were flattened and adjusted into the correct position.

Finally, the knob at the top of the lever is the part that the user actually touches in VR, and it is also a visual highlight during the interaction. For this reason, the original scan of the knob — which was rough and badly distorted — was not acceptable, and had to be remade to a higher level of visual quality. Similarly, the contact screws and the small cylindrical protrusions at the two ends of the lever were not captured by the scanner at all. Although these

parts are small in volume, they would directly affect how realistic the key looks at a close viewing distance, so they also had to be rebuilt by hand. The actual work included: creating a low-poly cylinder in Blender as the base structure of the contact point; making a custom material for the new parts using a Principled BSDF node mixed with a noise texture, in order to simulate the oxidised look of old brass; and downloading a drawer knob model released under a CC-BY licence from Sketchfab (by the author WolffC [11]) as a replacement, adjusting the proportions of its stem inside Blender to match the size of the original, and assigning it a black bakelite material.

After the refinement, the final key model consisted of 25 separate objects and a total of 91,689 vertices.

3.2.2 Receiver Model Refinement

Compared with the key model, the receiver is larger in size and mechanically more complex. As a result, although the same LiDAR scanning method was applied, the receiver scan produced a noticeably lower-quality result than that of the key. The handle, the body, the internal gears and the paper tape were all topologically merged together, making it impossible to separate them through any automatic operation. More seriously, the paper tape was almost completely fused with the paper-feeding mechanism of the body, and appeared in the scan as a flat, near-zero-thickness surface attached to the surrounding geometry. This meant that the refinement of the receiver could no longer follow the "patching and beautifying" strategy used for the key, and had to shift towards reconstruction: the boundaries between the components needed to be redefined, the key moving parts had to be rebuilt from scratch, and a clear judgement had to be made about which regions were beyond recovery from the scan data and had to be discarded.

The first challenge of the refinement came from component separation. Attempting Mesh → Separate → By Loose Parts in Blender produced no effect at all. As a result, every component that needed to move independently could only be separated by hand. The separation logic followed a single criterion: only the parts that needed their own animation were separated out, while the static portions of the body were left inside the original mesh. Following this principle, the model was eventually divided into the flywheel (Flywheel), the handle (Handle), the body (Mesh) and the rebuilt paper tape as its core components.

The most challenging part of the refinement was the reconstruction of the paper tape. The solution was carried out in two steps. On the body side, a thin rectangular cube was created as a shape reference for the paper-tape channel and placed at the position where the tape was

supposed to pass through; a Boolean Difference modifier was then applied to the scanned body using this cube, "cutting" a clean channel through the body geometry. On the paper-tape side, the scan data was abandoned entirely and the tape was rebuilt from scratch using Blender's procedural curve modelling. A NURBS Path was first created as the central path of the tape, and its control points were adjusted according to photographs of the real object so that the path would pass through the flywheel exit, a curved bending segment, the entrance and exit of the paper-feeding mechanism, and the hanging end. A Bezier Curve was then created as the cross-section of the tape, with the distance between its two endpoints matching the actual width of the tape. Finally, in the geometry properties of the path curve, the cross-section curve was assigned as the bevel object, allowing Blender to automatically generate, along the path, a paper-tape geometry with the correct width and thickness (see **错误!未找到引用源。**).

After the overall modelling of the tape was completed, it still had to be further divided in order to fit the animation requirements of the later Unity stage. When the user actually operates the receiver, the paper tape behaves visually as two distinct parts: the curved segment passing through the paper-feeding mechanism remains relatively static during printing, while the segment hanging out of the machine's exit needs to keep "growing" longer as the reception progresses, simulating the effect of the tape being continuously printed out. If the entire tape were a single object, it would be impossible to apply a stretching animation only to the hanging segment. For this reason, the Bisect tool was used in Edit Mode to cut the whole tape into two halves, which were kept as two independent objects: PaperTape_Static (the static segment inside the machine) and PaperTape_Extending (the extending segment hanging outside the exit). In addition, in order to make sure that PaperTape_Extending would stretch in the correct direction during a Scale animation — that is, growing downwards from the exit rather than expanding from both ends at once — its pivot was deliberately placed at the exit end of the tape. This series of operations prepare for the animation work that would later happen in Unity.

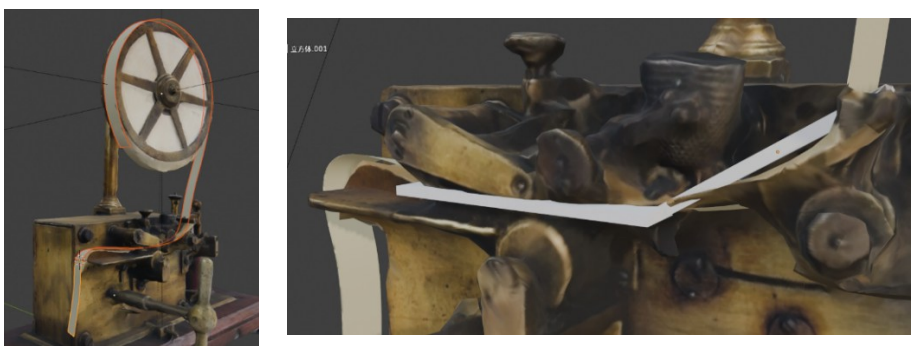


Figure 9: Reconstruction of the paper-tape system. (a) The full paper-tape geometry. (b) Close-up view of

the boolean-cut channel through the body.

Another problem that needed dedicated handling came from the rotation animation of the flywheel. During scanning, the flywheel had a slight tilt in its pose, and its geometric central axis was not aligned with any of the world coordinate axes. As a result, applying a rotation animation directly to the flywheel object would cause it to wobble around an incorrect axis instead of rotating smoothly around its own centre. To solve this, an Empty object (named Flywheel_Pivot) was created at the geometric centre of the flywheel and oriented to match the true rotation axis of the flywheel. The flywheel was then parented to this Empty, and all rotation animations were applied to the Empty itself, with the flywheel following passively as its child. The advantage of this approach is that the geometry of the flywheel does not need to be modified at all; the orientation of the rotation axis is fully controlled by the Empty, which makes future adjustments much more flexible.

After the refinement, the final receiver model consisted of 6 core objects (the body, the flywheel, the handle, the two segments of the paper tape, and the Empty acting as the parent of the flywheel), with a total of 15,118 vertices.

Finally, it is worth noting that the entire refinement process of the receiver clearly demonstrated that scan data alone is not enough to understand the geometry and movement logic of a complex mechanical artefact. When deciding the path of the paper tape, the position of the rollers and the direction of the channel, it was repeatedly necessary to refer back to photographs of the real object before any correct reconstruction decision could be made. This implies that, for the digitisation of complex mechanical cultural relics, physical reference materials and 3D scan data must be used together, and neither one is sufficient on its own.

3.3 Historical Scene Construction

3.3.1 Sending Room

The goal of the sending room scene is to provide the key model with an environment that carries a sense of historical authenticity and cultural context. Based on this consideration, the sending room was set as a typical Chinese telegraph office of the early twentieth century.

The way this setting was reached actually went through a deliberate trade-off. The initial idea was more specific: the scene was to be set at one end of the Tianjin–Shanghai telegraph line that opened in 1881 (the Tianjin telegraph office), because this was the first long-distance public telegraph line built and operated by China itself, and therefore has clear historical significance.

However, while collecting reference materials and planning the room layout, it gradually became clear that fully reconstructing the interior of the "Tianjin telegraph office" as a specific historical location would require an amount of historical research and scene resources well beyond the working budget of this project — for example, the reconstruction of the building structure, the verification of regionally specific furniture styles, and the working routines of telegraph operators in that particular period would all demand a great deal of additional research. In the end, the decision was made to give up the overly specific location setting and to generalise the scene into the more flexible context of "a Chinese telegraph office from the late nineteenth to the early twentieth century".



Figure 10: Reference image of the transmission room (a) Real photos from China Telecom Museum. (b) Staff member of the telegraph office in the 19th century, from L'Univers Illustré, Paris, 1859. [12]

The historical research drew mainly on two kinds of sources: early photographs of telegraph offices made publicly available by the China Telecom Museum, and Western black-and-white illustrations and engravings of telegraph offices from the late nineteenth to the early twentieth century, as shown in **错误!未找到引用源。**. These references helped to define the basic elements that the sending room should contain: the operator's working desk (on which the key is placed), a telegraph line map hung on the wall (using a Qing-dynasty telegraph line map as a visual anchor), brushes, ink and paper for writing telegrams on the desk, a balance scale used for precise weighing, an oil lamp providing the working light, and bamboo paintings as decorative elements that anchor the scene in an East Asian cultural context (see **错误!未找到引用源。**).

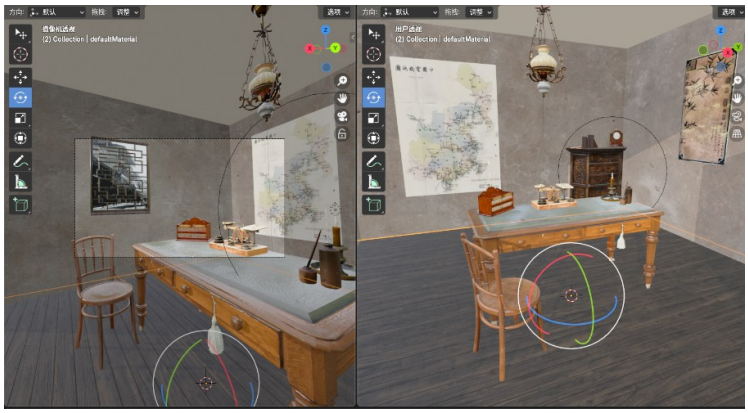


Figure 11: The reconstructed sending room scene in Blender.

The actual objects in the scene came from two different workflows. In the early stage of the project, due to limited familiarity with online 3D model libraries, the approach used was to first search on Pinterest for reference images that matched the historical period, and then to import those images into TRIPO (a free AI-based 3D model generation platform), which would generate the corresponding 3D model. The advantage of this method was a high degree of creative control: any desired interior style could be picked from a vast pool of reference images. However, AI-generated models tended to be unstable in terms of geometric precision and topology, and some of the resulting models contained unclear or poorly defined regions. In the later stage of the project, my supervisor recommended Sketchfab as a more reliable source of assets [11]. Sketchfab hosts a large number of high-quality 3D models uploaded by human modellers (many of them released under the CC-BY licence), and both their geometric quality and their material details were clearly better than those of the AI-generated versions. As a result, scene elements created in the later stage were obtained directly from Sketchfab. This experience reflects a common phenomenon in engineering projects: tool choices made in the early stage of a project are often not optimal, and as familiarity with the surrounding ecosystem grows, the working efficiency improves significantly.

3.3.2 Receiving Room

Compared with the sending room, the receiving room was designed with a clearly different visual identity. Although this project does not explicitly state the specific geographical location of the receiving room in its written narrative, given that the core meaning of telegraph communication lies precisely in "information being transmitted across long distances to another end", the receiving room was deliberately differentiated from the sending room in visual terms: if the sending room presents a traditional Chinese style, the receiving room presents a relatively refined Western style (see **错误!未找到引用源。**).

In terms of the scene construction workflow, the basic architectural structure of the room — including the four walls, the ceiling, the floor, the door and the windows — was modelled directly in Blender using basic geometric primitives (such as cubes). The individual furnishings inside the room were mainly obtained from Sketchfab, while a few pieces of furniture and decorative elements were kept from models generated using TRIPO in the earlier stage of the project [12].

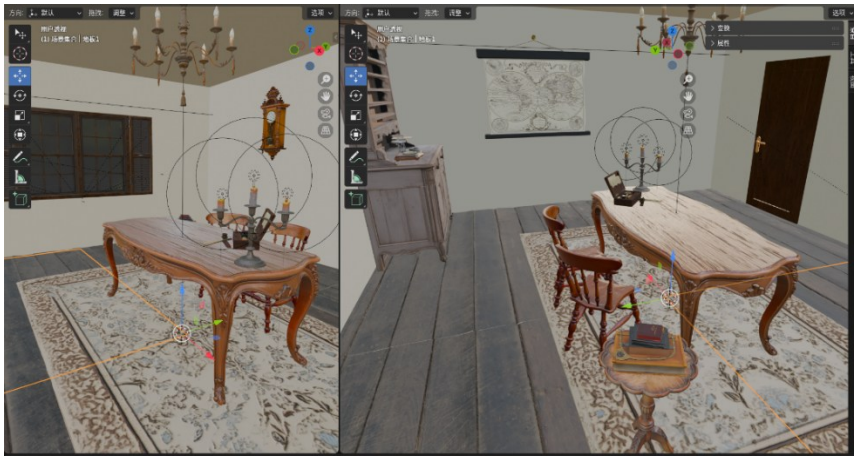


Figure 12: The reconstructed receiving room scene in Blender.

The entire scene was likewise built in Blender and finally exported for use in Unity.

3.4 Interaction System: Sending Station

The interaction system of the sending station is the most technically complex and code-heavy part of the entire digital twin system. It has to translate the user's physical actions on the key in VR into Morse code signals that conform to the international standard, and then, on top of this foundation, provide two very different modes of use: a Training Mode aimed at beginners, and an Operating Mode aimed at actually sending free-form messages. This section describes the overall architecture and key logic of this system, all of which is implemented in C# scripts inside Unity.

The interaction system of the sending station involves five C# scripts.

SimpleLeverPressController is responsible for controlling the pressing animation and input detection of the telegraph key lever. When the user presses the key, it drives the lever to rotate in real time to simulate the pressing effect; when the user releases it, the lever automatically springs back, and the duration of the press is passed to the next layer for signal determination.

SimpleMorseCodeSystem is the core Morse code service, running as a singleton. It receives the press duration from the layer above, calculates timing thresholds based on the WPM (Words

Per Minute) parameter, determines whether the signal is a dot (short press) or a dash (long press), and automatically recognises character boundaries and word boundaries based on the interval between consecutive presses.

TrainingManager implements the complete Training Mode workflow, internally maintaining a four-state finite state machine that guides the user through learning basic Morse code input.

OperatingManager implements the flow control of Operating Mode, managing character commitment, word separation and message sending, and triggering a scene transition once the message has been sent.

MorseSenderUI is the UI controller for Operating Mode, responsible for displaying the current dot-and-dash input and committed character cards, and for connecting the function buttons to the interaction logic of *OperatingManager*.

3.4.1 System Architecture Overview

The interaction system on the sending side was built with a layered architecture with a shared core. From bottom to top, it can be divided into three layers.

The physical layer is implemented by the *SimpleLeverPressController* script. It is responsible for translating the user's physical input (VR controller, mouse or keyboard) into the rotation animation of the lever, and for deciding, when the user releases the key, whether the input just made was a dot or a dash based on how long the key was held down.

The core service layer is implemented by the *SimpleMorseCodeSystem* script. This is a Singleton service that handles all logic related to Morse code itself — including calculating the various timing thresholds based on a WPM (Words Per Minute) speed parameter, maintaining the raw string buffer of the incoming signal stream, automatically inserting character spaces and word spaces based on the time intervals between signals, and detecting input timeouts. This layer broadcasts state changes upwards through C#'s event / delegate mechanism, with events such as OnSequenceUpdated (the raw sequence has been updated) and OnSignalAdded (a new signal has been added).

The upper business layer consists of two parallel manager classes, *TrainingManager* and *OperatingManager*. Both of them attach themselves as event subscribers on top of the core service layer, and each maintains its own state machine and UI logic independently, while sharing the same underlying signal-processing engine.

The value of this layered design is that all logic directly related to Morse code itself needs to

be written only once, and two very different modes of use (Training Mode and Operating Mode) can share the same underlying engine; at the same time, the teaching logic and the operating logic on the upper layer are isolated from each other, so that their UIs and state transitions do not pollute one another.

3.4.2 From Physical Press to Morse Signal: the Core Logic

The timing parameters of the core service *SimpleMorseCodeSystem* are calculated based on the international WPM (Words Per Minute) formula for Morse code. The formula states that one time unit equals 1200 / WPM milliseconds, and on top of this: the maximum duration of a dot is 2 units, the minimum duration of a dash is 2 units, a character space is 2.5 units, and a word space is 5 units. In this project, the default WPM value is set to 10, which corresponds to roughly 120 milliseconds per unit. All of these parameters are exposed in the Unity Inspector, so that they can be adjusted based on user testing experience.

When the user presses the key in VR, *SimpleLeverPressController* records a pressStartTime, and at the same time calls morseSystem.OnKeyPressed() to tell the core service that "the user is currently holding the key down". At this moment, all coroutines responsible for automatic space detection are paused — this step is critical, because if the user holds the key down for longer than the character-space threshold, an incorrect space would otherwise be inserted into the raw sequence. The lever itself uses Mathf.MoveTowards to implement a smooth rotational transition, visually showing the effect of being pressed down.

When the user releases the key, the controller computes the duration of this press: if it is greater than or equal to dashMinTime, the signal is judged to be a dash; otherwise it is judged to be a dot. The controller then calls morseSystem.AddSignal(isDash) to write the signal into the raw sequence. At this point a coroutine called CheckForSpacing is started. It waits for charSpaceTime seconds; if no new key input arrives during this interval, a character space is automatically inserted into the sequence. It then continues waiting up to the wordSpaceTime threshold, and if still no new input arrives, a word space is inserted. Each time a new signal arrives, the previous spacing coroutine is immediately stopped and restarted, so that the automatic segmentation behaviour always remains strictly aligned with the user's actual input rhythm.

3.4.3 Training Mode

The purpose of Training Mode is to allow users who have never encountered Morse code before

to learn basic sending skills in a controlled environment. It is implemented by the *TrainingManager* script, which internally uses a four-state finite state machine to manage the entire training flow.

The four states are: WORD_SELECT, INSTRUCTION, SUCCESS and FAIL. Upon entering Training Mode, the system first shows the WORD_SELECT panel, which offers three pre-defined words for the user to choose from: SOS, HELLO and BYE. After the user clicks on one of these words, the system uses a built-in letter-to-Morse lookup table to generate the standard Morse sequence for that word, and then transitions to the INSTRUCTION state.

In the INSTRUCTION state, the UI simultaneously shows several pieces of information: the spelling of the target word, its corresponding standard Morse sequence, a button that plays a standard audio recording of the word, and a reference display area in which one block per letter is dynamically instantiated from letterBlockPrefab (see **错误!未找到引用源。**). At the same time, the highlight effect on the knob of the key (controlled by the *HandleHighlight* component) is activated in order to visually guide the user towards the part of the key they are supposed to touch. The user then begins to input signals by pressing the lever; every dot or dash that is produced triggers a real-time update of the UI through the OnUserSequenceUpdated event, showing what the user has entered so far. In order to let the user intuitively feel their most recent input, the system also includes a "recent signal" flashing display area, in which a dot appears briefly in yellow and a dash in red before gradually fading out.



Figure 13: The Training Mode UI in the sending room.

The judging mechanism uses a delayed trigger. Whenever the user's input has stopped for longer than judgeDelay seconds (3 seconds by default), the system assumes that the user has finished and automatically performs a judgement. The judging logic strips away all automatically inserted spaces from the user's raw sequence and then compares it, as a string, with the target

Morse sequence (which is also stripped of spaces). If the two match, the state transitions to SUCCESS and the UI shows a success panel offering two options: "train another" (returns to WORD_SELECT) and "go to Operating Mode". If they do not match, the state transitions to FAIL, which briefly displays a failure message and then automatically returns to the INSTRUCTION state for the user to try again, without requiring any manual button press from the user.

The state transitions of Training Mode are illustrated in **错误!未找到引用源。**.

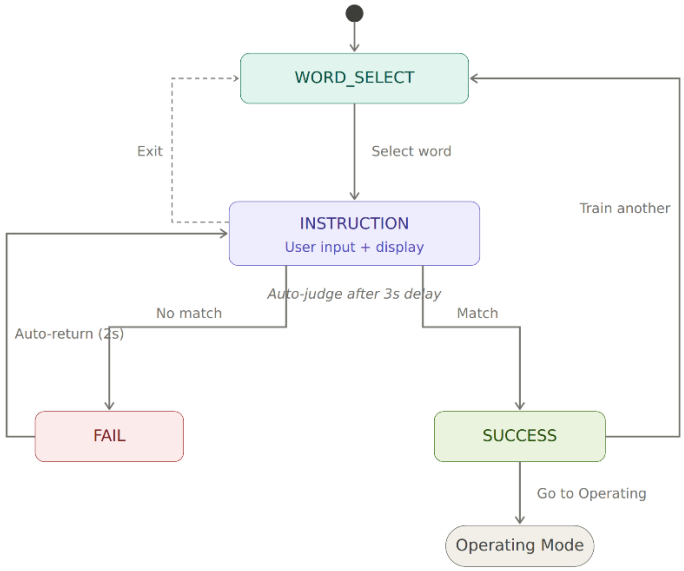


Figure 14: State diagram of the Training Mode finite state machine.

3.4.4 Operating Mode

Operating Mode is the mode in which the user, after completing the training, actually sends free-form messages, as shown in **错误!未找到引用源。**. It is implemented by the *OperatingManager* script. Unlike Training Mode, Operating Mode no longer involves any notion of "right or wrong" — the user is free to send any content, including character combinations that do not exist in the standard Morse code table.

The automatic spacing mechanism in the core service layer works well in Training Mode, because the training words are pre-defined and well-structured, and the user only needs to input each signal following the required timing. But in free-form transmission in Operating Mode, it is almost impossible for the user to precisely control the timing difference between a character space and a word space (the two differ by only 2.5 units, which is roughly 300 milliseconds). Once the input rhythm deviates even slightly, the boundaries of characters and words become unreliable, and the receiving side will then be unable to correctly decode the original text.

To address this problem, Operating Mode introduces a design in which segmentation is explicitly controlled by the user. The UI provides three key buttons — Next Character, Next Word and Send:

- When Next Character is clicked, OperatingManager.CommitCharacter() reads the full content of the current raw buffer, strips away all automatically inserted spaces, and commits the remaining dots and dashes as one complete character into the list of confirmed characters; it then clears the buffer in preparation for the next character.
- When Next Word is clicked, CommitWord() first commits the character currently being entered (if any), and then marks a word-break point after the last entry in the list of confirmed characters.
- When Send is clicked, FinalizeAndSend() packages all confirmed characters and word-break points together, writes them into a static cross-scene data class called SentMessageData, and finally switches to the Receiving Room scene through SceneManager.LoadScene.



Figure 15: The Operating Mode UI in the sending room.

It is worth noting that the automatic spacing mechanism in the core service layer is not disabled in Operating Mode — it still operates normally in the raw buffer display area, so that the user can continue to feel the rhythmic character of a real telegraph transmission. It is only when a character is finally committed that these automatic spaces are removed by a helper function called StripSpaces, so they never appear in the final sent data. This "authenticity at the bottom, user-driven segmentation on top" two-layer design preserves the rhythmic habits acquired in Training Mode while at the same time avoiding the ambiguity problems of free-form transmission.

3.5 Interaction System: Receiving Station

Unlike the real-time input system of the sending station, on the receiving side the user does not need to enter any content. Instead, by interacting with a few key components of the receiver, the user triggers, one stage after another, a predetermined sequence of reception animations and final result display.

The interaction system of the receiving station involves five C# scripts.

ReceivingRoomManager serves as the main controller, managing the four-phase pipeline from message reception to result display.

HandleClickHandler is attached to the receiver handle, detecting click actions from the user and notifying the main controller via event callbacks.

HandleHighlight adds an emissive glow effect to interactable objects such as the handle and the paper tape by modifying the material's emission property, providing visual guidance to indicate which object is currently interactable.

MorseDecoder is a static decoder that converts dot-and-dash sequences into plaintext characters, supporting a total of 36 characters including A–Z letters and 0–9 digits.

ResultPopup controls the display and hiding of the result popup panel, showing the received Morse code sequence and its decoded plaintext message.

3.5.1 System Architecture

The interaction logic on the receiving side is organised as a linear four-phase flow, as shown in **错误!未找到引用源。**. Each phase has a clearly defined start and end, and the completion of one phase automatically triggers the beginning of the next.

The four phases carry out the following responsibilities:

- **Phase 1 - Data Reception:** When the scene loads, the raw Morse sequence passed from the sending side is read from the static data class SentMessageData, and MorseDecoder.Decode() is immediately called to decode it into a plain-text string. This phase is an invisible data preparation stage that the user does not see.
- **Phase 2 - Handle Interaction:** The handle of the receiver is highlighted and a prompt UI is displayed, waiting for the user to click. Once clicked, the handle rotation animation is triggered, visually producing a "winding up" effect.
- **Phase 3 - Receiver Animation:** After the handle rotation completes, the continuous

spinning of the flywheel and the gradual extension of the paper tape are automatically triggered, simulating the process by which a real telegraph receiver "prints" the tape.

- **Phase 4 - Result Display:** Once the paper tape has finished extending, the tape is highlighted and a second prompt UI is shown, waiting for the user to click the tape. After the click, a result window pops up, showing both the raw Morse sequence and the decoded plain text at the same time.

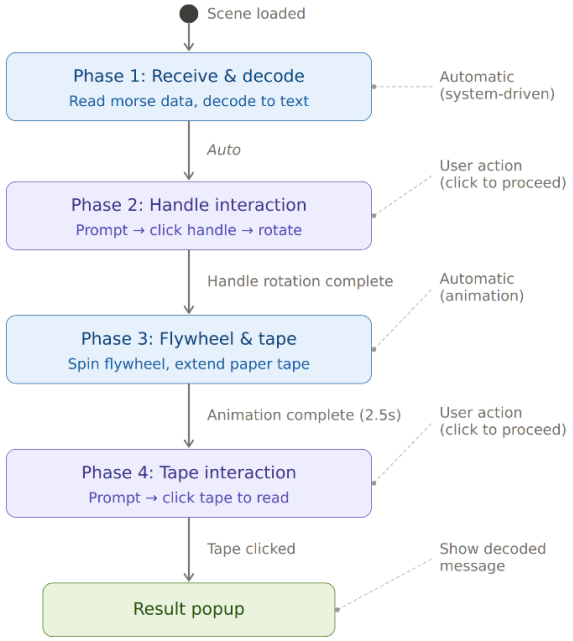


Figure 16: State diagram of the Receiving Room four-phase pipeline.

3.5.2 Receiver Animation

The receiver mainly involves the coordinated animation of three components: the rotation of the handle, the continuous spinning of the flywheel, and the paper tape that keeps "growing" out from inside the machine. The implementation of these animations is built directly on top of the geometric preparations completed in Blender, as described in section 3.2.2.

Handle rotation (Phase 2): After the handle is clicked (see **错误!未找到引用源。**), RotateHandleCoroutine() uses a coroutine to rotate the handle by 360 degrees over a fixed duration. Since the pivot of the handle was already placed at the end that is inserted into the body during the Blender stage, it is now possible to simply call handleTransform.Rotate(axis, delta, Space.Self) to make the handle rotate smoothly around its own axis without any additional transform correction. A "winding up" sound effect (handleAudio.Play()) is played during the rotation to reinforce the physical feedback of the action.



Figure 17: Interaction with handle.

Flywheel rotation (Phase 3): The rotation of the flywheel is implemented by calling the Rotate method on the Flywheel_Pivot empty object that was built in Blender, rather than by directly manipulating the flywheel itself. This is exactly the Unity-side realisation of the engineering solution mentioned in 3.2.2, in which "an empty object was used as the parent in order to solve the problem of a misaligned rotation axis".

Paper tape extension (Phase 3): When the scene loads, the Z-axis scale of PaperTape_Extending is pre-set to a small initial value (tapeHiddenScaleZ, about 0.3), so that visually only a very small portion of the tape is exposed. When Phase 3 begins, a coroutine uses Vector3.Lerp to linearly interpolate the Z-axis scale from this initial value to the final value (tapeFinalScaleZ, about 1.3) over tapeExtendDuration seconds, producing the visual effect of the tape "continuously growing downwards from the exit of the machine".

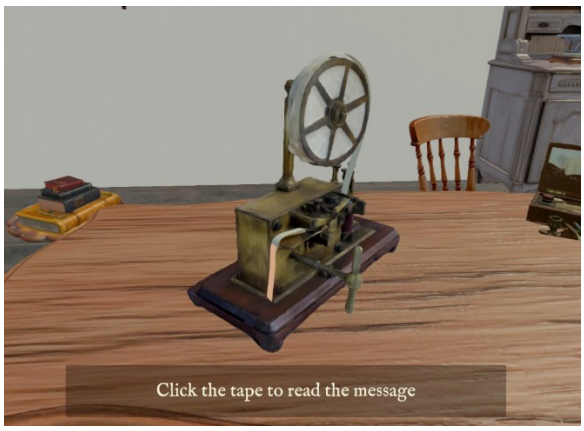


Figure 18: Interaction with paper tape.

These three animations are not independent visual decorations. Instead, they are triggered in a strict order through a chain of coroutines in *ReceivingRoomManager*: the user clicks the handle → the handle rotates for 1.5 seconds → the flywheel and the paper tape simultaneously start a

joint animation that lasts 2.5 seconds → once the animation ends, the system immediately switches to Phase 4 and waits for the user's next interaction (see **错误!未找到引用源。**).

3.5.3 Morse Signal Reception and Display

The interaction-triggering mechanism on the receiving side is much simpler than that of the sending side, since it only needs to respond to two discrete click events: the handle click in Phase 2 and the paper-tape click in Phase 4. These two interactions reuse the same pair of components: *HandleClickHandler* is responsible for detecting click events and broadcasting them through an OnClicked event, while *HandleHighlight* adds a highlight effect on the corresponding object as a visual cue. By attaching the same pair of components to both the handle and the paper tape, the entire receiving-side interaction code avoids duplicating click-handling logic for different objects.

The final result display is handled by the *ResultPopup* component, as shown in **错误!未找到引用源。**. When the user clicks the paper tape, *ResultPopup*.Show(rawMorse, decodedText) is called and a window is opened, showing both the raw Morse sequence and the decoded plain text. Keeping the raw Morse sequence in the result window is a deliberate choice — it allows the user to directly verify that the content they entered on the sending side has been faithfully transmitted and successfully decoded, thereby forming a complete cognitive loop from sending to receiving.

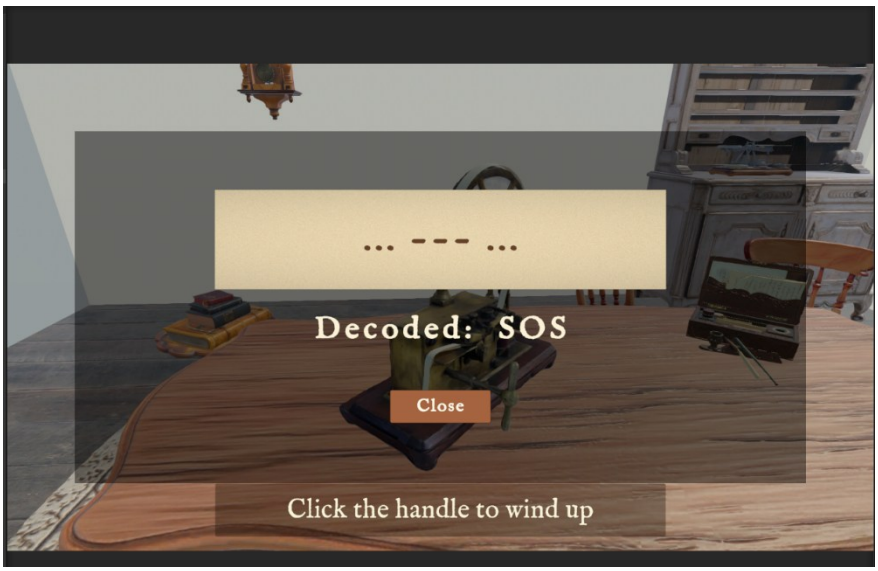


Figure 19: The final result display.

3.5.4 Morse Decoding Mechanism

Morse decoding is the technical pivot that connects the sending and receiving sides. The decoding logic in this project is implemented by an independent static utility class called MorseDecoder. In terms of code size it is remarkably compact — its core consists of nothing more than a 36-entry character lookup table (covering A–Z and 0–9) together with a string-splitting function. This simplicity, however, is made possible by the segmentation mechanism implemented on the sending side.

The Fundamental Problem: Segmentation Ambiguity

Morse code is by nature a time-based signal without any inherent character boundaries. If a stream of dots and dashes is simply handed over to a decoder without further information, the decoder has no way of knowing which specific combination of characters the sequence is supposed to represent, and ambiguity easily arises. For the decoding to be reliable, there must be a mechanism that marks the boundaries between characters and between words within the dot-dash sequence. In a real Morse telegraph system, these boundaries are created by time intervals — a character gap lasts three time units and a word gap lasts seven — but in a scenario where the user enters signals freely through a VR controller, accurately controlling the difference between these two kinds of gaps is practically impossible (see the discussion in 3.4.4).

Three possible solutions to this problem were considered during the design process of this project:

1. **Automatic segmentation based on time intervals:** Relying entirely on the rhythm of the user's input, and using time differences to distinguish characters and words, just like a real telegraph. This approach places too high a demand on the user's precision; in practical tests, users were almost unable to reliably input anything longer than a single word.
2. **AI-assisted decoding:** Using a language model to probabilistically interpret an un-segmented pure dot-dash sequence. This approach is technically feasible, but it introduces a black-box component, and the decoded result may not match what the user actually entered, thereby undermining the "what you see is what you get" quality of the teaching experience. In addition, it cannot guarantee fully correct decoding in all cases.
3. **Character-by-character submission:** Requiring the user to manually confirm every single character as it is entered, so that the system records them one by one. This approach offers the highest decoding reliability, but the user experience is poor: every input has to be followed by a pause and a button click, which completely destroys the natural rhythm

that a sending process should have.

The final choice sits in between options 1 and 3 — in Operating Mode, the automatic-spacing mechanism of the underlying service is kept, so that the user still feels the authentic rhythm while entering each individual character, while at the same time two buttons, Next Character and Next Word, are provided so that the user can actively mark the segmentation boundaries whenever a rhythmic unit comes to an end. This design has already been described in detail in 3.4.4, and its downstream consequence is what MorseDecoder is designed to handle.

The Decoder Design

MorseDecoder is a purely static class and exposes a single public method, Decode(string morseSequence). Its working procedure corresponds strictly to the segmentation convention used on the sending side:

1. The input string is first split into words by sequences of three consecutive spaces;
2. Within each word, the string is then further split into individual character codes by single spaces;
3. Each character code is then looked up in the 36-entry table: if it is found, the corresponding plain-text character is appended to the result; if not, a '?' is appended instead.

The data format produced by OperatingManager.BuildFullSequence() on the sending side corresponds exactly to the input format expected by the decoder: "Next Character" maps to a single space, and "Next Word" maps to three spaces. Every deliberate segmentation action the user performs on the sending side directly determines the character and word boundaries that the decoder on the receiving side will see.

At the same time, MorseDecoder deliberately does not introduce any fault tolerance or ambiguity-handling mechanism: any character code that cannot be recognised is simply replaced with a '?' and shown in the result. This "let the mistake be" behaviour is intentional — it allows the user to experience the telegraph process as faithfully as possible.

3.6 Integration and Cross-Scene Connection

The Unity application for this project contains two independent scenes — the Sending Room and the Receiving Room. In Unity, when a scene transition is triggered through SceneManager.LoadScene, all MonoBehaviour instances in the current scene are destroyed. A cross-scene data transfer mechanism is therefore needed to ensure that the Morse sequence generated on the sending side can be accurately passed to the receiving side.

Sections 3.4.4 and 3.5.4 have already described, respectively, how the sending side packages its data and how the receiving side parses it. SentMessageData is the data carrier that connects these two ends. It is designed as a static class that does not inherit from MonoBehaviour, and is therefore not destroyed when a scene is unloaded. This approach was chosen mainly because the cross-scene data in this project is one-directional, single-use and structurally simple, making a static class the most lightweight and straightforward solution.

In terms of scene flow, the system provides two transition points. The first is the Send button in Operating Mode: when the user clicks it, OperatingManager writes all committed characters and word-break positions into SentMessageData, and then calls SceneManager.LoadScene to switch to the Receiving Room. The second is the New Message button in the result popup of the Receiving Room: after viewing the decoded result, the user can click this button to return to the initial interface of the Sending Room (the word-selection panel of Training Mode), thereby re-experiencing the complete sending and receiving process from the beginning.

This integration mechanism is relatively simple in terms of technical implementation, but it connects the sending system and the receiving system — which were previously developed independently — into a complete, cyclable user experience loop.

3.7 VR Integration and Deployment

With the interaction system fully developed, the next step was to deploy the entire system onto a VR headset, so that users could operate the telegraph using hand tracking in a virtual reality environment.

Framework selection

The project originally planned to use Unity's XR Interaction Toolkit for VR interaction. However, during the actual configuration process, persistent compatibility issues were encountered, particularly when trying to get button-touch interactions working in VR.

After spending a considerable amount of time debugging without success, the decision was made to switch to the Meta XR SDK (version 85.0.0), which is provided directly by Meta and offers more thorough support for Quest-series devices. After the switch, the basic VR interaction setup was completed smoothly within a much shorter time. The final system uses OVRCameraRig in place of the original XR Origin as the main camera, and integrates the OVRInteractionComprehensive prefab from the Meta XR Interaction SDK, which contains all the interaction components needed for hand tracking (see **错误!未找到引用源。**). The system

ultimately supports hand tracking only, without the use of handheld controllers.

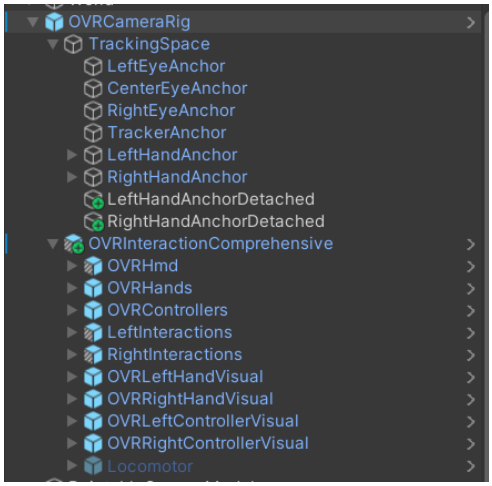


Figure 20: The Hierarchy view of OVRCameraRig and OVRInteractionComprehensive in Unity.

Interaction adaptation

To adapt to the VR environment, two additional C# scripts were developed.

KnobPressDetector is used for press detection on the telegraph key, checking the distance between the user's index fingertip and the knob on every frame; when the distance falls below a set threshold, a press is triggered, enabling natural interaction through hand tracking.

ProximityClickDetector uses the same distance-detection approach, attached to the handle and the paper tape in the Receiving Room, to detect click actions on 3D objects.

When migrating the system from desktop to VR, all existing interaction methods had to be re-implemented. The OnMouseDown and Input.GetKeyDown calls used on the desktop side do not work at all in an Android/VR environment, so two different replacement approaches were adopted to cover the three types of interaction in the system.

For pressing the telegraph key, the system checks, on every frame, the distance between the user's index fingertip (HandIndexFingertip) and the knob of the key. When this distance falls below a set threshold (see **错误!未找到引用源。**), StartKeyPress() is triggered; when the finger moves away, EndKeyPress() is triggered, thereby simulating the experience of physically pressing a real telegraph key.

For clicking 3D objects in the Receiving Room (the handle and the paper tape), the same distance-detection approach was used. A custom *ProximityClickDetector* script was attached to both the Handle and the PaperTape_Extending objects, firing an OnClicked event whenever the

fingertip distance meets the required condition, replacing the original mouse-click detection.

For UI button interaction, all Canvas elements were set to World Space mode and placed at fixed positions within the scene. The interaction itself relies on the Poke Interaction component chain provided by the Meta SDK, including PokeInteractable, PointableCanvas and ClippedPlaneSurface working together, which allows users to press virtual buttons directly with their fingers.

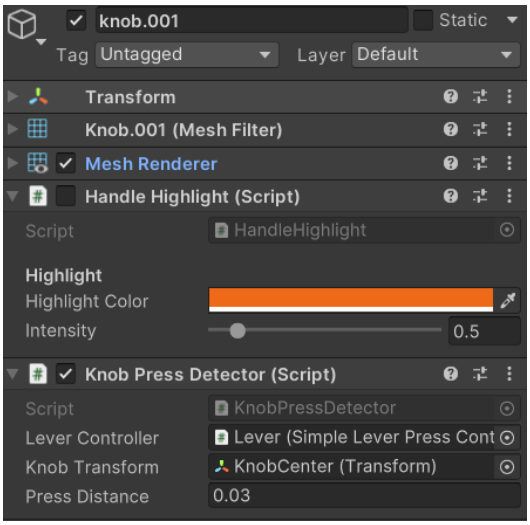


Figure 21: The Inspector view of the telegraph key knob.

Build and deployment

The target device was a Meta Quest 3. Before building, the platform in Unity's Build Settings was switched to Android, and developer mode was enabled on the Quest 3. After connecting the headset via USB, Build and Run was used to compile, install and launch the application directly on the device. Each successful Build and Run generates an APK installation package. This file can be distributed and installed on any compatible Meta Quest device independently of the development environment, facilitating further sharing and demonstration.

During the deployment process, since the development computer's hardware was insufficient to simulate the VR interface on the PC side, every test had to be carried out by connecting the headset via USB and using Unity's Build and Run to compile and install the application directly onto the Meta Quest 3. This process was time-consuming, with each full-scene build taking approximately four to five minutes. To improve development efficiency, following the supervisor's suggestion, a lightweight version containing only the core interaction components — the telegraph key, the receiver and the UI — was built first, reducing each build time to one to two minutes and allowing the interaction logic to be debugged and verified as a priority.

Once the interaction workflow had been fully tested, the project moved on to the environment integration stage. However, after incorporating the complete historical scenes, severe frame drops were observed on the VR headset. Investigation revealed that the scene polygon counts far exceeded the rendering capacity of the Meta Quest 3 — the Sending Room contained approximately 6.24 million triangles and the Receiving Room approximately 9.77 million, while the recommended upper limit for the Quest 3 is around 750,000. These high polygon counts originated mainly from 3D models generated by Tripo AI from images during the early stage of the project [12]. At that time, the rendering limitations of VR devices had not yet been considered, and higher-polygon versions were preferred in pursuit of greater visual realism. To address this issue, the Decimate modifier in Blender was used to selectively reduce the polygon count of these models, ultimately bringing the two scenes down to approximately 300,000 and 440,000 triangles respectively — both well below the 750,000 upper limit — and achieving smooth performance, as shown in Figure 22.

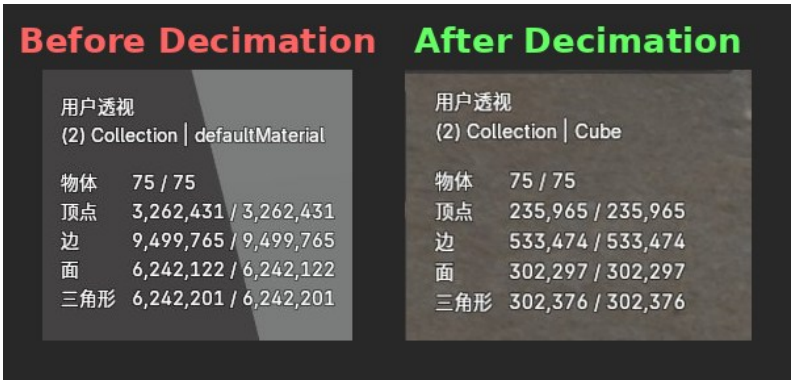


Figure 22: Comparison of the Sending Room statistics before and after polygon decimation in Blender.

Chapter 4: Results and Discussion

This chapter presents the final outcome of the project. Sections 4.1 and 4.2 walk through the complete operation procedures in the Sending Room and the Receiving Room respectively, from the user's perspective, illustrated with screenshots taken from the running system. Section 4.3 then evaluates the project against its original objectives, discusses the value of the workflow, and reflects on the current limitations. Figure 23 shows a user with headset is operating the key in sending room.



Figure 23: A user operating the digital twin system while wearing the Meta Quest 3 headset.

4.1 Sending Room: Walkthrough of Operation

When the application launches, the user is placed inside the Sending Room — a 19th-century style interior furnished with wooden furniture, oil lamps and period-appropriate wall decorations. A Morse telegraph key sits on the desk at the centre of the scene, and it is the main object the user will interact with throughout this room, as shown in Figure 24.



Figure 24: The Sending Room environment as seen by the user upon entering the scene.

Training Mode

The interaction begins in Training Mode, which is designed to help first-time users learn the basics of Morse code input before they attempt free-form messaging. A word selection panel is displayed, offering three predefined words — SOS, HELLO and BYE (see Figure 25).

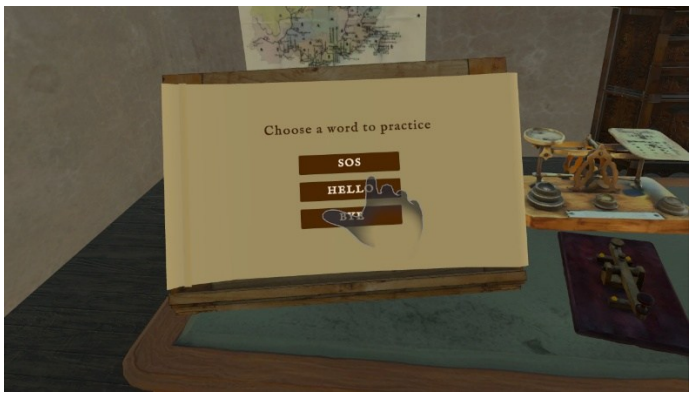


Figure 25: The Training Mode word selection panel.

After selecting a word, the system enters the instruction view, as shown in Figure 26. The upper part of the screen displays the target word along with its complete Morse code sequence, and a "Play Standard Audio" button allows the user to listen to the correct rhythm. Below this, the dots and dashes that the user is currently inputting by pressing the telegraph key are shown in real time. An "Exit Training" button in the lower-left corner allows the user to return to the word selection panel at any time. The knob of the key is highlighted with an orange glow to guide the user towards the correct component to touch.



Figure 26: The instruction view after selecting "SOS".

The user then presses the virtual telegraph key using hand tracking: short presses produce dots and longer presses produce dashes. After the user stops inputting for three seconds, the system automatically judges whether the input matches the target sequence, as shown in Figure 27. If the input is correct, a success panel appears with two options: "Train Another Word", which

returns to the word selection screen, and "Go to Operating Mode", which advances the user to free-form messaging. If the input is incorrect, a brief failure message is shown, and the system automatically returns to the instruction view for another attempt.



Figure 27: (a) The success panel after a correct input; (b) the failure message before automatic retry.

Operating Mode

Once the user enters Operating Mode, they are free to compose and send any message, guided by a Morse code chart displayed in the lower-left corner of the screen. The telegraph key remains the primary input method, but the UI changes considerably (see Figure 28).

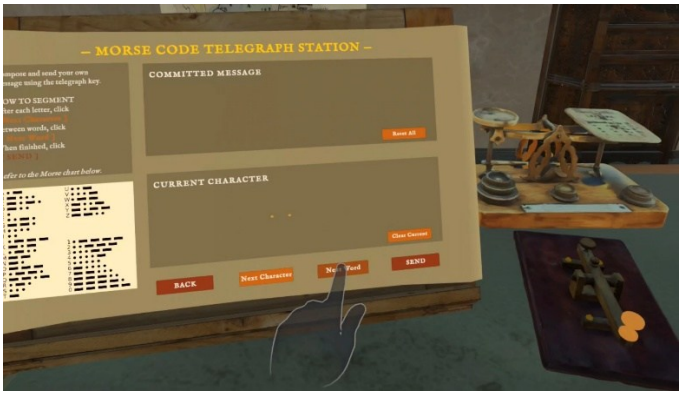


Figure 28: The Operating Mode UI.

The typical workflow in Operating Mode proceeds as follows. The user taps out the dots and dashes for one character, which appear in the "Current Character" area, and then presses "Next Character" to confirm it. This process is repeated for each subsequent character. When a complete word has been entered, the user presses "Next Word" to insert a word boundary (see Figure 29). All confirmed characters and word breaks are shown in a separate "Committed Message" area, allowing the user to review the message before sending. When the message is ready, the user presses "SEND", which writes the complete Morse sequence into the cross-scene data carrier and immediately transitions to the Receiving Room. Alternatively, pressing "BACK" returns the user to the word selection screen of Training Mode.

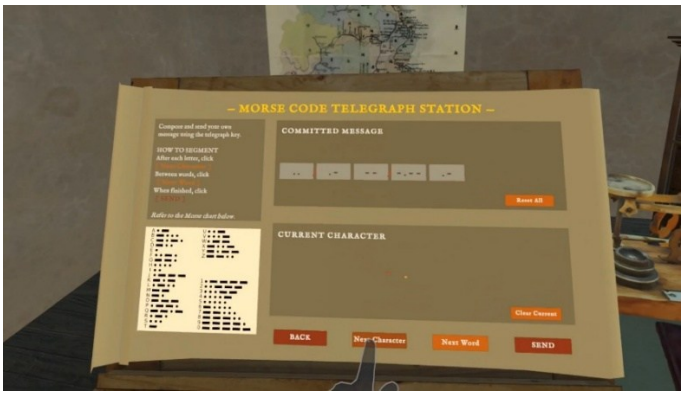


Figure 29: A message being composed in Operating Mode, with several characters already committed.

4.2 Receiving Room: Walkthrough of Reception

After the user presses "Send" in the Sending Room, the scene automatically transitions to the Receiving Room, where a Morse telegraph receiver sits on the desk (see Figure 30).



Figure 30: The Receiving Room environment as seen by the user upon entering the scene.

Handle Interaction

Once the scene has loaded, the handle of the receiver is highlighted with an blue glow and a prompt message appears on screen, guiding the user to click the handle (see Figure 31). After the user touches the handle through hand tracking, the handle begins a rotation animation that simulates the process of an operator winding up the clockwork mechanism of a real receiver, accompanied by a mechanical sound effect.



Figure 31: The handle highlighted with a prompt message during Phase 2, waiting for the user to click.

Flywheel and Paper Tape

After the handle rotation completes, the flywheel of the receiver begins to spin continuously and the paper tape gradually extends from inside the machine. The entire animation lasts approximately 2.5 seconds and is accompanied by the sound of the flywheel in motion.

Tape Interaction and Result Display

Once the paper tape has finished extending, the newly extended section of the tape is highlighted, and a new prompt message appears on screen, guiding the user to click the tape to read the message. After the user touches the tape, a result window pops up: the upper part displays the raw Morse code sequence, and the lower part shows the decoded plaintext. When the user clicks the "New Message" button at the bottom, the scene switches back to the word selection screen of Training Mode in the Sending Room, allowing the user to begin sending a new message.

Once the paper tape has finished extending, the newly extended section of the tape is highlighted, and a new prompt message appears on screen, guiding the user to click the tape to read the message (see Figure 32). After the user touches the tape, a result window pops up: the upper part displays the raw Morse code sequence, and the lower part shows the decoded plaintext. When the user clicks the "New Message" button at the bottom, the scene switches back to the word selection screen of Training Mode in the Sending Room, allowing the user to begin sending a new message (see Figure 33).



Figure 32: The paper tape highlighted and waiting to be clicked during Phase 4.



Figure 33: The result popup showing the Morse sequence and the decoded plaintext.

4.3 Discussion and Reflection

The project has essentially achieved all the specific objectives set out in Chapter 1. The only departure from the original plan was the choice of VR framework — due to persistent compatibility issues encountered during configuration, XR Interaction Toolkit was replaced by the Meta XR SDK, and the interaction method was changed from handheld controllers to hand tracking. The following two sections discuss the reusable value of the project's workflow and the limitations of the current system, respectively.

4.3.1 The Value of a Reproducible Workflow

Beyond the digital twin system itself, another outcome of this project is the validation of a complete and reproducible development workflow.

This workflow covers every stage from physical scanning to VR deployment: the scan produces a .usdz model → the model is refined in Blender and exported as .fbx → scenes are constructed and interactions are developed in Unity → the application is built as an .apk and deployed to the headset. The pipeline is linear and clearly structured, making it straightforward to reproduce step by step.

In terms of hardware and software, the entire pipeline maintains a low entry barrier. The scanning was performed with an iPhone equipped with a built-in LiDAR sensor, rather than a professional-grade 3D scanner; model processing was carried out in Blender, which is free and open-source; interaction development was based on the free Personal Edition of Unity; and the deployment target was a consumer-grade VR headset. This means that other researchers or museum professionals can attempt to reproduce similar projects without needing to purchase prohibitively expensive professional equipment.

More importantly, this workflow is not tied to any specific type of object. If the scanning target were replaced from the Morse telegraph to another mechanical cultural artefact — for example, the Changxin Palace Lamp or an ancient astronomical instrument — the methods used in the scanning, model refinement and scene construction stages would remain entirely applicable; only the interaction logic would need to be redesigned according to the functional characteristics of the new object.

Furthermore, VR deployment is not the only possible output of this workflow. If the VR integration stage were removed, the same set of scenes and interaction systems could also be built as a desktop application or deployed to the web via WebGL, further lowering the barrier to access and allowing users without VR equipment to engage with the digital twin content.

4.3.2 Limitations

Although the system has largely achieved its intended functional objectives, a number of limitations remain.

1. **No formal user testing was conducted for this project.** All functional verification was carried out by the developer alone, and no real museum visitors or other external users were invited to experience and evaluate the system. As a result, the system lacks objective, quantitative data to support claims about interaction fluency, learning curve and user experience.
2. **The precision and richness of hand-tracking interaction are limited.** The current system detects pressing and clicking actions by calculating the distance between the user's index fingertip and the target object. This distance-threshold approach occasionally produces false triggers or missed inputs in practice. Moreover, in the Receiving Room, actions such as winding up the handle and removing the paper tape would involve rotational gestures and grasping motions in a real setting, but the current system simplifies them to click-based triggers. Introducing gesture recognition to reproduce these actions

more faithfully would enhance the sense of immersion.

3. **The manual segmentation mechanism in Operating Mode sacrifices naturalness.** Although it effectively resolves the decoding ambiguity problem, a real telegraph operator relied on rhythmic timing to distinguish character and word boundaries, whereas this system requires the user to explicitly mark boundaries by pressing UI buttons, which interrupts the immersion of continuous tapping.
4. **The current system is a single-user, one-directional experience.** The user completes both the sending and receiving operations alone, whereas real telegraph communication in history involved two operators working collaboratively at the sending and receiving ends respectively. The system does not currently implement any multi-user online functionality to recreate this collaborative scenario.
5. **There is room for improvement in accessibility and engagement features.** All UI text in the current system is provided in English only, with no support for switching to Chinese, Spanish or other languages, which is less than friendly for non-native English speakers. In addition, incorporating a simulated per-word charging mechanism into Operating Mode could not only add an element of fun to the experience, but also help users better understand how telegraph services were actually used in their historical context.

Chapter 5: Conclusion and Further Work

5.1 Conclusion

This project has successfully developed an interactive digital twin system for a Morse telegraph and deployed it on the Meta Quest 3 VR headset. Users are able to experience the complete process of encoding, sending, receiving and decoding a telegraph message within a virtual environment.

The system demonstrates two key points. First, the value of mechanical cultural artefacts lies not only in what they look like but, more importantly, in how they work, and an interactive digital twin is an effective approach that can simultaneously satisfy the seemingly contradictory needs of preservation and hands-on experience. Second, the complete workflow from physical scanning to VR deployment can be achieved using consumer-grade hardware and free software, making it feasible to extend and reuse for other artefacts of a similar nature.

5.2 Reflection

5.2.1 Ethical Considerations

A digital twin is essentially a simplified reconstruction of the original object, not a perfect copy. Therefore, when displaying or disseminating digital twin content, its nature as a digital reconstruction should be clearly stated, so as to avoid giving audiences the impression that it is an exact replica of the original artefact. In addition, the scanning work for this project was carried out with the permission of the cultural heritage restoration workshop at UPV, respecting the informed consent and requirements of the artefact's custodians.

5.2.2 Social Impact

From a social perspective, the direct value of this project lies in lowering the barrier to cultural education. Traditional static museum displays are constrained by geographical location and physical space, whereas a VR digital twin allows people who cannot visit a museum in person — whether due to geographical distance, physical condition or other reasons — to operate a cultural artefact hands-on and understand how it works, gaining an experience close to that of handling the real object.

From a broader point of view, this project is also an exercise in interdisciplinary integration. It builds a bridge between cultural heritage conservation and engineering: the development

process simultaneously involved research in the humanities and history (understanding the historical context and usage of the telegraph), art and design (the visual style and atmosphere of the scenes), communication principles (the technical rules of Morse encoding), and computer science (Unity development and VR deployment). These seemingly unrelated fields converged naturally under the single goal of making a cultural artefact operable once again. This also reflects a growing trend: future work in the digitisation of cultural heritage will increasingly require multi-disciplinary collaboration rather than being confined to a single field.

5.2.3 Legal Considerations

The telegraph scanned in this project belongs to the collection of the Telecommunications Museum at UPV, and the use of the digitised model should respect the rights of the collection holder. All software used during the development of the project was either free or educationally licensed, with no violation of software licensing terms. All historical images and technical illustrations cited in the report have been properly attributed to their original sources.

5.2.4 Environmental Considerations

The digital twin is itself an environmentally friendly way of displaying cultural artefacts. It reduces the physical wear caused by frequent exhibition and handling of real objects, while also providing remote users with a means of experiencing artefacts without the need for long-distance travel. The hardware energy consumption during development was also modest — scanning required only an iPhone, development was carried out on an ordinary personal computer, and no large-scale servers or high-performance computing resources were involved.

5.2.5 Personal Reflection

This project was my first experience of completing the entire pipeline from physical 3D scanning to VR application deployment. Prior to this, I had never used Unity or Blender, and VR development and deployment were entirely unfamiliar territory. Learning and getting to grips with multiple new technologies within a limited timeframe was the greatest challenge this project presented to me. At the same time, however, the computer science foundations built up over four years of undergraduate study proved their value at key moments: familiarity with programming enabled me to quickly understand Unity's scripting system and state machine design; experience with object-oriented programming helped me organise the code structure in a logical way; and the debugging and troubleshooting skills developed through coursework provided support during unexpected situations such as the VR framework migration.

In addition, I have always had a strong interest in museums and cultural heritage, but this had previously been limited to visiting museums as a member of the public, without any hands-on involvement in cultural heritage digitisation projects. This development experience has given me a much deeper understanding of how technology can be applied in the service of cultural heritage preservation, and has also opened up a new perspective on my future career direction — the intersection of culture and technology may well be a field worth continuing to explore.

5.3 Further work

Future work could be pursued in the following directions.

1. The most important next step is to conduct formal user testing. Users from diverse backgrounds — such as museum visitors, school students and telecommunications students — could be invited to experience the system, and their feedback on educational effectiveness and user experience could be collected through questionnaires, interviews and behavioural data logging. The results would directly inform subsequent improvements to the interaction design and content.
2. The same workflow could be applied to other mechanical cultural artefacts to verify its universality. For example, the Telecommunications Museum at UPV holds other types of early communication devices; if the process developed in this project were repeated with those objects, it would both test the applicability of the workflow to different artefacts and gradually build up a collection of digital twins that could be exhibited together as a series.
3. Deployment formats beyond VR could be explored. As discussed in Section 4.3.1, the same set of scenes and interaction systems could be deployed to the web via WebGL, allowing users without VR equipment to access the experience through a browser. This would significantly expand the system's potential audience and make it easier for museums to embed the content in online educational platforms or official websites.
4. The integration of the digital twin with physical museum exhibitions could be explored. For example, a VR experience station could be set up alongside a real display case, allowing visitors to operate the digital twin immediately after viewing the physical artefact, creating a complementary "observe and operate" exhibition format.

References

[1] Ruan, Y.; Li, J. "A brief discussion on the composition principles and development applications of Morse code" (in Chinese), China Equipment Engineering, no. 9, pp. 269–271, 2023.

[2] China Telecommunications Museum, "A Century of Telecommunications, Red Heritage: Exhibition of the History of China's Telecommunications Industry" (百年电信 红色传承——中国电信业发展史陈列), permanent exhibition, Beijing, China. [Visited: November 2025]

[3] EB3EMD, "Historia de la Telegrafía" (in Spanish), [Online]. Available: http://www.ea1uro.com/eb3emd/Telegrafia_hist/Telegrafia_hist.htm [Accessed: 19 April 2026].

[4] F. P. Blanca, *Telegrafía Práctica* (in Spanish). Madrid, 1881. [Online]. Available: https://museotelecomvlc.webs.upv.es/wp-content/uploads/2020/07/Telegrafiapractica.pdf [Accessed: 19 April 2026].

[5] M. Grieves, *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*. New York: McGraw-Hill, 2006.

[6] E. Hedrick, D. Hatch, and E. Oliver, "Digital Twins in the Ancient and Historical World," in *2025 Intermountain Engineering, Technology and Computing (IETC)*, 2025. DOI: 10.1109/IETC64455.2025.11039432

[7] X. Li, J. Fan, W. Peng, M. Chang, X. Wang, and Y. Wang, "A Digital Twin Model for Museum Exhibit Services" (in Chinese), *Library and Information Service*, vol. 69, 2025. [Online]. Available: https://link.cnki.net/urlid/11.1541.G2.20260409.1835.008

[8] X. Hao and Y. Wang, "Application of Digital Twin and Meta-Universe Technology in Museum Exhibition Design," in *2024 Int. Conf. on Telecommunications and Power Electronics (TELEPE)*, pp. 262–265, 2024.

[9] K. Peng, "Research on interactive application of museum exhibition based on AR-VR mixed technology" (in Chinese), *Identification and Appreciation to Cultural Relics*, no. 2, pp. 65–68, 2026.

[10] Yang, X. "Classification of mechanical cultural relics in museum collections and reflections on the trends of standardised development" (in Chinese), *Natural Science Museum Research*, vol. 9, no. 5, pp. 81–88, 2024.

[11] Sketchfab, "Sketchfab — The best 3D viewer on the web," [Online]. Available: https://sketchfab.com [Accessed: April 2026].

[12] "Telegraph office worker forwarding a message in Morse Code, 19th century," illustration from *L'Univers Illustré*, Paris, 1859. [Online]. Available: https://www.alamy.com, Image ID: 2JCGX20 [Accessed: September 2025]

[13] Tripo AI, "Tripo — AI 3D model generator," [Online]. Available: https://www.tripo3d.ai [Accessed: April 2026].

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor for the opportunity to be part of this project, for the substantial assistance during the initial scanning phase, and for the patient and thorough guidance throughout the entire process. Whenever I encountered difficulties, practical and constructive suggestions were always provided. This supervision has been a truly enjoyable and motivating experience, and it also sparked my deep appreciation for the vibrant city of Valencia.

I would also like to thank the Vicente Miralles Segarra Museum of Telecommunications and the Heritage and Art Collection Office (Área de Fondo de Arte y Patrimonio) at UPV for their support of this project, providing access to valuable historical artefacts and scanning facilities.

My thanks also go to the China Telecommunications Museum, whose rich exhibition content deepened my understanding and appreciation of the project during the early research phase.

In addition, I would like to express my gratitude to the Universitat Politècnica de València, Queen Mary University of London, and Beijing University of Posts and Telecommunications for giving me the opportunity to undertake this overseas final year project, which has broadened my horizons significantly.

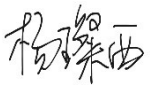
Finally, I am deeply grateful to my parents for their unwavering support, and to all my family and friends for their help and companionship.

Appendices

Disclaimer

This report is submitted as part requirement for the undergraduate degree programme at Queen Mary University of London, and Beijing University of Posts and Telecommunications. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

BUPT No.:2022213370
QMUL No.:221168420
Full Name (Pin Yin):Yang Canxi
Full Name (Chinese):杨璨西

Signature: 

Date:26-04-2026

Acknowledgement of using Generative AI

I acknowledge the use of the following Generative AI tools

Publisher	**Name, version, URL**	**Use**	**Section**
DeepSeek	DeepSeek, https://www.deepseek.com/	Used to search and gather reference materials during the background research phase.	Chapter 2, Section 2.1.1 and Section 2.2.1
Anthropic	Claude,Opus 4.6, https://claude.ai/	Used to organise ideas and structure the report, proofread and polish the English writing, translate between Chinese and English.	Chapters 1–5
Tripo AI	Tripo AI, https://www.tripo3d.ai/	Used to generate 3D models from reference images for historical scene construction.	Chapter 3, Sections 3.3.1 and 3.3.2

I confirm the submitted work is my own work.

Project specification

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 1 - Supervisor

论文题目 Project Title	Development of a Digital Twin of a Morse Telegraph		
题目分类 Scope	Research	Data Science and Artificial Intelligence	Software
主要内容 Project description	Project Objectives: Digital Twin of a Morse Telegraph Station Develop a Digital Twin of a historical Morse telegraph station, including both the transmitter (key) and the receiver (printing mechanism), to accurately replicate its structure and functionality in a virtual environment. Create 3D Models of the original telegraph components by scanning physical artifacts housed at the Telecommunications Museum, ensuring high fidelity to the original objects. Process and Optimize Models using Blender for accurate geometry, textures, and materials, preparing them for real-time use in interactive applications. Implement Interactivity using Unity, allowing users to interact with the telegraph components, simulate message transmission, and observe the internal mechanical processes. Enable VR Compatibility with Meta (formerly Oculus) headsets, providing a fully immersive experience for users to explore and operate the telegraph station in virtual reality. Serve Educational Purposes by making the digital twin accessible to students, educators, and the general public, helping them understand the principles and historical importance of Morse code communication. Preserve Cultural Heritage by digitally archiving an operational model of the telegraph station, contributing to the conservation of technological heritage and making it accessible beyond physical limitations.		
关键词 Keywords	Telecommunications, Technological Heritage, History		
主要任务 Main tasks	1 1.Develop a Digital Twin of a historical Morse telegraph station.		
	2 2.Create 3D Models of the original telegraph components by scanning physical artifacts housed at the Telecommunications Museum.		
	3 3.the student is required to Process and Optimize Models using Blender.		
	4 4.the student is required to Implement Interactivity using Unity.		
主要成果 Measurable outcomes	1 1.An interactive, high-fidelity digital twin of a Morse telegraph station, accessible in virtual reality using Meta headsets.		
	2 2.A preserved and documented 3D archive of historical telegraph equipment, created from accurate scans of museum artifacts.		
	3 3.An immersive educational experience that enhances public understanding of early telecommunications technology and its cultural significance.		

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 2 - Student

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	Yang	**名 First Name**	Canxi		
BUPT 学号 BUPT number	2022213370	**QM 学号 QM number**	221168420	**班级 Class**	2022215112
论文题目 Project Title	Development of a Digital Twin of a Morse Telegraph				
论文概述 Project outline **Write about 500-800 words** **Please refer to Project Student Handbook section 3.2**	**1. Project Background and Objectives** This project aims to develop a Digital Twin of a historical Morse telegraph station, accurately replicating the structure and functionality of both its transmitter (key) and receiver (printing mechanism) in a virtual environment. The core objective is to create a high-fidelity, interactive virtual model for educational purposes and cultural heritage preservation. By utilising advanced 3D scanning, modelling, and game engine technologies, this project will transform a physical artifact housed at the Telecommunications Museum into a dynamic digital asset accessible and operable in virtual reality, thereby overcoming the limitations of traditional physical displays. **2. Problem Analysis and User Requirements** Currently, many historically significant technological devices, such as the Morse telegraph station, are difficult to display and teach effectively to students and the public due to physical space constraints and the fragility of the artifacts. This project addresses this issue. The target users include students at various educational levels, educators, and the public interested in telecommunications history. The core user need is the ability to intuitively understand the working principles of the Morse telegraph and to immersively experience the process of receiving and transmitting messages first-hand. **3. Technical Methodology and Implementation Path** The project will follow a systematic technical process, divided into the following key stages: **3.1 3D Data Acquisition and Modelling** LiDAR technology will be used to scan the physical telegraph equipment in the museum to obtain a high-precision 3D representation, laying the foundation for creating a high-fidelity digital model. **3.2 Model Processing and Optimisation** Subsequently, the models will be processed and optimised using Blender. This step involves cleaning the scanned data, repairing meshes, optimising polygon count for real-time rendering performance, and applying accurate textures and materials to achieve high visual fidelity. **3.3 Interactive Logic and System Development**				

Within the Unity game engine, a virtual scene will be constructed, and the optimised 3D models will be imported. The core development will be accomplished using the C# programming language, encompassing two main parts: firstly, implementing animation for the moving parts of the telegraph components (e.g., the key, rotating disk, paper tape) to simulate the entire process of message encoding, transmission, reception, and decoding; secondly, designing and implementing the user interaction interface and logic, such as the two modes for training and free transmission, input feedback, and some engaging design elements.

3.4 VR Integration and Immersive Experience

Finally, the project will be optimised and built for Meta series VR headsets. Using Unity's XR Interaction Toolkit, natural user interaction within the virtual reality environment will be enabled—for example, allowing users to "press" the key using controllers to send Morse code and observe the real-time response at the receiver end, providing a fully immersive operational experience.

4. Expected Outcomes and Significance

The direct outcome of this project is a fully functional, educationally robust Digital Twin of a Morse telegraph station. It will be an operable, interactive simulation system. More significantly, the complete development workflow explored and implemented in this project—from the 3D scanning of the artifact and model optimisation to the interactive logic design and VR integration in Unity—forms a reusable, standardised workflow.

Its profound significance lies in a dual contribution: on a societal level, it preserves and showcases technological heritage in an innovative way, allowing precious artifacts to transcend physical limitations and be accessed and understood by a wider audience, thereby contributing to the education and dissemination of the history of technology and telecommunications. On a technical methodology level, it provides a valuable reference template and proof of feasibility for developing Digital Twins of other types of technical artifacts in the future, offering broadly applicable insights for the digital preservation of cultural heritage.

5. Reference

[1] Tao F, Xiao B, Qi Q, Cheng J, Ji P. Digital twin modeling. J Manuf Syst. 2022;64:372-89. DOI: https://doi.org/10.1016/j.jmsy.2022.06.015.

[2] History.com Editors. Morse Code & the Telegraph [Internet]. History.com. 2019 Jun 6 [cited 2024 Nov 15]. Available from: https://www.history.com/news/telegraph-morse-code

[3] Museo de la Telecomunicación Vicente Miralles Segarra. Telégrafo Morse: El manipulador [Internet]. Universitat Politècnica de València; [date unknown] [cited 2024 Nov 15]. Available from: Provided by project supervisor.

[4] Museo de la Telecomunicación Vicente Miralles Segarra. Telégrafo Morse: Receptor telegráfico de código Morse de cinta impresa [Internet]. Universitat Politècnica de València; [date unknown] [cited 2024 Nov 15]. Available from: Provided by project supervisor.

[5] Pérez Blanca F. Elementary Treatise on Practical Telegraphy [Internet]. Madrid: [publisher unknown]; 1881 [cited 2024 Nov 15]. Available from: https://museotelecomvlc.webs.upv.es/wp-content/uploads/2020/07/Telegrafia-practica.pdf

道德规范 **Ethics** **Please discuss ethical issues with your supervisor.** **Please refer to Project Student Handbook section 4.1 and Project Ethics Guidance**	Please confirm by checking the box: ☒ I confirm that I have discussed ethical issues with my supervisor.
	Summary of ethical issues: This project involves human participants for user testing. The use of Artificial Intelligence (AI) tools in the project's development has also been considered. **1. Concerning Human Participants:** This project involves user testing of the VR Digital Twin application. I have reviewed the ethics checklist for human participants with my supervisor, and we confirm the following measures will be implemented: **Informed Consent:** A written consent form will be provided to all participants involved in the formal testing phase, clearly explaining the project's purpose, procedures, and data usage. **Right to Withdraw & Contact Details:** Participants will be explicitly informed of their right to withdraw at any time without penalty and will be provided with the investigator's contact details. **Debriefing:** Participants will be debriefed about the project upon completion of their session. **Data Anonymity:** All feedback and data collected from participants will be stored securely and anonymized for analysis. **Future Public Exhibition:** If the application is made available for public demonstration, users will be presented with a simplified information notice outlining the project's nature and data usage practices. Other than the points above (corresponding to items 3, 9, 10, 11, 12 in the checklist), this project does not involve any other ethical issues (e.g., the use of non-standard hardware, deception, participants under the age of 16, etc.). **2. Concerning Artificial Intelligence (AI):** Generative AI tools (such as large language models) were used in an assistive capacity during the development of this project, primarily for tasks including: 1. Translating historical Spanish documentation. 2. Providing general guidance on code structure and report writing. These tools were used as productivity aids, and their output was critically reviewed and integrated by the author. The core functionality of the Digital Twin (e.g., 3D modelling, interaction logic, Morse code translation) was implemented directly by the student. The use of these tools will be transparently acknowledged in the final report.

中期目标 **Mid-term target.** **It must be tangible outcomes, E.g. software, hardware or simulation.** **It will be assessed at the mid-term oral.**	**1.Two Distinct Virtual Stations:** one Transmitter Station deploying the telegraph key, and one Receiver Station deploying the printing mechanism. **2.Core Interactive Animations:** the core components of both key and receiver will have interactive animation, including the lever movement of the key and the corresponding rotation of the receiver's wheel with paper tape. **3.Preliminary User Interaction Framework:** Implementation of a preliminary user interaction framework, including a basic UI for mode switching and the basic logic to support user input in both "Training Mode" and "Free Transmission Mode".

Work Plan (Gantt Chart)

Fill in the sub-tasks and insert a letter X in the cells to show the extent of each task

	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-28	Mar 1-15	Mar 16-31	Apr 1-15	Apr 16-30
Task 1 Develop a Digital Twin of a historical Morse telegraph station												
Background Research and Requirements Analysis	X											
System Design and Technical Planning	X	X										
System Integration and Validation Testing										X	X	X
Task 2 Create 3D Models of the original telegraph components by scanning physical artifacts housed at the Telecommunications Museum												
Scanning Preparation and Data Acquisition						X	X					
Preliminary Point Cloud Data Processing							X	X				
Task 3 the student is required to Process and Optimize Models using Blender												
Post Office Scene Modeling: Model the environmental assets (e.g., desk, walls, windows)	X	X	X									
Telegraph Asset Processing: Clean, optimize, and texture the scanned telegraph models							X	X				
Scene Assembly and Final Polish: Combine all assets into a cohesive scene and set up lighting								X	X			
Task 4 the student is required to Implement Interactivity using Unity												
Scene Import, Configuration, and Prefab Setup	X	X						X	X			
Component Animation development			X	X	X	X			X	X		
User Interface and Interaction Mode Development	X	X	X		X				X	X		
VR Integration and Optimization										X	X	

5

Early-term progress report

北京邮电大学 本科毕业设计（论文）初期进度报告

Project Early-term Progress Report

学院 **School**	International School	专业 **Programme**	**Internet of Things Engineering**		
姓 **Family name**	Yang	名 **First Name**	Canxi		
BUPT 学号 BUPT number	2022213370	**QM 学号 QM number**	221168420	班级 **Class**	2022215112
论文题目 **Project Title**	Development of a Digital Twin of a Morse Telegraph				

I. Literature Review

1.Research Status of Digital Twin Technology in Cultural Heritage Protection and Educational Virtual Asset Development

Since its proposal, digital twin technology has been maturely applied in fields like intelligent manufacturing and smart cities. In recent years, its value in cultural heritage protection and the development of educational virtual assets has gradually become prominent, acting as a key technical path to assist the display and inheritance of cultural relics.

In the field of high-precision reproduction of cultural heritage, scholars have formed a technical framework of "data collection - model construction - physical-virtual interaction". In the digital practice at the Beit Lehi archaeological site in Israel and the village of Lagkadia in Greece, Hedrick et al. adopted AI-assisted drones and LiDAR scanning technology to collect point cloud data in underground limestone environments 20-25 feet deep. The built digital twin model has millimeter-level accuracy, which can support the detailed restoration of architectural repairs (such as the mosaic floor textures of Byzantine churches) [4]. Similarly, in the Notre-Dame de Paris restoration project, the research team obtained over one billion data points through 3D laser scanning. Combined with the digital twin model, they completed the mechanical analysis of the spire and the optimization of the fire protection system. This proves the advantage of digital twin in the "preventive protection" of cultural heritage — predicting risks through virtual simulation to avoid secondary damage to cultural relics caused by physical intervention[6]. Li Xin et al. further proposed the "5D scene continuum" framework for Cultural Heritage Digital Twin (CHDT), integrating 3D models, spatio-temporal information and context (such as historical events and humanistic backgrounds). This breaks the limitations of traditional static models and endows cultural relics with "historical context". For example, in the digital twin model of the Grand Canal cultural heritage, by superimposing the canal transportation scenes of the Ming and Qing dynasties, the virtual assets have both "protection" and "education" values[3].

In the development of educational virtual assets, the core value of digital twin technology lies in breaking "time and space constraints" and "cognitive barriers". Taking museum cultural relics as the research object, Tang Xuejun et al. proposed an interaction strategy of "emotional narration + multi-sensory immersion". Through the digital twin model, they restored the craft details of cultural relics (such as the separate casting technology of the Square Zun with Four Sheep). Combined with Unity real-time rendering and VR headsets, users can "disassemble" virtual cultural relics to understand their structural logic. This model turns abstract historical knowledge into operable immersive experiences[1]. In the practice of tourist exhibition halls, Sui Yi further verified the educational effect of VR integration — real-time pushing cultural relic background information (such as the historical use of Morse telegraph) through the VR navigation system. The user participation rate has increased by more than 40% compared with traditional exhibition boards, proving the potential of digital twin in "active learning"[2].

2. Research on Morse Telegraph

As the core invention of the 19th-century telecommunications revolution, the historical value and mechanical structure of Morse telegraph have formed a relatively complete theoretical system. However, existing display methods are still limited to traditional physical exhibitions, making it hard to balance "protection" and "experience".

Historically and technologically, the development of the Morse telegraph depended on breakthroughs in electromagnetism and technological integration. Inspired by Ampère's electromagnetic experiment in 1832, Morse developed the first electromagnetic telegraph in 1837 with Alfred Vail and Leonard Gale. Its core innovation was the combination of a "code system" and "mechanical structure": Vail's American Morse code encoded letters via the time difference of "dots, dashes, and spaces" (e.g., "·-" for A), solving the key problem of digital information transmission. Mechanically, early telegraphs consisted of a manipulator (transmitting end) and a receiver (receiving end): the manipulator used a lever to control circuit on-off (pressing duration determined "dot" or "dash"), while the receiver relied on electromagnet suction to drive a stylus marking paper tape, with a clockwork mechanism powering continuous tape movement[7]. In 1844, the 64-kilometer Washington-Baltimore telegraph line launched, and the first telegram "What hath God wrought" marked humanity's entry into the "instant communication" era. Later refined by Friedrich Clemens Gerke, Morse code became an international standard in 1865, widely adopted in military, shipping, and other fields[8].

However, existing display methods of Morse telegraph are still dominated by physical exhibitions, with three core limitations. Firstly, space constraints. Physical cultural relics need to be displayed in fixed showcases, so users cannot observe mechanical details up close. Moreover, the exhibition coverage is limited, making it difficult to open to audiences in multiple places at the same time[4]. Secondly, fragility of cultural relics: the metal components of Morse telegraph are prone to rust due to temperature and humidity changes. Frequent physical contact will accelerate wear and tear, and most museums only allow "distant viewing" and prohibit operation[7]. Thirdly, single experience: existing exhibitions mostly introduce telegraph functions through text descriptions or video demonstrations. Users cannot personally experience the complete process of "transmitting - receiving telegrams", making it hard to understand the connection between code encoding logic and mechanical transmission, and the educational effect is limited.

3. Research Status of Key Technologies

(1) 3D Scanning Technology

3D scanning is the foundation for the high-fidelity construction of digital twin models. LiDAR technology has become the preferred solution for cultural heritage data collection due to its "high precision and anti-interference" characteristics. In the scanning of the village of Lagkadia in Greece, Hedrick et al. (2025) used consumer-grade drones equipped with LiDAR devices. Through a 60%-80% image overlap rate, they obtained point cloud data of the 0.18-hectare Byzantine church site under complex terrain (952 feet elevation difference) with an accuracy of 5mm, which can clearly restore stone textures and architectural structure details[4]. Similarly, in the Changting ancient city digital twin project, the team recorded the carved patterns and wood textures of Ming and Qing buildings at the millimeter level through the combination of 3D laser scanning and UAV oblique photography, providing "precision navigation" for model restoration[3]. In the scanning of mechanical cultural relics, LiDAR technology can effectively capture tiny structures (such as the contact gap of the Morse telegraph manipulator and the gear pitch of the receiver).

(2) Blender Modeling Optimization

As an open-source 3D modeling software, Blender is widely used in cultural heritage model optimization, focusing on two core needs: "mesh lightweight" and "texture authenticity". In terms of mesh repair, Jiang et al. proposed a "hierarchical modeling method based on Discrete Event

System (DES)", decomposing complex mechanical structures into a hierarchy of "components - sub-assemblies - total assemblies". For example, the Morse telegraph is decomposed into the "manipulator - receiver - line" subsystem. The mesh topology is optimized through Blender's "repair tools" (such as deleting redundant vertices and filling holes), reducing the number of model polygons from millions to 100,000 levels to ensure the smoothness of subsequent Unity real-time rendering[5]. In terms of texture mapping, Tang Xuejun et al. proposed a "material sampling - texture baking" process. The surface textures of cultural relics (such as rust marks on telegraph metal components and wood grains on wooden handles) are collected through high-definition cameras. The textures are accurately mapped to the model surface through "UV unwrapping" in Blender. At the same time, Physically Based Rendering (PBR) technology is used to simulate metal reflection and wood diffuse reflection effects, improving the visual fidelity of the model[1].

(3) Unity Interaction Development

Unity is the core platform for the interaction development of cultural heritage digital twins. Its C# programming and animation system can realize "mechanical function simulation" and "user operation feedback". In terms of logical interaction, Zheng et al. realized "component motion correlation" through C# scripts in the digital twin of the helicopter transmission system — such as gear rotation driving shaft transmission. This logic can be migrated to Morse telegraph: writing a linked script of "key pressing - circuit on-off - electromagnet attraction - stylus motion". In terms of animation implementation, Sui Yi proposed a method combining "keyframe animation + physical engine". In the virtual cultural relics of tourist exhibition halls, the motion trajectories of mechanical components (such as the rotation speed of the telegraph paper tape and the swing angle of the lever) are set through Unity's "Animator" component. At the same time, the "Rigidbody" component is used to simulate physical collisions (such as the contact pressure between the stylus and the paper tape), improving the authenticity of operation[2].

(4) VR Integration

VR integration is the key to realizing immersive operation experience. XR Interaction Toolkit has become the mainstream solution due to its "cross-device adaptation" and "interaction logic visualization". In the interaction of museum cultural relics, Tang Xuejun et al. realized "accurate mapping between VR controllers and virtual cultural relics" through this toolkit — users hold VR controllers to simulate the pressing of telegraph manipulators. The haptic feedback of the controller (vibration intensity corresponds to pressing force) is synchronized with the motion of the virtual model. At the same time, "XR Ray Interactor" is used to realize the zoom-in view of cultural relic details (such as observing electromagnet coils up close) [1]. In the Greek stone carving project, Hedrick et al. (2025) further combined VR with spatial audio. When users operate the virtual telegraph, the "tick-tock" sound of electric current and mechanical transmission sound are played in real time, enhancing the multi-sensory immersion[4].

4. Identification of Research Gaps: Limitations of Existing Literature and Core Value of This Project

(1) Lack of Digital Twin Research on Mechanical Cultural Heritage

Existing digital twin technologies in cultural heritage are mostly focused on architectural heritage (like ancient cities and churches) or static cultural relics (such as ceramics and calligraphy paintings). There is very little research on "mechanical cultural heritage" (especially 19th-century telecommunications equipment). Retrieving keywords such as "Digital Twin for Mechanical Artifacts" and "Mechanical Artifact Digital Replica" shows that existing literature mostly focuses on the twinning of industrial machinery (such as helicopter transmission systems and machine tools) [5], and does not involve the "historical feature restoration" and "educational function adaptation" of "mechanical cultural heritage". In addition, there is a lack of dynamic twin methods for "movable mechanical components". For example, the motion damping of telegraph levers and

the rotation synchronization of paper trays. Most existing models adopt simplified animations, which cannot accurately simulate the motion laws of real machinery[8].

(2) Existing Cultural Heritage Digitization Focuses More on Static Display and Lacks Immersive Operation Experience

Although digital twin technology has achieved high-precision reproduction of cultural heritage, most studies still stay at the "static browsing" level (such as 360° model viewing and scene roaming), lacking "functional operation experience". For example, the Notre-Dame de Paris digital twin model is mainly used for "structural repair simulation", and users cannot conduct functional interaction with the virtual building[6]. The Changting ancient city digital twin focuses on "spatial planning" and does not open immersive operation permissions to ordinary users[3].

Reference

[1]唐学军,胡玉娇.基于数字孪生技术的博物馆文物展陈交互设计策略研究[J].艺术家,2025,(09):32-34.
[2]睢毅.VR 技术在旅游展馆数字交互中的应用研究[J].漫旅,2024,11(24):37-39.
[3] L. Xin, G. Hongyu, S. E. Kyeong, W. Qitao, Y. Guojun and D. Bangkun, "Towards Cultural Heritage Digital Twin: Concept, Characteristics, Framework and Applications," 2023 9th International Conference on Virtual Reality (ICVR), Xianyang, China, 2023, pp. 549-557, doi: 10.1109/ICVR57957.2023.10169702.
[4] E. Hedrick, D. Hatch and E. Oliver, "Digital Twins in the Ancient and Historical World," 2025 Intermountain Engineering, Technology and Computing (IETC), Orem, UT, USA, 2025, pp. 1-6, doi: 10.1109/IETC64455.2025.11039432.
[5] Jiang, H. F., Qin, S. F., Fu, J. L., Zhang, J., & Ding, G. F. (2021). How to model and implement connections between physical and virtual models for digital twin application. Journal of Manufacturing Systems, 58, 36-51. https://doi.org/10.1016/j.jmsy.2020.05.012
[6] Zheng, M. L., & Tian, L. (2022). A hierarchical integrated modeling method for the digital twin of mechanical products. Machines, 10(1), 1-19. https://doi.org/10.3390/machines10010002
[7] Blanca, F. P. (1881). Elementary Treatise on Practical Telegraphy.
[8] Martín, M. C. B., & Mora, M. D. R. (2016). The beginnings of telecommunications in the Valencian community.

II. Summary of Progress

1. Completed Tasks

(1) Background Research and Requirements Analysis

- Conducted an in-depth literature review focusing on three core themes: the application of digital twin technology in cultural heritage protection, the mechanical structure and historical background of Morse telegraphs, and key technologies (LiDAR scanning, Blender modeling, Unity interaction, VR integration). Integrated Chinese and English core literatures to clarify the current technical status and research gaps.
- Visited the China Telecom Museum to understand the history of telegraphs after their introduction to China and experience Morse telegraph sending interaction.
- Collected structural drawings, historical materials, and cultural relic photos of Morse telegraphs, as well as reference photos of historical post office scenes.
- Through user needs analysis, identified three core requirements:
  ① High-fidelity reproduction of Morse telegraph's mechanical components (manipulator, receiver) and historical post office scenes;
  ② Implementation of immersive VR interaction (simulating the full process of telegraph sending and receiving);

③ Educational functions (supporting Morse code learning and mechanical principle demonstration).

(2) System Design and Technical Planning

- Completed the overall system framework design, and drew flowcharts and state machines for main programs, covering two modes of the sending scene and the receiving scene.
- Determined technical tools and version compatibility: Blender 4.5 for modeling, Unity 2022.3 for interaction development, XR Interaction Toolkit 2.4 for VR integration, and Meta Quest 2 as the testing device to ensure compatibility and stability of all tools.

(3) Model Processing and Optimization Using Blender

- Completed the modeling of core environmental assets for the historical post office scene, including:
  ① Functional assets: 19th-century-style wooden desk (suitable for telegraph placement size), writing supplies, supporting chairs, balances, and other scene-specific items;
  ② Scene assets: Decorative paintings, maps, clocks, windows, and floors on the walls.

2. On-Going Tasks

(1) Component Animation Development

- Development objectives: Realize animations for key movable components of the Morse telegraph to restore real mechanical movement logic:
  ① Manipulator animation: Key pressing/rebound (simulating the lever principle, adding damping effect to restore real pressing feel);
  ② Receiver animation: Electromagnet attraction (driven by "circuit on-off" signals), stylus up/down movement (synchronized with electromagnet action), and paper tray rotation (driven by a clockwork mechanism, matching the stylus marking rhythm).
- Current progress: Completed the keyframe animation drafting for the manipulator pressing action.

(2) User Interface and Interaction Mode Development

UI Design

- Completed the preliminary prototype design of the VR UI interface, including: Morse code reference table (pop-up on demand), operation prompts (real-time guidance for key pressing);

Interaction Mode Development

- Defined the logic of two core interaction modes:
  ① Training mode: The system randomly generates text, users send corresponding Morse code via the virtual telegraph, and the system provides real-time correct/incorrect feedback;
  ② Free mode: Users freely press the telegraph keys, and the receiver synchronously outputs paper tape marks (dots/dashes) and plays the "tick-tock" current sound.
- Current progress: Completed the preliminary C# script framework for the training mode.

是否符合进度? On schedule as per GANTT chart?

YES

下一步 Next steps:

1. Scan physical cultural relics to complete the refined modeling of the key and receiver.
2. Complete component animation development and correct mechanical movement deviations.
3. Finish UI and interaction mode development
4. Promote the integration of the telegraph model and the post office scene, import the animated telegraph model into the scene, adjust the placement position and scene proportion to ensure reasonable spatial logic.
5. Supplement and optimize scene details, such as adding historical pictures to simulate outdoor scenes outside the windows.

Mid-term progress report

北京邮电大学 本科毕业设计（论文）中期进度报告

Project Mid-term Progress Report

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	Yang	**名 First Name**	Canxi		
BUPT 学号 BUPT number	2022213370	**QM 学号 QM number**	221168420	**班级 Class**	2022215112
论文题目 Project Title	Development of a Digital Twin of a Morse Telegraph				

是否完成任务书中所定的中期目标？Targets met (as set in the Specification)?

Yes.

Refinement and animation of receiver model is under developing, which required more time than planned after careful analysis of its complex mechanical structure.

All other sub-tasks have been accomplished in line with the project plan.

Plan to catch up: Receiver model refinement and animation development will commence immediately after the mid-term oral presentation, with a target completion date of 15 March 2026.

已完成工作 Finished work:

The following summarises the work completed from early term to mid-term stage, covering 3D data acquisition, model processing, scene construction, and interactive system development.

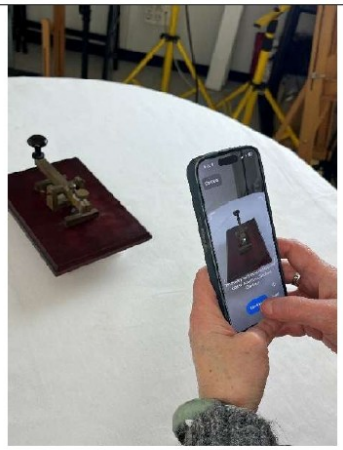
1. 3D Data Acquisition

We used iPhone LiDAR sensor and the Reality Composer application to do the physical scanning of both telegraph components from UPV's Telecommunications Museum.

A studio environment was prepared with professional softbox lighting and a white backdrop to ensure consistent illumination.



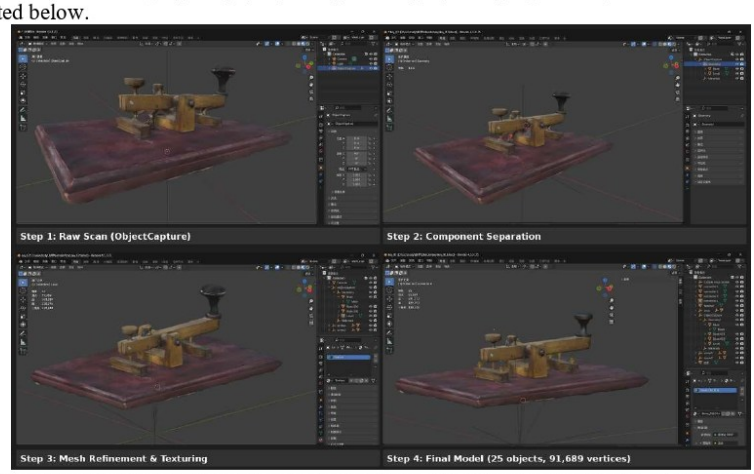
For the key, the iPhone was moved around the object in a systematic pattern while Reality Composer captured depth and texture data in real time.



The same setup was subsequently used to scan the receiver. Both raw scan outputs were successfully exported and imported into Blender 4.5 for further processing.

2. Key Model Refinement in Blender

The raw scan of the telegraph key (ObjectCapture output) was progressively refined in Blender, as illustrated below.



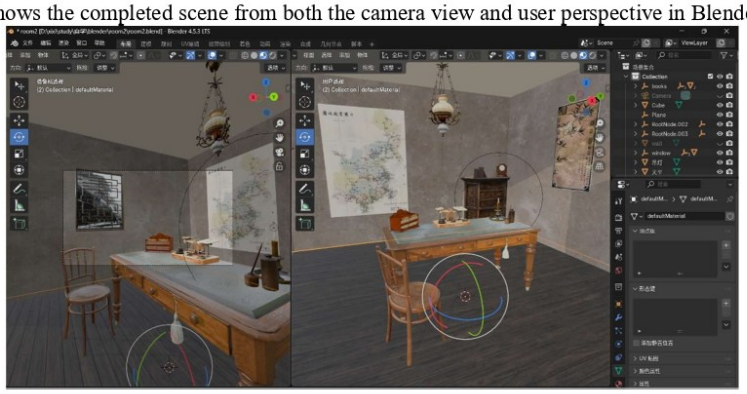
Several problems were found in the scanned model: the hollow areas inside the central lever arm and U-shaped bracket were filled in as solid geometry; small mechanical details like contact pins and cylindrical protrusions at both ends of the lever were too fine to be captured by the scanner; connection pins at the base needed to be moved to their correct position on the upper part; and the knob was badly distorted.

These issues were fixed one by one. The hollow sections were cut open to restore the correct shape. Missing details were rebuilt by hand. A replacement knob was downloaded from Sketchfab and reshaped to match the original. The final model contains 25 objects and 91,689 vertices.

3. Sending Room Scene Construction

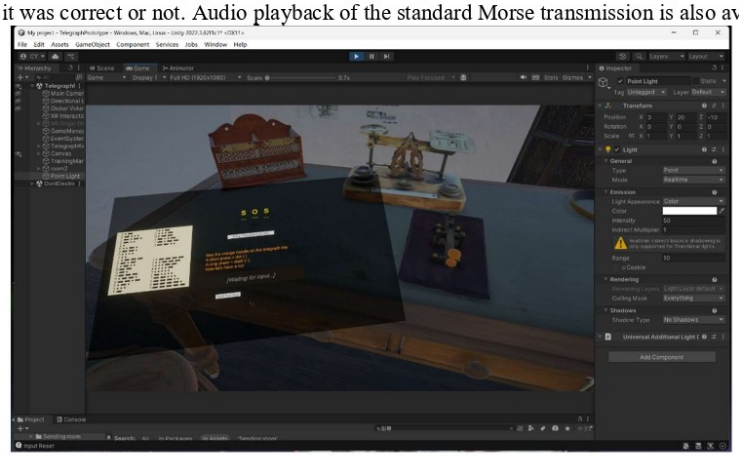
I built a historical Chinese telegraph office scene in Blender as the virtual environment for the Transmitter Station. The scene includes period-appropriate furniture such as wooden desk, chair,

and cabinet. To give the scene a Chinese historical context, several decorative elements were added: a Qing Dynasty telegraph line map on the wall, a bamboo painting, and a black-and-white historical street photo placed outside the window. Other props like hanging oil lamps, a balance scale, and writing materials were also included to make the scene feel more authentic. The figure below shows the completed scene from both the camera view and user perspective in Blender.



4. Unity Interactive System Development

The refined key model and sending room scene were imported into Unity 2022.3, where the interactive system for the Transmitter Station was built. Two modes have been developed: Training Mode gives the user a target word (e.g. SOS, HELLO, BYE) and shows its correct Morse code sequence. A Morse code reference table is displayed on screen as a guide. The user then can try reproduce the sequence by pressing the telegraph key. The system checks the input and shows whether it was correct or not. Audio playback of the standard Morse transmission is also available.



Operating Mode allows the user to freely press the key and generate dots and dashes, which are shown on screen in real time.

The interaction logic is built across four C# scripts:

1. SimpleMorseCodeSystem.cs handles the timing of dots and dashes based on press duration.
2. SimpleLeverPressController.cs controls the lever animation.
3. TrainingManager.cs manages the whole flow, which includes word selection, instruction, success, and fail states.
4. MorseSenderUI.cs handles all the UI updates, including colour-coded signal display and error messages.

尚需完成的任务 Work to do:

1. Refine the receiver model in Blender and implement its animations.
2. Further develop the Receiving Room scene in Blender.
3. Implement the interaction logic for Operating Mode in the Receiving Room, including the receiver's response to incoming Morse signals.
4. Integrate VR support for Meta headsets, conduct testing, and add additional features such as multi-language support.

存在问题 Problems:

1. The current UI is functional but does not fit well with the style of the scene. Improving the UI design to look more consistent and beautiful within the historical setting is still a challenge.
2. Inputting Morse code correctly is still quite difficult for beginners. It is not yet clear how to balance historical accuracy with a good user experience.
3. The receiver has a complex mechanical structure, making both the model refinement and animation significantly more difficult than the key.
4. When word spaces are removed to simplify user input, Morse code becomes ambiguous. How to handle this?

拟采取的办法 Solutions:

1. Refer to and learn from existing VR museum experience cases. The UI will also be gradually improved based on feedback from user testing in later stages.
2. Add on-screen text guidance showing the timing rules for dots and dashes (e.g. press duration thresholds), so beginners have a clearer reference while practising. Further solutions may be identified after user testing.
3. For complex parts of the receiver that were not captured clearly by the scanner, reference photos taken from multiple angles can be used to manually reconstruct those components in Blender, replacing the unclear parts.
4. Two possible approaches are being considered: introducing AI-based decoding to intelligently interpret ambiguous Morse input, or simplifying the input method by asking users to input letter by letter.

论文结构 Structure of the final report: (Chapter headings and section sub headings)

Chapter 1: Introduction
1.1 Project Background and Motivation
1.2 Project Objectives
1.3 Methodology and Development Workflow
1.4 Structure of This Report

Chapter 2: Background
2.1 The History of Telegraph Technology
2.2 The Origins and Standards of Morse Code
2.3 Technical Principles and Structure of the Telegraph Key and Receiver
2.4 Introduction to the Specific Artefacts on Display
2.5 The Significance of Digital Preservation of Cultural Heritage
2.6 Review of Existing VR Museum Experience Projects
2.7 Overview of 3D Scanning and Reconstruction Techniques
2.8 Tools and Technologies Used (LiDAR / Blender / Unity)

Chapter 3: Design and Implementation
3.1 Artefact Data Acquisition (Scanning and Photo Documentation)
3.2 Reconstruction and Refinement of the Telegraph Key Model
3.3 Reconstruction and Refinement of the Receiver Model
3.4 Sending Room Scene Design and Historical Reconstruction

3.5 Receiving Room Scene Design
3.6 Interaction Design
  3.6.1 Morse Code Input and Decoding Mechanism
  3.6.2 Implementation of Training Mode
  3.6.3 Implementation of Operating Mode
3.7 VR Adaptation and Meta Quest Deployment

Chapter 4: Results and Discussion
4.1 Development Results and Demonstration
4.2 User Testing Plan and Feedback
4.3 Discussion and Improvements Based on Feedback

Chapter 5: Conclusion and Further Work
5.1 Conclusion
5.2 Reflection
5.3 Further Work

References

Appendices (as required by template)

Supervision log

北京邮电大学 本科毕业设计（论文）教师指导记录表

Project Supervision Log

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	Yang	**名 First Name**	Canxi		
BUPT 学号 BUPT number	2022213370	**QM 学号 QM number**	221168420	**班级 Class**	2022215112
论文题目 Project Title	Development of a Digital Twin of a Morse Telegraph				

Please record supervision log using the format below:

Date: dd-mm-yyyy
Supervision type: face-to-face meeting/online meeting/email/messages/other (please specify)
Summary:

Date: 08-09-2025
Supervision type: face-to-face meeting
Summary: Discussed the project timeline and key deadlines.

Date: 16-09-2025
Supervision type: email
Summary: Contact the mentor and schedule an online meeting.

Date: 23-09-2025
Supervision type: email
Summary: Confirm the online meeting time.

Date: 24-09-2025
Supervision type: online meeting
Summary: Discussed the project scope and initial task breakdown.

Date: 26-09-2025
Supervision type: email
Summary: Confirm the version of the software being used.

Date: 01-10-2025
Supervision type: online meeting
Summary: Submitted questions regarding the telegraph artefacts and project design.

Date: 03-10-2025
Supervision type: email
Summary: Mentor responds to the questions I previously raised.

Date: 06-10-2025
Supervision type: online meeting
Summary: Discussed artefact selection and the scene modelling plan.

Date: 20-10-2025
Supervision type: online meeting

Summary: Clarify task details and discuss background research work.

Date: 28-10-2025
Supervision type: face-to-face meeting
Summary: Discuss the current progress and provide suggestions.

Date: 04-11-2025
Supervision type: online meeting
Summary: Discussed the program flowchart and details of the specification.

Date: 07-11-2025
Supervision type: email
Summary: Got feedback and approval of the specification.

Date: 10-12-2025
Supervision type: online meeting
Summary: Reviewed the program flowchart.

Date: 17-12-2025
Supervision type: online meeting
Summary: Discussed challenges in Morse input recognition and animation.

Date: 12-01-2025
Supervision type: email
Summary: Got feedback and approval of the early term report.

Date: 21-01-2026
Supervision type: face-to-face meeting
Summary: Discussed Morse decoding logic and mode design.

Date: 28-01-2026
Supervision type: face-to-face meeting
Summary: Discussed Training Mode design, tried the VR.

Date: 02-02-2026
Supervision type: face-to-face meeting
Summary: Scanned the morse telegraph key artifact.

Date: 10-02-2026
Supervision type: face-to-face meeting
Summary: Showed the refinement of key in Blender.

Date: 18-02-2026
Supervision type: face-to-face meeting
Summary: Scanned the morse telegraph receiver artifact.

Date: 18-02-2026
Supervision type: message
Summary: Confirm whether the refinement result of the key model in Blender is feasible.

Date: 26-02-2026
Supervision type: face-to-face meeting
Summary: Showed the interaction program of sending part in Unity.

Date: 02-03-2026

Supervision type: face-to-face meeting
Summary: Reviewed mid-term presentation slides.

Date: 03-03-2026
Supervision type: face-to-face meeting
Summary: Clarify the later schedule and UI design details

Date: 09-03-2026
Supervision type: Face-to-face meeting
Summary: Completed mid-term oral examination.

Date: 30-03-2026
Supervision type: Online meeting
Summary: Progress report.

Date: 14-04-2026
Supervision type: Face-to-face meeting
Summary: VR deployment related guidance, review of draft report.

Date: 20-04-2026
Supervision type: Face-to-face meeting
Summary: Report VR achievements and report progress.

Date: 21-04-2026
Supervision type: Face-to-face meeting
Summary: Review and guidance of report.

Risk and Environmental Impact Assessment

1. Risks to the successful completion of the project

Risk Event	L	C	R	Rating	Action
LiDAR scan quality insufficient for key components	3	3	9	Significant	Supplement scan data with manual modelling in Blender based on reference photographs.
VR framework compatibility issues preventing deployment	3	4	12	Significant	Switch to an alternative SDK (e.g. from XR Interaction Toolkit to Meta XR SDK), as was done during the project.
Scene polygon count exceeding VR device rendering capacity	4	3	12	Significant	Use the Decimate modifier in Blender to reduce polygon counts before deployment.
Loss of project files due to hardware failure	2	5	10	Significant	Maintain regular backups on cloud storage and external drives.
Supervisor unavailable for extended period	1	2	2	Low	Plan work independently and prepare questions in advance for meetings.

2. Risks of potential harm to people and/or animals

Risk Event	L	C	R	Rating	Action
User experiencing motion sickness or dizziness during VR use	3	2	6	Moderate	Maintain stable frame rates above 72 FPS; avoid sudden camera movements in the scene design.
Eye strain from prolonged VR headset use	3	1	3	Low	Recommend users take regular breaks during extended sessions.

3. Risks of potential harm to the environment

Risk Event	L	C	R	Rating	Action
Electronic waste from development hardware (PC, headset)	2	1	2	Low	Use existing personal devices where possible; return borrowed equipment after the project.
Energy consumption during development and testing	3	1	3	Low	No high-performance servers were used; development was carried out on an ordinary personal computer and a consumer-grade VR headset.

This project is primarily software-based and does not involve chemical substances, biological

materials, or large-scale energy consumption. The digital twin approach itself contributes positively to the environment by reducing the physical wear on real artefacts and eliminating the need for visitors to travel long distances.

4. Risks of potential financial loss

Risk Event	L	C	R	Rating	Action
Damage to borrowed equipment (iPhone, VR headset)	2	3	6	Moderate	Handle all borrowed devices with care and return them promptly after use.
Need to purchase paid software or assets	2	1	2	Low	Use free and open-source tools (Blender, Unity Personal Edition) and CC-licensed 3D assets from Sketchfab.