



FILIPE MIGUEL ALVAREZ MARQUES DE MIRANDA

Bachelor in Computer Science and Informatics Engineering

DEVELOPING SUPPORT FOR THE HERITAGE DIGITAL TWIN IN CONSERVATION - 3D

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon

October, 2023

DEVELOPING SUPPORT FOR THE HERITAGE DIGITAL TWIN IN CONSERVATION - 3D

FILIPPE MIGUEL ALVAREZ MARQUES DE MIRANDA

Bachelor in Computer Science and Informatics Engineering

Adviser: Armanda Rodrigues

Associate Professor, NOVA University Lisbon

Co-advisers: Nuno Correia

Full Professor, NOVA University Lisbon

Márcia Vilarigues

Assistant Professor with Habilitation, NOVA University Lisbon

Developing support for the Heritage Digital Twin in Conservation - 3D

Copyright © Filipe Miguel Alvarez Marques de Miranda, NOVA School of Science and Technology, NOVA University Lisbon.

The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

This document was created with the (pdf/Xe/Lua)LaTeX processor and the [NOVAthesis](#) template (v6.10.17) [22].

ACKNOWLEDGEMENTS

I want to express my heartfelt gratitude to the individuals and institutions who were essential in completing my master's dissertation.

Firstly, I am very grateful for the support provided by my Adviser, Armanda Rodrigues, and Co-Advisers, Nuno Correia and Márcia Vilarigues. Their guidance, support, and expertise were invaluable in shaping this dissertation and making it possible.

I extend my sincere appreciation to Fundação para a Ciência e Tecnologia for providing a research scholarship that allowed me to focus on my studies.

I also want to express my gratitude to Professor João Lourenço for freely providing a template for this document [22].

Lastly, I would like to thank my family and friends for their continued support, not only in this dissertation but throughout my academic and personal journey. I cannot imagine doing it all without you.

To everyone mentioned above and to all those who have supported me along this path, I am sincerely grateful for your contributions to the successful completion of my master's dissertation.

ABSTRACT

Heritage artefacts are often preserved and protected using physical conservation techniques under the direction of experienced professionals following thorough analyses of the artefacts.

Digital restoration techniques and, when appropriate, the use of 3D models to portray the artefacts as they were originally and currently are some of the recent innovations in digital technologies that have been developed to support researchers. Additionally, the information gathered, used, and produced in these contexts needs to be organized and curated in order to support future studies, reflect the development of the artefacts, and support the findings of researchers. The Digital Twin concept has been applied in various fields and can be utilized to address this demand. It is a virtual representation of an object with an automatic bidirectional flow of information between the physical and digital counterparts.

This dissertation aims to contribute to the definition and concretization of the Heritage Digital Twin based on 3D modelling and historical and contextual data. Our objective is to represent artefacts and spaces that have human visibility and cultural value and, as such, should be conserved.

It is highly beneficial for open-to-public establishments to have an integrated approach for managing both the conservation condition and the display of multiple artefacts. Therefore, a user-friendly interface that encourages engagement with the digital objects and allows the user to monitor the environment through sensors and their respective data is provided. These external data sources should be considered and used to make decisions regarding the physical object and its placement, aiming for its conservation and safekeeping.

Keywords: Digital Twin, Digital Twin for 3D Assets, Digital Twin in Conservation, Digital Heritage, Digital Heritage Conservation, 3D Interaction

RESUMO

Artefatos com um valor cultural elevado são frequentemente preservados e protegidos usando técnicas de conservação física sob a direção de profissionais experientes após análises minuciosas terem sido realizadas aos artefatos.

As técnicas de restauro digital e, quando adequado, a utilização de modelos 3D para representar os artefactos tal como eram originalmente e atualmente, são algumas das recentes inovações em tecnologias digitais que têm vindo a ser desenvolvidas para suportar os investigadores. Além disso, as informações obtidas, usadas e produzidas nesses contextos precisam de ser organizadas e selecionadas para apoiar futuros estudos, refletir o desenvolvimento dos artefatos e apoiar as descobertas dos especialistas. O Digital Twin é um conceito que tem sido aplicado em diversas áreas e pode ser utilizado para resolver esse problema. É uma representação virtual de um objeto com um fluxo bidirecional de informação entre o objeto físico e a representação digital.

Esta dissertação tem como objetivo contribuir para a definição e concretização do Digital Twin com base em modelação 3D e dados históricos e contextuais. O objetivo é representar artefatos e espaços com visibilidade e valor cultural que devem ser preservados.

É altamente benéfico para estabelecimentos abertos ao público ter a possibilidade de gerir, tanto a exposição de múltiplos artefatos, como a sua condição de conservação. Assim, é fornecida uma plataforma que promove a interação com os objetos digitais e permite ao utilizador monitorizar o ambiente através de sensores e os respetivos dados. Essas fontes de dados externas devem ser consideradas e utilizadas para tomar decisões sobre o objeto físico e a sua colocação, visando a sua conservação e proteção.

Palavras-chave: Digital Twin, Digital Twin para Artefactos 3D, Digital Twin em Conservação, Herança Digital, Conservação de Herança Digital, Interação 3D

CONTENTS

List of Figures	ix
Acronyms	xii
1 Introduction	1
1.1 Motivation	1
1.2 Context	2
1.3 Problem Description and Objectives	2
1.4 Main Contributions	3
1.5 Document Structure	3
2 Background	5
2.1 Digital Twin (DT)	5
2.1.1 Advantages	7
2.1.2 Historic Digital Twin	8
2.2 Building Information Modelling	8
2.3 Adjacent fields of study	9
2.3.1 Internet of Things	9
2.3.2 Artificial intelligence (AI)	10
2.3.3 Augmented Reality (AR) and Virtual Reality (VR)	12
2.4 Heritage Conservation	13
2.4.1 Digital Heritage	14
2.4.2 Digital Heritage Conservation	14
2.5 National Palace of Sintra	15
2.6 Discussion	17
3 Related Work	19
3.1 Various uses of Digital Twin	19
3.1.1 Digital Twin applications in different areas	19
3.1.2 Digital Twin for Heritage Interpretation in a Mobile App	21

3.2	Preservation of Cultural Heritage using the Digital Twin	22
3.2.1	Combining Integrated Informative System and Historical Digital Twin	22
3.2.2	Using the Digital Twin in Preventive Conservation	25
3.2.3	Ontology supporting Digital Twin	26
3.3	Digital Twin Challenges	29
3.4	Discussion	30
4	Defining the Solution	32
4.1	Requirements	32
4.1.1	Functional Requirements	32
4.1.2	Non-functional Requirements	33
4.2	Technologies	34
4.3	System Design	36
4.4	Conclusions	38
5	Implementing the Solution	40
5.1	3D Model Importer	40
5.2	Application Interactions	44
5.2.1	Imported Models	45
5.2.2	Sensors	52
5.2.3	Camera	54
5.3	Data Management	55
5.3.1	Save	56
5.3.2	Load	64
5.4	Sensor Data	67
5.4.1	Importing Data	68
5.4.2	Visualizing Data	71
5.5	Limitations	74
6	Evaluation	77
6.1	Usability Evaluation	78
6.2	Specialist Evaluation	79
6.3	Evaluation Results	79
6.3.1	User Analysis	79
6.3.2	Usability Results	80
6.3.3	Specialization Results	87
7	Conclusions and Future Work	90
7.1	Conclusions	90
7.2	Future Work	91

Bibliography	93
Appendices	
A Consent and Informative Documents	97
B Usability Test	101
C Specialist Test	111
D Evaluation Results	116

LIST OF FIGURES

2.1	Illustration of Component, Asset, System and Process Twins [13].	6
2.2	Digital Twin misconceptions [8].	7
2.3	IoT device growth [8]	10
2.4	Example of an AR app.	12
2.5	National Palace of Sintra [31].	15
3.1	Percentage share of research areas using Digital Twin [8]	20
3.2	Roadmap of the AR creation and testing from the Digital Twin of Niemeyer's Pampulha Buildings [6].	22
3.3	General framework of the research workflow [24].	23
3.4	The Mater Matuta and the Resting Satyr [24].	24
3.5	From the very dense scanned and tessellated model to the FE (Finite Element) mesh model, passing through the quad mesh and the surface model (from left to right) [24].	25
3.6	The complete value-based risk management plan process. The phases highlighted in red are the key steps toward implementing the method. [20] . . .	27
3.7	The semantic graph of the HDT ontology. HC: HeritageClass. HP: Heritage Property [27].	28
3.8	Schema of direct action activated by an automatic alert system based on HDT data [27].	29
3.9	Part of a semantic description using the HDT ontology [27].	30
4.1	Full system architecture	36
4.2	Digital Twin environment functionality	37
4.3	Unity platform's features and user interactions	38
5.1	Initial view of the application.	44
5.2	View of a selected artefact.	46
5.3	Wireframe visualization of two artefact representations.	52
5.4	Default sensor data graph visualization.	53

5.5	Sensor data file provided by Department of Conservation and Restoration (DCR).	68
5.6	(a) Temperature and (b) Humidity graph using sensor data.	72
5.7	Temperature and humidity merged graph using sensor data.	72
5.8	Wooden crockery with no texture on the honey scoop.	75
6.1	Gender distribution.	80
6.2	Age distribution.	81
6.3	Level of education distribution.	81
6.4	Time taken by each participant to complete the tasks, ordered by their ages.	82
6.5	System Usability Scale (SUS) results.	83
6.6	One of the feature-oriented questions.	84
6.7	Result distribution of the usability questions referring to the ease of use of specific functionalities of the application.	85
6.8	One of the feature-oriented questions.	85
6.9	Result distribution of the usability questions referring to the understandability and realness of specific aspects of the application.	87
6.10	Result distribution of the specialist questions referring to the relevance of some of the application's features.	89
D.1	User Analysis.	117
D.2	Time taken to complete the list of tasks, ordered by age.	118
D.3	Usability section 1 (SUS) results, Part 1.	119
D.4	Usability section 1 (SUS) results, Part 2.	120
D.5	Usability section 1 (SUS) results, Part 3.	121
D.6	Usability section 1 (SUS) results, Part 4.	122
D.7	Usability section 2 results, Part 1.	123
D.8	Usability section 2 results, Part 2.	124
D.9	Usability section 2 results, Part 3.	125
D.10	Usability section 3 results, Part 1.	126
D.11	Usability section 3 results, Part 2.	127
D.12	Usability section 3 results, Part 3.	128
D.13	Specialist's results, Part 1.	129
D.14	Specialist's results, Part 2.	130
D.15	Specialist's results, Part 3.	131
D.16	Specialist's results, Part 4.	132
D.17	Specialist's results, Part 5.	133

LIST OF LISTINGS

5.1	"LoadModel" method from "LoadModels" file.	42
5.2	"ObjectResize" method from "LoadModels" file.	43
5.3	"OnMouseDown" method from "InteractionController" file.	47
5.4	"OnMouseDrag" method from "InteractionController" file.	47
5.5	"GetMouseWorldPosition" method from "InteractionController" file. . .	48
5.6	Rotation logic from "Interaction Controller" file.	50
5.7	Rotation logic from "Interaction Controller" file.	51
5.8	"ChangeVisualization" method from "Sensor Controller" file.	53
5.9	Free camera movement logic from "Camera Controller" file.	55
5.10	"Save" method from "SystemManager" file.	57
5.11	"SaveModels" method from "SaveSystem" file.	60
5.12	"ModelData" class.	61
5.13	"SensorData" class.	63
5.14	"Load" method from "SystemManager" file.	65
5.15	"LoadModel" method from "SaveSystem" file.	66
5.16	"ParseCSV" method from "CSVParser" file.	69
5.17	Old "Clear" method from "CSVParser" file.	71
5.18	New "Clear" method from "CSVParser" file.	71
5.19	"UpdateGrpah" method from "GraphMaker" file used to update data values on the active graph.	73

ACRONYMS

AI	Artificial Intelligence (<i>pp. 6, 9, 19</i>)
AR	Augmented Reality (<i>pp. 9, 19, 21, 31</i>)
BIM	Building Information Modelling (<i>p. 4</i>)
DCR	Department of Conservation and Restoration (<i>pp. x, 35, 68, 69</i>)
DI	Department of Computer Science (<i>p. 35</i>)
DSM	Dense Surface Model (<i>p. 21</i>)
DT	Digital Twin (<i>pp. 4, 5, 8, 9, 19, 21, 22, 26, 30, 31</i>)
FCT	NOVA School of Science and Technology (<i>pp. 2, 35</i>)
HBIM	Historic Building Information Modelling (<i>pp. 4, 21, 26, 31</i>)
HDT	Historic Digital Twin (<i>pp. 4, 8, 22, 23, 25, 26, 31</i>)
IIS	Integrated Information System (<i>pp. 23–25, 31</i>)
IoT	Internet of Things (<i>pp. 3, 5, 8, 9, 19</i>)
MSAA	Multisample anti-aliasing (<i>p. 45</i>)
SUS	System Usability Scale (<i>pp. x, 77, 79, 80, 82–84, 119–122</i>)
TSL	Terrestrial Laser Scanning (<i>p. 21</i>)
UAV	Unmanned Aerial Vehicle (<i>p. 21</i>)
VR	Virtual Reality (<i>p. 19</i>)

INTRODUCTION

This chapter serves as an introduction to the problem that this dissertation aims to solve. It starts by presenting the motivation for the development of the work, supported by the need and opportunity to address it at this time. The problem is then formulated, and objectives and expected contributions are laid out. Finally, the document's organization is provided at the end of the chapter to help the reader navigate the report.

1.1 Motivation

The main conservation concerns of World Heritage Sites today include inadequate environmental conditions, climate change, the massification of tourism, and poor management and resources [29]. Usually, the need to prevent and respond to danger becomes even more significant with the rise of an asset's heritage value, making it even more important to protect. High conflict zones are an example of areas that may lose assets of unimaginable value ¹.

Technology has advanced to the point where it can create highly detailed models of objects that hold significance. This advancement primarily relies on 3D modelling techniques coupled with data models that allow for complex simulations. Additionally, the Internet of Things plays a crucial role in collecting extensive datasets, which, in turn, support more comprehensive monitoring capabilities [9].

This technological progress has led to increased adoption of Digital Twin (DT) technology, particularly in sectors such as industry, construction, and Cyber-Physical Systems (CPS)², where it has proven valuable in improving operational efficiency and decision-making processes [12].

In the field of conservation, there is a recognized need to establish structured systems for preserving the lifecycle of historically and culturally significant artefacts. Simultaneously, it is essential to monitor the environmental conditions in which these objects are stored or

¹<https://reliefweb.int/report/syrian-arab-republic/five-years-conflict-state-cultural-heritage>

²<https://www.geeksforgeeks.org/introduction-to-cyber-physical-system/>

displayed. Digital Twin technology has emerged as a practical solution to address these challenges and effectively manage and safeguard valuable artefacts.

1.2 Context

This dissertation work is developed in collaboration and with the support of Department of Conservation and Restoration (DCR), the VICARTE³, the Department of Computer Science (DI) and the Nova Laboratory for Computer Science and Informatics (NOVA LINCS)⁴ of NOVA School of Science and Technology (FCT). VICARTE is a research unit in a partnership between the NOVA School of Science and Technology and the Faculty of Fine Arts of the Universidade de Lisboa that indulges in research aimed at developing new materials for glass and ceramics and studying traditional and historical production practices.

The goal of this dissertation is to create support for an environment involving Digital Twins of cultural assets to enable resource-efficient monitoring and conservation processes. The assets used will consist of a room and some objects with cultural value from Parques Sintra⁵. The environment in which this work will take place and be applied is Palácio Nacional de Sintra⁶, a palace located in Sintra known for its two big white chimneys.

1.3 Problem Description and Objectives

Heritage artefacts are often preserved and protected using interventive conservation techniques under the direction of experienced professionals, following thorough analyses of the objects. Digital restoration techniques and, when appropriate, the use of 3D models to portray the artefacts are examples of recent innovations in digital tools that are focused on aiding these procedures.

Additionally, the information gathered, employed, and produced in these contexts needs to be organized and curated in order to assist future studies, reflect the development of the artefacts, and support the results of the researchers. The Digital Twin concept, which has been used in a variety of contexts as a virtual and comprehensible representation of a constantly updated object, can be used to address this demand.

Having an integrated manner of managing not only the conservation condition but also the display of several artefacts is very valuable for open-to-public establishments.

This dissertation aims to develop an initial study and support for Digital Twin of 3D artefacts applied to heritage conservation and management. This includes:

³<https://vicarte.org/>

⁴<https://nova-lincs.di.fct.unl.pt/>

⁵<https://www.parquesdesintra.pt/en/>

⁶<https://www.parquesdesintra.pt/en/parques-monumentos/palacio-nacional-sintra/>

- The study of fundamental concepts regarding Digital Twins, previous works using Digital Twins in Conservation and other relevant topics such as [Internet of Things \(IoT\)](#);
- Establish and outline the foundation for the integration and effective incorporation of the Digital Twin technology into a contextual and historical workflow;
- Contribute to the creation of an interactive platform to manage and explore 3D models of heritage artefacts in addition to monitoring the artefacts using environmental data.

1.4 Main Contributions

The research conducted for this project has practical implications for the preservation of culturally significant artefacts in heritage-rich settings, particularly within the context of the Palácio Nacional de Sintra. It is important to note that, for this dissertation, preliminary data with characteristics comparable to those found at the Palace was employed.

The main contribution of this work is the platform that serves as support for Digital Twins of artefacts present in culturally rich environments. It can be divided into the following main points:

- The platform offers detailed tri-dimensional representations of culturally valued artefacts and the environments they are placed in. The application offers the ability to visualize and interact with the artefacts. In addition, it integrates environmental sensorial data to allow the monitorization of the artefacts and provides support for the decision-making needed to position and expose environmentally sensitive artefacts.
- The platform was designed and evaluated with the support and close collaboration of the community, who will later use it. Their expertise and input were instrumental in shaping the system and its features, which inspires confidence in the relevance of the system, its features, and its applicability in a culturally rich environment, like the one at the National Palace of Sintra.

1.5 Document Structure

- **Chapter 1 - Introduction:** The first chapter provides an overview of this project's work. It references the context of this dissertation along with the motivation behind it. It also explains briefly the problem that exists and what the objectives are. Finally, it presents the structure of the document.
- **Chapter 2 - Background:** The second chapter explains all the concepts intricately related to this work, that are necessary to thoroughly understand the whole process.

Firstly, the idea of [Digital Twin \(DT\)](#) and [Historic Digital Twin \(HDT\)](#) are explained along with fields of study that are directly connected and are often applied together. Secondly, the [Building Information Modelling \(BIM\)](#) and [Historic Building Information Modelling \(HBIM\)](#) information models are explained, as these are important in understanding the DT system. Lastly, some ideas on conservation are presented to allow a better understanding of the environment the DT is trying to be applied to.

- **Chapter 3 - Related Work:** The third chapter is responsible for the analysis and breakdown of relevant literature and work done by others. This serves not only as additional and essential technical and circumstantial knowledge but also as a reference for future work.
- **Chapter 4 - Defining the Solution:** The fourth chapter showcases the process taken to define the solution to this problem. It delineates the requirements, establishes the project's environment and elaborates on the system's architecture. This a very important chapter that paves the way to the implementation of the solution.
- **Chapter 5 - Implementing the Solution:** The subject of the fifth chapter is the solution implementation. Here, the whole implementation process is described. It showcases the importation of 3D models, all forms of interaction with them, data management and finally, data visualization.
- **Chapter 6 - Evaluation:** The sixth chapter discusses the system evaluation that was carried out at the conclusion of the project. It includes information about the format utilized to conduct user testing, the participants and the evaluation results.
- **Chapter 7 - Conclusions and Future Work:** The seventh and last chapter summarizes all the developed work, emphasizing the main conclusions of the dissertation and discussing potential future developments for the system that have been considered as future work.

BACKGROUND

The topics covered in this chapter are crucial to the development and support of Digital Twin technology as they form the foundational knowledge and key building blocks necessary to harness the full potential of this innovative concept.

It begins with a comprehensive explanation of Digital Twin and then explores misconceptions and advantages associated with these systems. The concept of Historic Digital Twin is also discussed.

Building Information Modelling (BIM) and Historic Building Information Modelling are explored as information models associated with Digital Twins. Additionally, the chapter explains fundamental concepts related to important areas such as the Internet of Things (IoT), Artificial Intelligence (AI), and Augmented and Virtual Reality (AR/VR).

The concepts of Digital Heritage and Heritage Conservation are also introduced, highlighting the significance of Digital Twin applications in preserving cultural heritage and historical artefacts.

Lastly, an introduction to the history and some context on the National Palace of Sintra is provided.

2.1 Digital Twin (DT)

There are definitions of DT dating back to the early 2000s when this was still an idea, and there were slight changes as time passed. The first recorded definition happened in 2003 in a presentation hosted by Dr. Michael Grieves and later documented in a white paper [11].

NASA published a paper in 2012 that served as a crucial step in the definition of the DT: "A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin." [10].

Nowadays, a DT is conceived as a virtual model that reflects, as accurately as possible, a physical object [13]. In addition, it should possess real-time data, usually delivered by IoT sensors placed on the physical object that automatically link the visual and physical objects. Because of this, the IoT implementation needs to be structured and built according

to the system's needs. This data flow is usually bidirectional and can happen at varied frequencies, depending on the system and its needs. This allows simulations and studies on the digital counterpart to study and decide the best measures to apply to the physical one without reducing resources committed in the process.

Artificial Intelligence (AI) and Machine Learning are fields of study that are often essential for treating and processing Digital Twin's data.

There are four types of Digital Twins, and it is very common to use them together within a system to cover all needs. The major difference between these is the scope of the system they represent:

- **Component or Part Twins:** Represent the most basic components of a system. Example: a gear in a mechanical system.
- **Asset Twins:** An asset is an interaction between components, and an Asset Twin represents precisely this. Example: an engine.
- **System or Unit Twins:** These represent groups of Assets assembled into a complete unit, enabling the overview of a functioning system. Example: all the units necessary for the production of a component of the finished car.
- **Process Twins:** Represent an entire working facility, showcasing how different systems work together. Example: entire production facility.

These are showcased in Figure 2.1:

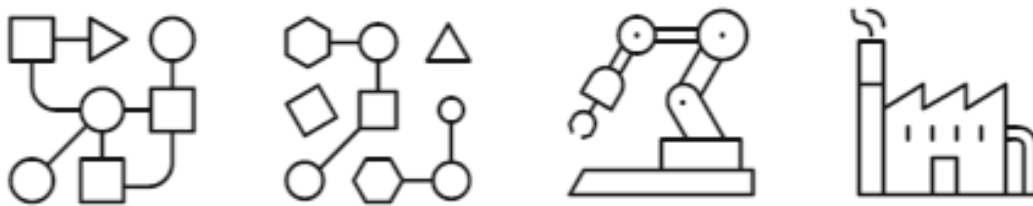


Figure 2.1: Illustration of Component, Asset, System and Process Twins [13].

Some concepts are often mistaken for a Digital Twin [8]. These have slight differences, usually in how data flows between the physical and digital objects. The most common ones are *Digital Model* and *Digital Shadow*:

- **Digital Model:** It is defined as a digital replica of a pre-existing physical object. The defining feature of a Digital Model is that there is no automatic flow of information between the physical and digital objects, which happens in a Digital Twin. In this way, if a change occurs in the physical object, there will be no effect on the Digital Model.

- **Digital Shadow:** It differs from the Digital Twin by having a unidirectional flow of information from the physical object to the digital one. A change in the physical object affects the digital one but not the other way around.

These are showcased in Figure 2.2:

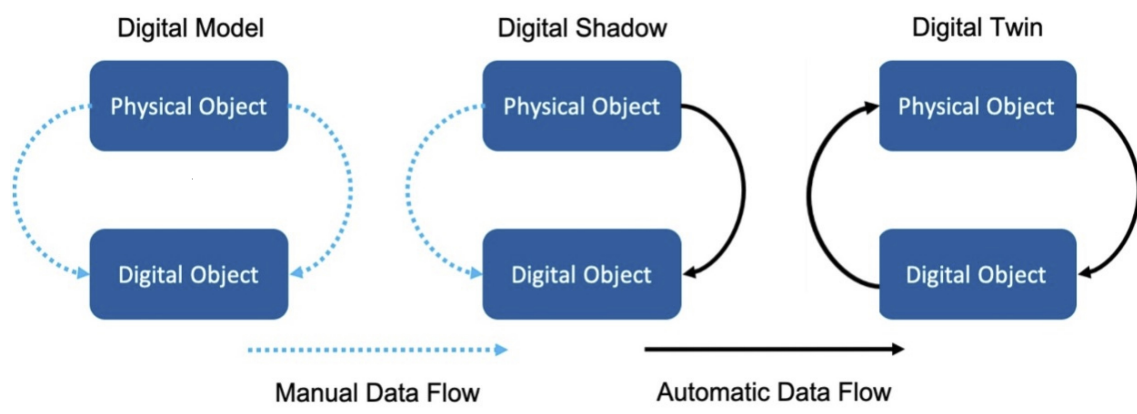


Figure 2.2: Digital Twin misconceptions [8].

2.1.1 Advantages

As mentioned in Section 2.1, a Digital Twin is a relatively recent concept, representing a digital replica of a physical object with automatic data flow between the two counterparts. This can be applied in several technological areas and brings many advantages, the major ones being [13] [7]:

- **Better R&D:** With the resources to research and test with digital prototypes instead of physical ones, which are expensive and hard to modify, teams can improve their development phases and make all the necessary tests and changes before production in a more efficient and flexible way.
- **Improved Efficiency:** Even after a product is in production, a Digital Twin can help recreate the production environment and, as such, achieve greater efficiency.
- **Predictive and Smaller Maintenances:** Utilizing digital twins to forecast maintenance or repair needs enables producers to schedule maintenance in advance, minimizing downtime and extending asset lifespan.
- **Reduced Risk:** Digital twins can assist businesses in identifying and mitigating potential risks, such as safety dangers or environmental repercussions, by simulating the performance of an asset under multiple circumstances.
- **Improved End-Of-Life Processing:** By better understanding a product's lifetime and the materials that could be harvested from it at the end, producers can better

decide how to process these objects. Usually, this includes, but is not limited to, recycling or refurbishment.

2.1.2 Historic Digital Twin

An **HDT** has a **DT** as its basis, but it considers certain peculiar characteristics of historical and cultural assets. It mainly consists of historical, geometric, mechanical, and structural characteristics [24]. These are easier to define and obtain, thanks to technical advances in the architectural field, like metric surveys [30].

These characteristics and the combined usage with other data systems, **IoT** based technologies, and Machine and Deep Learning applications, create a more significant opportunity to understand the historical context and account for risks, potentially increasing the efficiency of conservation and maintenance strategies of historical and cultural heritage.

2.2 Building Information Modelling

Building Information Modeling (BIM), from a technological standpoint, is a project simulation made up of 3D models of the project's components linked to all the necessary data regarding the project's planning, design, construction, or operation [1]. The two more important features that discern it from regular 3D CAD¹ are:

- BIM contains structured and multi-disciplinary data, forming an adjustable and more complex information model.
- Regular 3D CAD defines a building using separate 3D views, making it difficult to alter one view without also checking and updating the others, which naturally increases the possibility of committing errors.

The US National Building Information Model Standard Project Committee² proposes the following definition:

"Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life cycle; defined as existing from earliest conception to demolition."

Historic Building Information Modelling:

The term HBIM was first used by Professor Murphy in 2009 [25]. HBIM strives to preserve historical heritage while using BIM technology to manage and improve building architecture.

¹<https://www.designlaunchers.com/what-is-3d-cad-modeling>

²<https://www.nationalbimstandard.org/>

The major challenge while creating an HBIM is modelling since it is required to create straightforward procedures for BIM models that simultaneously provide correctness, precision, and quality of representation. The model should also be enhanced by non-purely geometric data and information, such as historical data, analyses of degradation or deformations, etc.

A survey of the asset is the first step in HBIM modelling. Typically, a digital survey is conducted using a laser scanner³ or a camera for land photogrammetry⁴. Consequently obtained are the point clouds⁵, which are collections of points with known spatial coordinates. Given that each point is connected to a certain colour received from a photograph taken by the digital tool during scanning, these points include both geometric and colourimetric information⁶.

2.3 Adjacent fields of study

Some technological areas are tightly related to DT development and are usually used together to enrich the system and provide and manage the Digital Twin data. The most prominent and relevant concepts are IoT, AI, and Augmented Reality (AR).

2.3.1 Internet of Things

Internet of Things refers to the interconnection of physical devices, vehicles, buildings, and other items embedded with sensors, software, and network connectivity, enabling them to collect and share data [28].

The rise of IoT technology is evident in the number of devices registered annually; the increase in IoT devices since 2016 is depicted in Figure 2.3. The number exceeded 17 billion in 2018, and according to estimates, there will be over 75 billion gadgets in use by 2025, with the sector having a value of over 5 trillion dollars. It demonstrates these gadgets' significant influence and advances the idea of a completely wired world [8].

One of the main benefits of IoT is the ability to collect large amounts of data from connected devices and systems. This and the use of systems that can analyze large quantities of data can improve efficiency, reduce costs, and make better decisions. For example, in the manufacturing industry, IoT-enabled devices can be used to monitor and optimize production processes, leading to increased productivity and reduced downtime. Similarly, in the healthcare industry, IoT-enabled devices can be used to monitor patients' health and provide real-time data to doctors and nurses, leading to improved patient outcomes [8].

Another key benefit of IoT is the ability to control and monitor devices remotely. This can be used to improve safety and security and provide new services and experiences.

³<https://www.faro.com/en/Resource-Library/Article/understanding-laser-scanners>

⁴<https://www.photomodeler.com/uav-photogrammetry-for-surveying/>

⁵<https://www.dronegenuity.com/point-clouds/>

⁶<https://www.sciencedirect.com/topics/chemical-engineering/colorimetry>

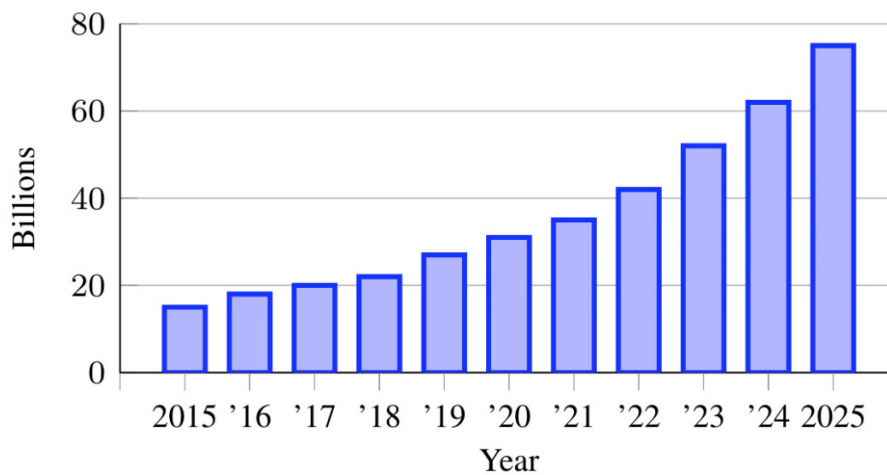


Figure 2.3: IoT device growth [8]

This is exceptionally beneficial for the transportation industry [26], smart-home logistics [4], and resource management [44]. IoT-enabled vehicles can be used to monitor traffic conditions and optimize routes, leading to reduced congestion and improved safety. Similarly, IoT-enabled devices can control lighting and heating, leading to reduced energy consumption and improved comfort. Finally, IoT-enabled devices can monitor and manage resources such as water and energy, leading to more efficient use and reduced waste.

However, despite the many benefits of IoT, some challenges need to be addressed. One of the main challenges is security. As more and more devices are connected to the Internet, the risk of cyber-attacks increases. This is particularly concerning in critical industries such as healthcare and transportation, where the consequences of a security breach could be severe.

Another challenge is interoperability and scalability. As more and more devices become connected to the Internet, there is a risk that they will not be able to communicate with one another. To address this challenge, developing IoT standards and encouraging device manufacturers to adopt them is essential. This will ensure that all devices can communicate with one another and that data can be easily shared and analyzed.

In conclusion, IoT technology has the potential to revolutionize many industries and improve the way we live and work. In addition, integrating digital twin technology with IoT can significantly enhance the capabilities and benefits of IoT. The use of digital twin technology can lead to improved efficiency, and digital replicas can be used to simulate and optimize the performance of the physical asset or system.

2.3.2 Artificial intelligence (AI)

Artificial intelligence refers to the interdisciplinary field that aims to develop algorithms and systems to perform tasks and solve problems that would typically require human intelligence. AI systems can be trained to process and analyze data, recognize patterns, make decisions and predictions, and even learn and adapt over time [14].

AI is a rapidly growing technology and has gained a significant increase in efficiency and correctness due to the increased availability of large data sources.

Rule-based systems and machine-learning systems are the two primary divisions of AI systems. While machine learning systems employ algorithms and statistical models to learn from data and make predictions, rule-based systems use a set of predetermined rules to do a task.

Machine learning is a branch of AI that trains a model to recognize patterns and make predictions on new data [16]. It can be divided into supervised, unsupervised, and semi-supervised learning. Supervised learning algorithms are trained on labelled data, unsupervised learning algorithms are trained on unlabeled data, and semi-supervised learning uses small labelled data in training as a guide.

Artificial neural networks that try to mimic the human brain through a combination of data inputs, weights, and biases are used in deep learning [15] to model complex data abstractions. This makes it possible for deep learning models to perform better as they are exposed to more data. It is possible to employ AI-based decision support systems to evaluate vast volumes of data and generate predictions or suggestions based on that data. This can be used in various industries, such as finance, healthcare, and customer service, to make decisions and improve performance. Another important application of AI is in natural language processing (NLP) [17]. NLP is a branch of AI that involves the development of algorithms that can understand and process text and words. This has the potential to improve human-computer interaction and make it possible for computers to understand and respond to natural language input.

There are many challenges regarding AI implementation. One of the main ones is the lack of explainability of AI systems. As AI systems become more complex and sophisticated, it becomes increasingly difficult to understand how they arrive at their decisions. For example, this can be a problem in healthcare industries, where AI system decisions can have serious consequences.

Another challenge is the issue of bias in AI systems. AI systems can learn from the data they are trained on; if the data is biased, the AI system can also become biased. This can lead to unfair or discriminatory decisions, which can have severe consequences in certain industries [32].

Integrating AI with digital twin technology can lead to improved decision-making, increased efficiency, and reduced costs, making it a powerful combination for various industries to improve their processes and operations.

Zhihan et al. [23] present an example combining AI and DT for the maintenance and quality assurance of aerocrafts. In this example, a Digital Twin of the aircraft is created, and AI algorithms are used to evaluate the digital aircraft's status and consequently assess the status of the real aircraft.

2.3.3 Augmented Reality (AR) and Virtual Reality (VR)

Augmented Reality integrates the user's environment with some extra digital data overlaid in real-time, enhancing the users view of the real world [36]. AR is mostly used to provide users with additional contextual information or enhance part of their environment.

AR achieves this by providing sound, visual and other sensory information through an external device, often a smartphone or glasses. The device adds digital information, changing part of the users' perception of the real world. Figure 2.4 shows how an AR app was used to change part of the users' perception of reality.

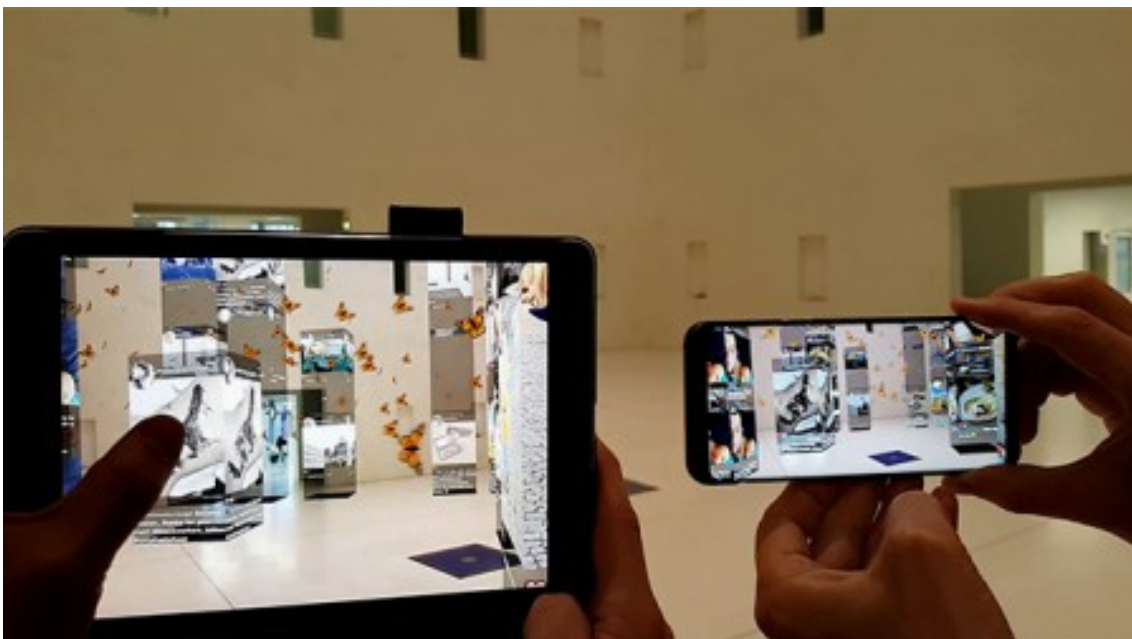


Figure 2.4: Example of an AR app.

AR is occasionally confused with Virtual Reality (VR), but these are different. VR is a completely computer-generated virtual environment; there is no real-world component. The device that allows the user to experience these environments is also different; VR uses a full and specialized headset, while AR can use just a smartphone to superimpose artificial artefacts on the physical world, as mentioned before.

AR is being used in many prominent and fast-growing areas like entertainment, gaming, retail, navigation, and measurements. One specific way AR can be used is to help visualize architectural structures, like buildings, as can be seen by the AR app that creates AR layers on buildings to enrich the viewing experience, introduced by Lee et al. [21]. This ties strongly to the Digital Twin, since we can use the digital replica as the basis of the digital information to be transmitted to the user and even build on it through external sources, as seen in Section 3.1.2.

2.4 Heritage Conservation

The phrases "preventive conservation", "remedial conservation", and "restoration", generally referred to as "conservation", have been introduced by ICOM-CC in the field of cultural heritage preservation. Using these words, preservation activities are grouped according to the particular goals and techniques they employ. [19]

The definitions of each of these terms are provided below:

- **Conservation:** All initiatives and plans under this general heading are meant to preserve tangible cultural heritage while also making it accessible to both current and future generations. Preventive conservation, remedial conservation, and restoration are the three main subcategories of conservation. The historical and physical qualities of the cultural heritage item must be taken into account in all conservation efforts.
- **Preventive Conservation:** The variety of steps and efforts intended to avert and reduce the possibility of further degradation or loss of cultural heritage artefacts are together referred to as preventive conservation. Notably, these activities preserve the objects' original appearance because they do not entail changes to the objects' materials or structures.

Security measures, environmental control (such as limiting light, humidity, pollution, and pest infestations), and emergency preparedness are a few examples of preventive conservation methods.

- **Remedial Conservation:** Actions taken directly on a single item or a collection of items with the aim of minimizing ongoing deterioration processes or reinforcing their structural integrity are referred to as remedial conservation. These actions are only taken when an item is highly fragile or is degrading quickly, endangering its preservation over a short period of time. It is significant to remember that remedial conservation can occasionally result in changes to the artefacts' appearance.

Activities like removing bugs from textiles and desalting pottery are a few examples of remedial conservation efforts.

- **Restoration:** The term "restoration" describes acts that are particularly carried out on a unique and stable cultural heritage item to increase its understanding, appreciation, and usefulness. These procedures are often carried out when a product has lost some of its value or functionality as a result of prior modifications or degradation. Although they frequently lead to obvious changes in the item's appearance, restoration operations prioritize the preservation of the original material.

Restoring a painting's visual appeal through retouching and putting a damaged sculpture back together are examples of restoration work.

In summary, these terms play a crucial role in guiding and categorizing the diverse preservation efforts aimed at safeguarding our tangible cultural heritage. This allows for a

clear distinction between the various objectives and methods employed in the conservation field.

UNESCO ⁷ defines conservation of cultural heritage as [40]: "The conservation of cultural heritage refers to the measures taken to extend the life of cultural heritage while strengthening the transmission of its significant heritage messages and values. In the domain of cultural property, the aim of conservation is to maintain the physical and cultural characteristics of the object to ensure that its value is not diminished and that it will outlive our limited time span."

2.4.1 Digital Heritage

Computer-based assets of significant worth that must be preserved for future generations make up digital heritage [38]. Digital legacy comes from various communities, businesses, sectors, and geographical areas. Not all digital resources have everlasting value, but those that do demand proactive preservation techniques to preserve the transmission of digital legacy.

New options to preserve cultural content and increase the accessibility of cultural heritage to all audiences are made possible by digital technologies. According to the UNESCO's Charter for the Preservation of Digital Heritage [37], resources for human knowledge, such as those in the sectors of culture, education, science, and administration, in addition to those involving technical, legal, medical, and other types of data, are more often digital or transformed from analogue resources. Furthermore, by using computers, which are increasingly more accessible to the world population, humans are creating digital resources they deem worthy of preserving and sharing with others. All this suggests an emerging digital heritage instead of the traditional physical heritage.

2.4.2 Digital Heritage Conservation

Digital Conservation consists of processes created to ensure the continued accessibility of Digital Heritage [39]. This requires figuring out how to show data in a different way than how it was initially displayed to people through a combination of hardware and software tools.

Probably, the most concerning threat to Digital Preservation is losing the means of accessibility to the assets since digital materials cannot be preserved if their access is lost. These are some strategies built to prevent this, but not limited to:

- Selecting the materials to be preserved since not everything can be.
- Apply standards to increase the available means of access and reduce unknown problems.
- Place the digital materials in a safe place.

⁷<https://www.unesco.org/en>

- Control the material using structured metadata.
- Protect the identity and integrity of data.

2.5 National Palace of Sintra

This section will serve as an introduction to some of the National Palace of Sintra's⁸ characteristics and history [31].

This Palace is the only one that has lived through almost Portugal's entire history and is situated in the centre of Sintra. Its two conical chimneys atop the royal kitchen stand out on the picturesque Sintra landscape; these are what the Palace is most known for and can be seen in Figure 2.5.



Figure 2.5: National Palace of Sintra [31].

The present-day Palace of Sintra emerges as the culmination of numerous royal residences. It constitutes a complex of structures that were erected, expanded, and modified over several centuries. Pinpointing the precise date of the inception of the country's most ancient Palace remains an enigma awaiting resolution. It is quite probable that the initial edifice was raised around the 10th or 11th century, a time when Sintra was governed by Moorish authority.

⁸<https://www.parquesdesintra.pt/en/parks-monuments/national-palace-of-sintra/>

Throughout different periods, nearly every monarch of Portugal, together with their consorts, visited the National Palace of Sintra. In doing so, they imprinted their distinct legacies and personal recollections onto its walls.

Throughout the years, the Palace expanded and also changed some of its already-existing decorations. During the reign of Manuel I (1469-1521), the Palace underwent enhancements that remain evident in its distinct attributes today, particularly the captivating Hispano-Moresque ceramic embellishments. It was under his direction that the impressive Room of the Coat of Arms was introduced. As his rule drew to a close, the Palace of Sintra stood among Portugal's most magnificent, its chambers adorned with the gilded riches acquired from lands colonized by the Portuguese during that time.

In the aftermath of the famous earthquake in 1755, which dealt a severe blow to this compound, the Palace of Sintra underwent a process of restoration that maintained its prevailing outline, consistent with the form it had possessed since the mid-16th century. This silhouette endures to the present day.

Nonetheless, it was not until the latter part of the 1930s that the Palace commenced welcoming the public on a consistent basis, adopting a museum-like approach. Over the course of this era, extensive efforts were undertaken to present a depiction of Portugal's majestic history. This period coincided with the authoritarian rule of the Estado Novo regime (1933-1975), which emphasized Portugal's favourable contributions to global advancement, even if it tended to downplay the full extent of its repercussions.

In contemporary times, the Palace of Sintra has experienced a resurgence, reclaiming its status as a paramount cultural hub at the core of Sintra. It stands as an essential component of the Cultural Landscape of Sintra, a UNESCO-designated World Heritage site since December 6, 1995.

Since September 2012, the oversight of this monument has been under the stewardship of Parques de Sintra, and in 2013, it proudly became a part of the European Royal Residences Network.

For a more comprehensive understanding of the influence this project will have, it is essential to include critical information regarding the diversity of objects and materials found in the Palácio Nacional de Sintra, both in terms of the production of these objects and the structure of the building itself. It is also important to discuss the challenges involved in their conservation, taking into account factors such as Sintra's climate and the increasing number of tourists.

The Parques de Sintra website offers detailed insights into the collections within the Palácio Nacional de Sintra⁹. The Palace houses a diverse range of artistic and architectural elements. Notably, its collections include paintings, ceramic tiles, ceramics, and an array of wooden materials. These woods are utilized in construction, furniture crafting, and the creation of frames for paintings, highlighting the historical craftsmanship techniques. Additionally, the palace features various metal elements as part of its collection, contributing

⁹<https://www.parquesdesintra.pt/en/learn-at-home/discover-the-collections-of-the-palaces/national-palace-of-sintra/>

to its historical and artistic significance.

Furthermore, the International Council of Museums - Committee for Conservation (ICOM-CC) has released a noteworthy declaration concerning environmental guidelines and conservation practices, which holds particular relevance [18]. This declaration, issued by a globally recognized authority in the field of conservation, outlines a comprehensive framework for managing environmental factors that can impact the integrity of museum collections and heritage sites. It delves into key considerations such as temperature control, humidity management, lighting standards, and the mitigation of pollutants, offering valuable insights into the preservation of cultural artefacts and architectural heritage within the context of the Palácio Nacional de Sintra.

It is crucial to consider how various environmental conditions can affect the preservation of the materials within the palace. The South-West Museums, Libraries, and Archives Council's factsheet on Preventive Conservation offers insightful guidance in this regard [35]. The factsheet discusses strategies and best practices for safeguarding valuable artefacts and architectural elements from factors such as temperature fluctuations, humidity, light exposure, and pollutants, all of which can significantly impact the long-term preservation of historical items. This resource serves as a valuable reference for understanding and implementing preventive conservation measures within the Palácio Nacional de Sintra.

2.6 Discussion

This chapter starts by providing a detailed explanation of the Digital Twin concept and showcases its variations and misconceptions. It is important to ensure that the concepts are well-defined. Some advantages of using a Digital Twin system are presented in Section 2.1.1, showing some benefits of the Digital Twin use and how it can help solve critical issues, including the problem this dissertation tries to solve. Historic Digital Twin comprises historical and contextual information as part of the Digital Twin that can be used to assess risk situations and make decisions. This is very valuable when discussing environments and artefacts with a high cultural value, since there are many types of information that can vary depending on the artefact, that are crucial when planning and executing conservation strategies.

Section 2.2 presents the BIM and HBIM, which are core parts of a Digital Twin system since they serve as an example of a data and object linkage model.

Section 2.3 explains basic concepts regarding adjacent fields of study tightly related to a Digital Twin system. These systems usually provide and treat the data for the Digital Twin. Internet of Things technology plays a big role in the Digital Twin environment since it is often responsible for the collection of data and its integration into the digital Twin system. Artificial Intelligence can be used in combination with a Digital Twin system for many uses, like assessing the status of varied physical systems without stopping the activities. Augmented and Virtual Reality are recent technologies that can complement

a Digital Twin system by enhancing the user's perception and perspectives on the 3D model.

Section 2.4 explains some concepts related to Heritage Conservation relevant to this dissertation. This is important because the target objects have high cultural and historical value and must be preserved. There are several strategies to achieve conservation, and the best choice ultimately depends on the artefact's current condition, the available resources and the external conditions it is affected by.

Finally, Section 2.5 provides an introduction to the National Palace of Sintra history. This is the place where the working project will be placed in the future, and as such, it is important to understand part of its history and the consequences it had on the Palace over time. It is important for conservationists to know the present materials and what conditions them the most. The developed application will help the palace's conservationists manage the distribution of the artefacts along their exposition rooms, based on the environmental data provided and the artefacts' compositional information.

RELATED WORK

The purpose of this chapter is to present the related work on the subject of the dissertation, referencing some systems, concepts and techniques relevant to the main technologies and concepts used in the developed solution.

Section 3.1 presents some significant areas in which DT is being applied, with great hopes of further developments in the future. Specifically, Section 3.1.1 showcases this technology applied to smart cities, manufacturing, and health, while in Section 3.1.2, we switch to mobile development, AR and Virtual Reality (VR).

Section 3.2 presents works related to the preservation of cultural heritage, with a specialized information model in Section 3.2.1, the advocacy of preventive conservation in Section 3.2.2, and the description of an ontology to support a Digital Twin in Section 3.2.3.

Finally, Section 3.3 is reserved for discussing some challenges that are usually present while developing and maintaining a DT.

3.1 Various uses of Digital Twin

This Section presents various applications of DT and an example of a mobile app using AR to allow for a better visualization/interpretation of a heritage site. This can be observed in Sections 3.1.1 and 3.1.2, respectively.

3.1.1 Digital Twin applications in different areas

Fuller et al. [8] studied the potential of DT, discussing the domain, sectors, and specific problems for this technology. According to this, many industries are growing and starting to use DT in their operations, and the rapid growth in IoT and AI played a crucial part in it. However, certain industries have shown more progress than others, the primary ones being Manufacturing, Smart Cities, and Healthcare. Their distribution is depicted in Figure 3.1.

Smart Cities:

The way cities are designed, constructed, and run has the potential to change thanks to digital twins. A city's infrastructure, including its structures, roadways, utilities, and

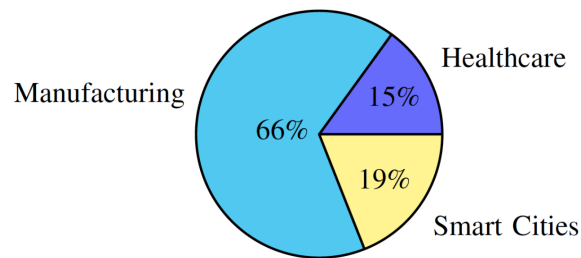


Figure 3.1: Percentage share of research areas using Digital Twin [8]

other physical assets, would be represented virtually in a digital twin of the city. Before making any changes in the actual world, city planners and engineers can evaluate and optimize the city's various systems' performance by performing simulations in a virtual environment.

Real-time monitoring and management of the city via digital twins would also enable proactive maintenance and effective resource management. For instance, a digital twin may spot and address flaws with the city's electrical or water systems before they worsen.

Manufacturing and Industry:

Digital twins have the ability to completely change how a business is conducted in the manufacturing and industrial sectors. For instance, a virtual version of the plant's machinery, procedures, and systems would be a digital twin of a manufacturing facility. Manufacturers can evaluate and optimize the plant's performance before making any modifications in the real world by replicating the plant's activities in a virtual environment.

A Digital Twin, for instance, may be used to model how a new manufacturing process will affect output rates, energy use, and waste creation. To increase productivity and cut expenses, the layout of the facility could also be optimized.

Real-time monitoring and management of the plant using digital twins would also enable preventative maintenance and effective resource management. For instance, a digital twin may be used to spot and address equipment concerns at the factory before they worsen.

Overall, digital twins have the potential to dramatically increase manufacturing and industrial sustainability, cost-effectiveness, and environmental friendliness.

Healthcare:

The use of digital twins in healthcare, even though still in the initial phases of development, can completely change how care is provided. The use of digital twins in healthcare enables the simulation and virtual portrayal of illnesses and treatment choices. Healthcare professionals can evaluate and optimize treatment plans before putting them into practice in the real world by replicating these scenarios in a virtual setting.

In addition to these uses, digital twins in healthcare might be employed to teach medical personnel, enabling them to polish their abilities in a virtual setting before

treating patients in the real world.

Bridging into an urban application, the design and operation of healthcare institutions could be optimized using digital twins. For instance, a hospital's digital twin might mimic and optimize patient flow through the establishment, increasing effectiveness and decreasing wait times.

A very ambitious but revolutionary application is the DT of a human. Healthcare professionals can simulate and evaluate a patient's response to various treatments, enabling the development of personalized treatment programs.

3.1.2 Digital Twin for Heritage Interpretation in a Mobile App

By creating a DT of a heritage site, historians and educators can analyze and interpret the site in new ways and use technology such as 3D modelling and AR to explore the site in detail. This creates the opportunity to create immersive experiences that allow visitors to explore the site more interactively and engagingly compared to a physical visit to the site. For example, an AR experience at a heritage site might allow visitors to see how the site looked in the past or to learn more about the people and events associated with the site.

The Pampulha Modern Ensemble includes the St. Francis of Assisi Church¹, located in Belo Horizonte, Brazil and designed by the architect Oscar Niemeyer in the modern organic style, was the first modern architectural monument in Brazil and was completed in 1943. Due to its universal value, the Ensemble is listed under UNESCO's World Heritage Convention².

An AR app was built by Dezen-Kempton et al. [6] to consolidate and disseminate heritage information regarding the Ballroom and the St. Francis of Assisi Church, part of the Pampulha Modern Ensemble.

The HBIM model was used to account for the historical parameters of the object in the DT. The DT creation employed Terrestrial Laser Scanning (TSL)³ and a low cost Unmanned Aerial Vehicle (UAV)⁴. The process was based on three fundamental steps:

- Collection of spatial and documentary data;
- Data processing and Dense Surface Model (DSM)⁵ creation;
- HBIM modelling.

The detailed process is depicted in Figure 3.2.

The app uses different visualization techniques and filters, allowing visualization and interaction with specific design patterns. This gives important insights that could not be obtained by merely reading about or looking at photos of those buildings in books.

¹https://en.wikipedia.org/wiki/Church_of_Saint_Francis_of_Assisi

²<https://whc.unesco.org/en/list/1493>

³<https://www.laserscanning-europe.com/en/servicesdevices/terrestrial-laser-scanners>

⁴https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

⁵<https://workshops.distancesampling.org/duke-spatial-2015/slides/4-what-is-a-dsm.html>

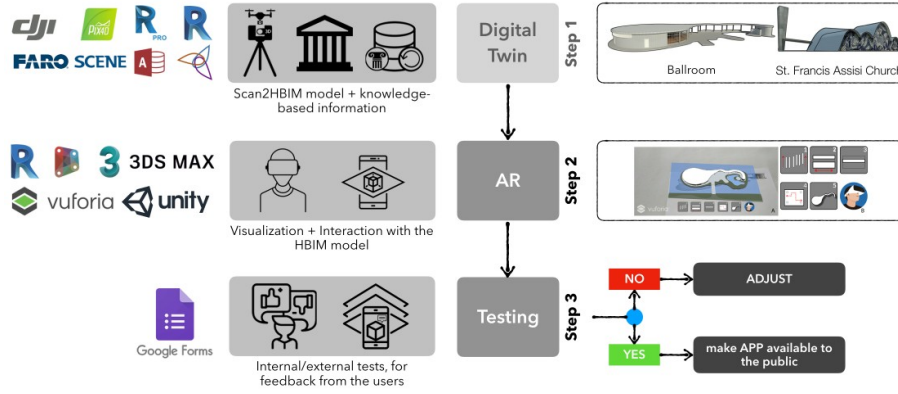


Figure 3.2: Roadmap of the AR creation and testing from the Digital Twin of Niemeyer's Pampulha Buildings [6].

Furthermore, the textures used are based on real ones acquired during data collection, allowing specialists to study them and the average user to enjoy them.

3.2 Preservation of Cultural Heritage using the Digital Twin

This separate Section is dedicated to the studies on Artefact Heritage Preservation using DT due to its relevance.

In Section 3.2.1, a survey about the combined use of an Integrated Informative System and a DT is presented. In contrast, in Section 3.2.2, a study focusing on the preventive conservation of artefacts is presented.

3.2.1 Combining Integrated Informative System and Historical Digital Twin

To define and validate qualitative and quantitative indicators of the current state of artistic heritage in an interdisciplinary framework, a rational approach to preservation is based on traditional methods supported and integrated by novel technologies.

With the need to protect cultural heritage from natural and artificial risks, Marra et al. [24] proposed a method for the upkeep and preservation of artworks housed in a museum that is based on a specific digital pathway leading to a HDT.

Major seismic events' devastation has brought attention to the need for risk reduction and asset protection for buildings and works of art. However, the uniqueness of this cultural evidence and the interplay between structural and non-structural elements, specifically between structures and artistic assets, make these objectives extremely difficult to accomplish.

This approach uses the intricate three-dimensional model produced by a laser scanning survey to estimate the stress regime and the dynamic parameters of two sculptures.

The proposed methodology is based on an integrative and systemic approach. Through the approach's several phases, it is feasible to gain a thorough understanding of the artefacts

3.2. PRESERVATION OF CULTURAL HERITAGE USING THE DIGITAL TWIN

and an overall integrated perspective. Figure 3.3 depicts the procedure, which aims to enhance conservation and safety through particular technological and scientific activities.

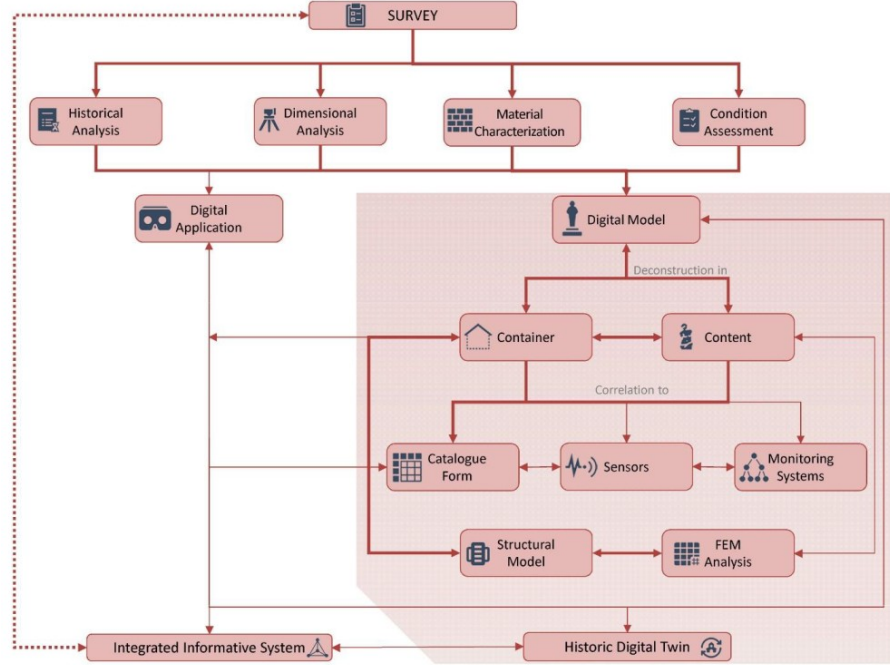


Figure 3.3: General framework of the research workflow [24].

The key phase of the process is represented by the survey of the artefact, which is divided into four moments:

- **Historical Analysis:** historical, artistic, and architectural analysis of the artefact, resulting in a summary of the history, transformations, restorations, and partial destruction that may have affected the artefact along with the mechanical properties of the materials;
- **Dimensional Analysis:** a direct or indirect metric survey serves as a helpful foundation for the graphical representation, allowing for the creation of a two- and three-dimensional representation of the asset as a whole and in each of its components;
- **Metrical Characterization:** Catalogue of materials and construction techniques;
- **Condition Assessment:** Catalogue of degradation and damage;

This procedure makes the development of an **Integrated Information System (IIS)** possible. The foundation for developing the **HDT** is the implementation of the IIS and its linkage to dynamic data, specifically the monitoring systems connected to the IIS (like moisture, light, or temperature detection systems) and the automated processing procedure of the information gathered in the system.

Some parts of this procedure were tested in two artworks exhibited in the Archaeological Museum of the Ancient Capua⁶. The museum is part of a 19th-century building.

The two art pieces analyzed, due to their cultural value, were the Mater Matuta (Figure 3.4(a)) and the Resting Satyr (Figure 3.4(b)).



(a) The Mater Matuta.

(b) The Resting Satyr.

Figure 3.4: The Mater Matuta and the Resting Satyr [24].

For extensive assessments, a three-dimensional model created using 3D laser scanning technology was used to get the main geometrical features of the statues, and photogrammetric research was employed to create a photo-realistic depiction with detailed information about the state of conservation.

Individual analysis was conducted with its results being digitalized with a tool that can assess the state of conservation and main seismic responses, among others. After this, an analysis of the structural components of the museum building was done in order to provide a better description of the relationship between the art pieces and the building.

With this, the IIS was defined.

Combining these results with several layers of digital processing (some results shown in 3.5) and filtering to analyze structural behaviours (with the possibility of using deep

⁶<https://www.opencampania.it/en/poi/Museo-Archeologico>

learning algorithms for better results), the researchers were able to assess the level of conservation of each statue. With this data, it was also possible to run seismic simulations to predict the statue's physical response.



Figure 3.5: From the very dense scanned and tessellated model to the FE (Finite Element) mesh model, passing through the quad mesh and the surface model (from left to right) [24].

The researchers concluded that, although the data gathered and the outcomes in validating several critical elements of the proposed integrated approach are positive, additional work and research are required for widespread practical implementations. However, the results also demonstrate that [IIS](#) and [HDT](#) may enable the linkage of digital replicas of cultural assets to other databases or web platforms, concluding new processes based on the automation of data integration and analysis.

3.2.2 Using the Digital Twin in Preventive Conservation

As mentioned in Section [2.4](#), preventive conservation refers to a range of actions and initiatives designed to stop or lessen the likelihood of further deterioration or loss of cultural heritage artefacts [19]. Because they do not include alterations to the objects' materials or structures, these actions are notable for maintaining the objects' original appearance.

According to Van Balen [41], preventative conservation methods, primarily based on routine maintenance and monitoring work, enable better preservation of intangible cultural property and a decrease in the expenditures associated with conservation operations. However, the total recovery of damaged historical objects and places is never possible, and necessary repair work entails significant costs. Instead, the preventative strategy advocates acting preemptively through early diagnosis and regular site monitoring to comprehend and diminish the deteriorating process.

The drastic changes in heritage conservation practice observed over the past century further reinforce the issues faced by professionals in heritage documentation. These issues exist due to the increased variety and complexity of information sources. This pushes researchers to increasingly consider the [HBIM](#) model as a solution. Pierre Jouan et al. [20] suggest the application of [DT](#) principles, using [HBIM](#) models as a digital replica, to support the preventive conservation of heritage places.

The process is divided into four steps and should be applied as a continuous loop instead of in a linear fashion:

- **Analysis:** It strives to give a good understanding of the context and define the scope of the asset and its environment.
- **Diagnosis:** Its goal is to provide an accurate analysis of the defects and risks of the asset with a basis for experts to propose a solution.
- **Therapy:** It identifies the purpose of the conservation work and defines the appropriate conservation strategy.
- **Control:** The actual implementation of the conservation policies with strict monitoring and possible future changes.

Figure 3.6 showcases this process in more detail.

3.2.3 Ontology supporting Digital Twin

Digital representations of heritage assets must be able to perceive all of their possible representations (these could be buildings, music, objects, landscapes, etc.) and the connections between the various parts of the assets they represent.

A formal description of knowledge called ontology outlines the ideas and connections inside a domain. An ontology can be used in the context of cultural heritage to depict the various components of heritage, such as the structures, artefacts, and landscapes, as well as the connections among them.

Niccolucci et al. [27] define an [HDT](#) as a digital representation of an asset with all its surrounding context and knowledge derived from one HDT ontology.

The researchers describe their proposed ontology by grouping the concepts into classes. All classes are subclasses of an empty class called *Heritage Entity*. The core classes are the *Heritage Asset* and the *Heritage Digital Twin*. The *Heritage Asset* represents, as the name suggests, an actual heritage asset that can be related to instances of the same class, while the *Heritage Digital Twin* represents the digital information associated with the *Heritage Asset*. The *Heritage Document* is a class that defines all documentation related to an *Heritage Asset*, ranging from digital to physical objects. As the need arises, more classes can be added to the model.

Some other important classes of the model:

3.2. PRESERVATION OF CULTURAL HERITAGE USING THE DIGITAL TWIN

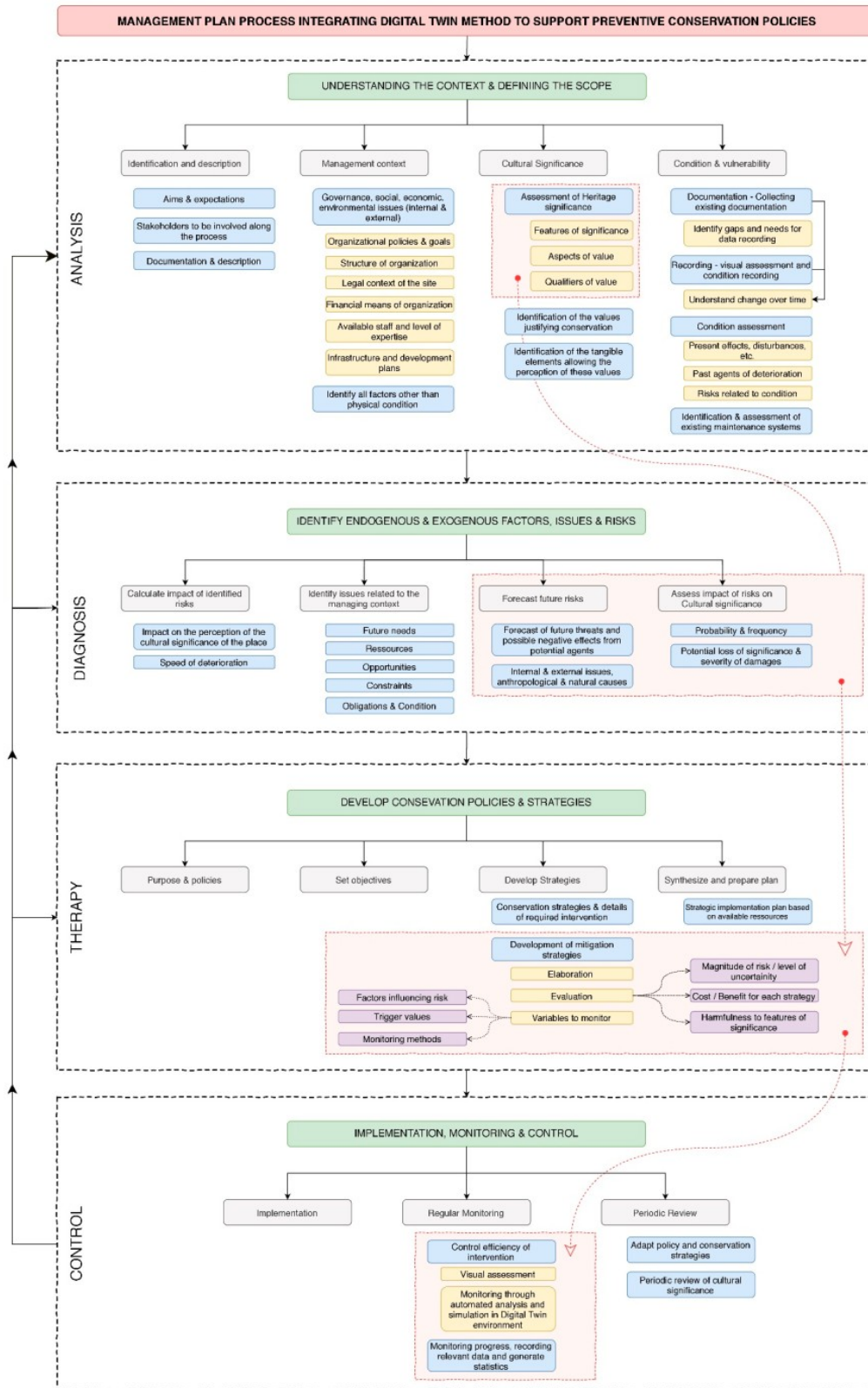


Figure 3.6: The complete value-based risk management plan process. The phases highlighted in red are the key steps toward implementing the method. [20]

- *Heritage Location*: location of the asset;
- *Heritage Story*: narrative, historical source, etc;
- *Heritage Document*: data created to enrich the asset's knowledge;
- *Heritage Documenting*: activity of producing a document;
- *Heritage Storytelling*: creating or spreading a story;

Figure 3.7 showcases the semantic graph created.

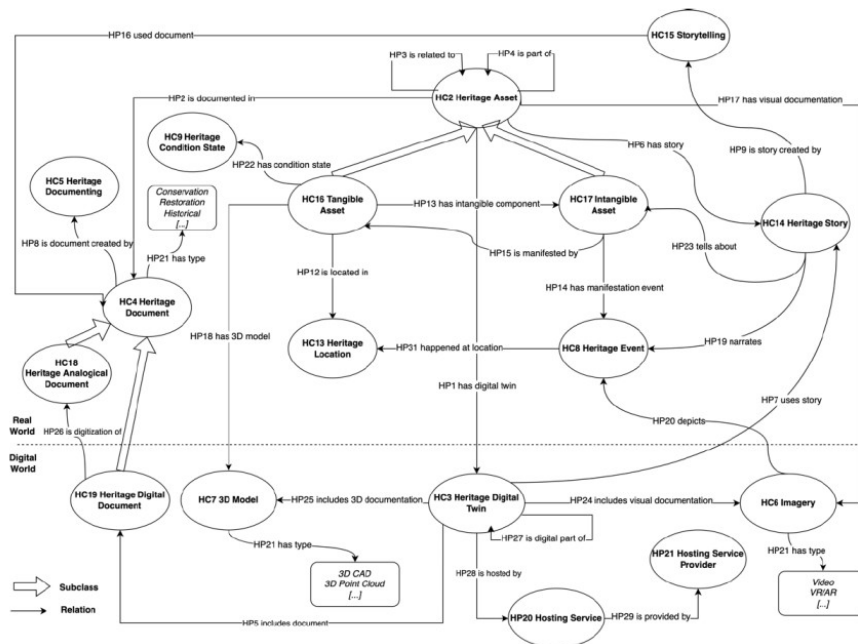


Figure 3.7: The semantic graph of the HDT ontology. HC: HeritageClass. HP: Heritage Property [27].

A heritage Knowledge Base (KB), which manages information organized through the HDT ontology approach, can be created using the HDT ontology, where information about heritage is arranged in a knowledge graph and managed and searched using a NoSQL DBMS⁷.

A KB can be used not only by humans but also by machines. A DT must respond to real-time stimuli, usually delivered by sensors or external information (like a weather service). As depicted in Figure 3.8 for a flood risk procedure, instructions and measures can be taken to ensure asset safety based on the KB. In this example, the system checks the asset's type and location after receiving a flood alert to check for possible dangers and consequent safety measures.

The HDT ontology was applied to Giorgio Vasari's "Last Supper"⁸ to demonstrate how it can be used to document the assets features. The painting is currently exhibited

⁷<https://www.mongodb.com/nosql-explained>

⁸<https://www.santacroceopera.it/en/catalogue-of-works/last-supper-vasari/>

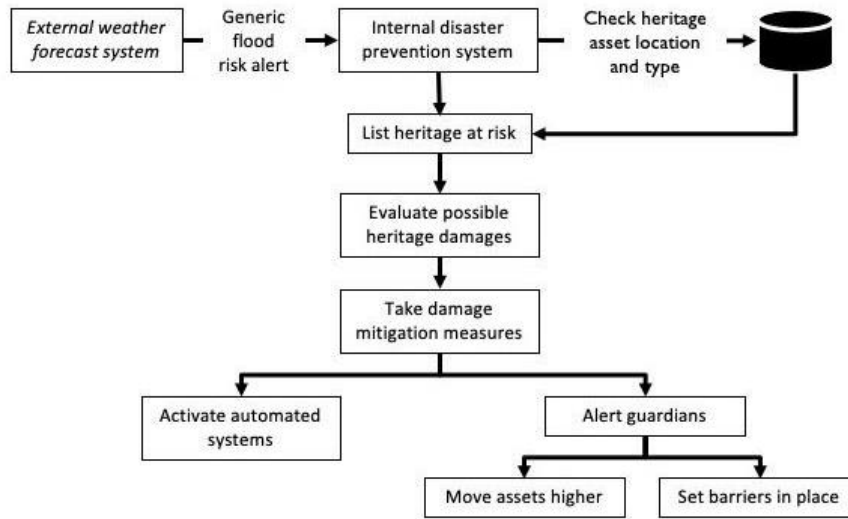


Figure 3.8: Schema of direct action activated by an automatic alert system based on HDT data [27].

in Santa Croce Museum in Florence ⁹. In 1966 it suffered greatly as a consequence of the flooding of the Arno River ¹⁰, and after a long period of restoration, it was placed back in the museum.

Part of a semantic description of this story using the HDT ontology is presented in Figure 3.9. In this example, we can see the definition of several tangible assets and their properties, like location, condition, etc.

The authors admit that the ontology needs further refinement but that with appropriate metadata, the system provides the necessary features for the user to manage and query information about the assets.

3.3 Digital Twin Challenges

According to Fuller et al. [8], the existing literature lacks unified models or a general Digital Twin design, and there is no agreement on the best way to create a Digital Twin system.

It is a common objective for researchers to understand how to handle such systems because the Digital Twin environment is heterogeneous and connected to extensive distributed networks.

The diverse ecosystems used for creating and developing current systems are one key obstacle. These occur due to the various study disciplines involved in collaboration, which can help with breakthroughs but also be a burden because aims in several sectors ultimately steer the research in different directions, slowing down outcomes. The absence of standardization is an additional difficulty for digital twins. As a result, the projects

⁹<https://www.santacroceopera.it/en/places/museum/>

¹⁰<https://en.wikipedia.org/wiki/Arno>

S. Croce Church and Convent (HC16 Tangible Asset)

HP12 is located in **Piazza S. Croce, Florence, Italy** (HC13 Heritage Location)

HP4 is part of **Florence Historic Centre** (HC1 Heritage Entity)

crm:P2 has type "UNESCO WHS" (crm:E55 Type)

Refectory in S. Croce Church and Convent (HC16 Tangible Asset)

HP12 is located in Cloister of S. Croce Church, Piazza Santa Croce, Florence, Italy (HC13 Heritage Location)

HP4 is part of S. Croce Church and Convent (HC16 Tangible Asset)

Murate Convent (HC16 Tangible Asset)

HP12 is located in **Via Ghibellina 2, Florence, Italy** (HC13 Heritage Location)

HP4 is part of **Florence Historic Centre** (HC1 Heritage Entity)

crm:P2 has type "UNESCO WHS" (E55 Type)

Painting Ultima Cena (HC16 Tangible Asset)

HP30 is titled "Ultima Cena" (HC12 Heritage Title)

HP30 is titled "Last Supper" (HC12 Heritage Title)

HP10 is identified by "1234-5678" (HC 11 Heritage Identifier)

HP12 is located in Refectory in S. Croce Church and Convent, Florence, Italy (HC13 Heritage Location)

HP32 was affected by Florence Flooding of the Arno River in 1966 (crm:E5 Event)

crm:P31 was modified by Restoration of Vasari's Last Supper (crm:E11 Modification)

crm:P14 carried out by Opificio delle Pietre Dure (crm:E39 Actor)

HP2 is documented in Vasari's Last Supper Restoration (HC4 Heritage Document)

HP22 has condition state Restored (HC9 Heritage Condition State)

crm:P27 moved from location of Murate Convent in Florence, Italy (HC13 Heritage Location)

crm:P28 moved to location of S.Croce Church and Convent in Florence, Italy (HC13 Heritage Location)

Figure 3.9: Part of a semantic description using the HDT ontology [27].

using the Digital Twin differ from one another. The diversity of definitions seen is a contributing element; combined with a lack of standardization, it is a challenge, impeding the development of digital twin technology.

A DT system usually has a tight relationship with IoT since most data analyzed originate from IoT sensors. With real-time data being delivered by these sensors, there is a considerable flux of unstructured data that needs to be organized. This and the fact that the data are often sensitive imposes a possible threat to security, which poses a big challenge. Also, the risk of DDoS¹¹ attacks increases because of the rapid growth of IoT.

3.4 Discussion

Section 3.1 of this chapter presents some use cases of DT to allow a better understanding of where this technology can be efficiently used. Section 3.1.2 depicts a project combining

¹¹<https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>

a mobile application containing AR technology with a DT system. This mobile application allows users to explore and study, through their mobile device, a digital replica of a building with high cultural value.

Section 3.2 is reserved for studying strategies and policies regarding the conservation of cultural heritage. Section 3.2.1 proposes a method to ensure the preservation of artworks based on a IIS that serves as a foundation for the HDT. These concepts require a deeper analysis of an object's characteristics and the circumstances of its environment. While not directly applied in this initial phase of the platform, these concepts are integral to a heritage DT project and will prove valuable in subsequent stages.

Section 3.2.2 refers to preventive conservation, an approach that keeps gaining relevance and support. Here, the key is to use a HBIM model as a digital replica. The process of building this model is also discussed. Preventive conservation plays a significant role in our approach and is realized through the use of sensors and the data they provide. The ultimate goal is to empower users to make informed decisions about where to position objects to ensure their optimal conservation.

In Section 3.2.3, an ontology made to represent all information related to a DT is presented. This ontology allows for easier and more precise exploration of information by the user. This ontology is an example which may be, in the future, adapted and incorporated into the platform to enable an effective response to certain sensor readings.

In Section 3.3, some challenges that DT systems encounter are enumerated and discussed. The consideration of challenges is an ongoing process throughout the platform's development. These challenges are evaluated and continually addressed in subsequent phases.

DEFINING THE SOLUTION

In the context of this dissertation, the primary objective is to develop an interactive platform that enables users to monitor and explore 3D models of heritage artefacts. This chapter serves as a comprehensive exposition, delineating the ideation and process definition that form the foundation of the project. It outlines the requirements, establishes the project environment, and elaborates on the system architecture.

Section 4.1 is dedicated to defining the core requirements of the developed solution, covering both functional and non-functional aspects. This section aims to provide a clear and structured framework that guides the subsequent phases of development. In Section 4.2, an examination of the chosen technologies is presented. This section elucidates the thought process and rationale underlying the selection of technologies, reflecting an approach that aligns with the project's goals. Section 4.3 explores the system's architecture, the Digital Twin environment and the platform's functionality. Finally, Section 4.4 showcases a digital scanner workshop that was done on the university campus, detailing some of the scanners acquired and how to use them.

4.1 Requirements

The functional and non-functional requirements of the established comprehensive solution are laid out, taking into account the major objectives of the dissertation and the proposed solution system. These specifications form the basis for designing and implementing a successful solution that meets the established goals and expectations.

The dissertation is able to proceed with a well-defined roadmap for the development process, leading to the effective implementation of the proposed solution by explicitly specifying both functional and non-functional requirements.

4.1.1 Functional Requirements

Functional requirements are essential elements that play a fundamental role in shaping the capabilities and behaviour of the solution. They serve as a detailed blueprint, outlining the specific functionalities that the system must possess to fulfil its intended purpose

successfully. These requirements encompass a comprehensive list of tasks, features, and interactions that the system should support.

Each functional requirement acts as a building block, contributing to the overall functionality of the solution. As the backbone of the development process, functional requirements guide the creation of the system's architecture, the design of user interfaces, and the implementation of core features. These are the main functional requirements identified for this project:

- **Import 3D models of artefacts** - Ability to easily and directly import 3D representations of the artefacts on the platform, mostly captured by digital scanning, during runtime. This enables seamless integration of digital scanning results into the platform's runtime environment.
- **Interact with and manage the imported artefacts** - Possibility to interact with the imported 3D artefacts through a user-friendly interface. This includes moving, rotating, zooming, monitoring, and exploring the imported representations dynamically.
- **Import sensors, add and visualize data** - Users can precisely monitor specific aspects and areas of interest by providing the flexibility to place new sensors in the environment. Furthermore, the system should facilitate the easy addition of data to these sensors, allowing for the incorporation of various types of information. Once the data is integrated, users can visualize it through representations like a graph, providing valuable insights for analysis and decision-making. With these, the platform becomes a robust tool for data-driven exploration and monitoring of the environment.
- **Save and Load progress** - The platform should enable users to effectively save and load their progress during runtime. With this capability, users can securely store the entire environment, including imported artefacts, placed sensors, and associated data. It facilitates continuity, collaboration, and experimentation, giving users the confidence to explore and experiment with various setups and data without the fear of losing progress or data integrity.

4.1.2 Non-functional Requirements

In the context of digital twins and their support systems, non-functional requirements play a crucial role in ensuring the overall success and effectiveness of the solution. These requirements are concerned with the quality attributes and characteristics of the system rather than specific functionalities. These are the main non-functional requirements identified for this project:

- **Performance** is a key non-functional requirement that addresses the system's responsiveness and efficiency. This is very important when building a platform that

interacts with complex data-heavy 3D artefacts. Digital twins entail real-time data processing, low latency, and rapid simulation capabilities to provide timely and accurate data changes from the physical counterpart.

- **Reliability** is another essential aspect, as the process of managing the 3D artefacts and the digital twins are relied upon for critical decision-making processes. The solution must consistently deliver accurate and dependable results, minimizing errors and discrepancies between the virtual and physical systems.
- **Usability** refers to the user-friendliness of the artefact and digital twin support system. A well-designed interface and intuitive user interactions are vital to facilitate ease of use and enable stakeholders to effectively interact with and leverage the artefacts and the digital twins.
- **Scalability** is a critical non-functional requirement, particularly considering the platform's and digital twin ecosystem's potential growth and expansion. The solution should be capable of accommodating increased data volumes and complexity, supporting larger-scale simulations and accommodating future enhancements.

4.2 Technologies

This project was developed entirely on Unity due to its flexibility and varied available resources. The decision to undertake this project entirely on Unity was primarily influenced by the platform's exceptional flexibility and diverse range of resources. Unity's versatility allowed for the seamless integration of various components, enabling us to create a dynamic and immersive experience tailored to our project's unique requirements.

Throughout the development process, two essential plugins, OBJLoader^{1,2} and Graph and Chart³ were used to fit specific needs of the project. OBJLoader is a free plugin that provides methods to load OBJ files in runtime, and Graph and Chart is a paid plugin that provides graph and chart presets, functions and more. The first plugin, OBJLoader, proved to be an invaluable asset as it offered free runtime methods to load OBJ files effortlessly. This capability significantly expedited the process of importing 3D models and ensuring real-time interaction with the imported artefacts. However, the only downside we encountered was that it had not received updates since March 2019 and is no longer actively maintained. Although this situation had the potential to cause significant issues, fortunately, we managed to navigate through the project without encountering any major problems.

Though a paid resource, the second plugin, Graph and Chart, delivered tremendous value with its comprehensive collection of pre-designed graph and chart presets along

¹<https://github.com/PhalanxHead/UnityRuntimeOBJLoaderDocs>

²<https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547>

³<https://assetstore.unity.com/packages/tools/gui/graph-and-chart-78488>

with a plethora of functional capabilities. By leveraging this plugin, we could present data and analytics in an organized and visually appealing manner, enriching the overall user experience and enhancing the project's data visualization capabilities.

Combining Unity's inherent flexibility with the contributions of OBJLoader and Graph and Chart streamlined the development workflow and empowered the development of the solution.

Scanner Workshop:

DCR and Department of Computer Science (DI) have acquired three Artec⁴ Scanners to use on a variety of projects, including this dissertation's work. As such, a workshop to learn how to use these scanners, hosted by Artec, took place in FCT's campus.

The scanners acquired all function using structured-light⁵ and serve different purposes; they each have preferred object sizes to work with. Some models work better with large objects, while others work better or only function with small objects. The three scanners acquired are:

- **Artec Leo⁶:** Leo is a wireless scanner capable of capturing both medium-sized objects such as gearboxes, castings, furniture, statues, and the human body, as well as large objects like automotive parts, turbines, ship propellers, small boats, and room interiors. It has a point accuracy value of up to 0.1 mm.
- **Artec Space Spider⁷:** Spider is a cabled scanner that can capture medium to small objects or fine detail with high precision. It has a point accuracy value of up to 0.05 mm.
- **Artec Micro⁸:** Micro is a Desktop scanner made to capture small objects. It has a point accuracy value of up to 10 microns.

The point accuracy measure represents the distance a captured point may deviate from its real position. Lower values represent a better result.

The workshop consisted of the display of the scanners and their features and characteristics, along with practical examples using several different objects. With this, we experimented and learned how to scan various objects and use Artec software to manage and process the output from the lasers and export the final digital model.

This work is relevant since the imported models of the 3D artefacts will be digitally scanned and later imported onto the platform.

⁴<https://www.artec3d.com/>

⁵<https://www.artec3d.com/learning-center/structured-light-3d-scanning>

⁶<https://www.artec3d.com/portable-3d-scanners/artec-leo>

⁷<https://www.artec3d.com/portable-3d-scanners/artec-spider>

⁸<https://www.artec3d.com/portable-3d-scanners/artec-micro>

4.3 System Design

To enhance the platform's capabilities, the ideal scenario, which had to be shortened due to time constraints, involves establishing a consistent and bi-directional data flow by integrating IoT sensors in the heritage environment or artefacts. This integration would enable real-time data acquisition from the physical surroundings, completing the Digital Twin model and providing a dynamic experience. The data collected from the IoT sensors would then need to be efficiently stored in a structured manner, ensuring easy accessibility and retrieval for processing within the platform whenever required.

Figure 4.1 showcases an example of the entire system architecture, depicting the interconnected components and their interactions. The diagram illustrates the integration of IoT sensors with the heritage environment or artefacts, highlighting the continuous flow of data between the physical and digital realms. The platform serves as the central hub, orchestrating data processing, visualization, and user interaction, offering users a holistic and immersive experience.

The server component is depicted in Figure 4.1, but it is currently not included in the project scope. As of now, there is no real-time data flow from the sensors to the application, making the server unnecessary. However, it is important to note that in the future, when this feature is implemented, the server will become an integral part of the solution.

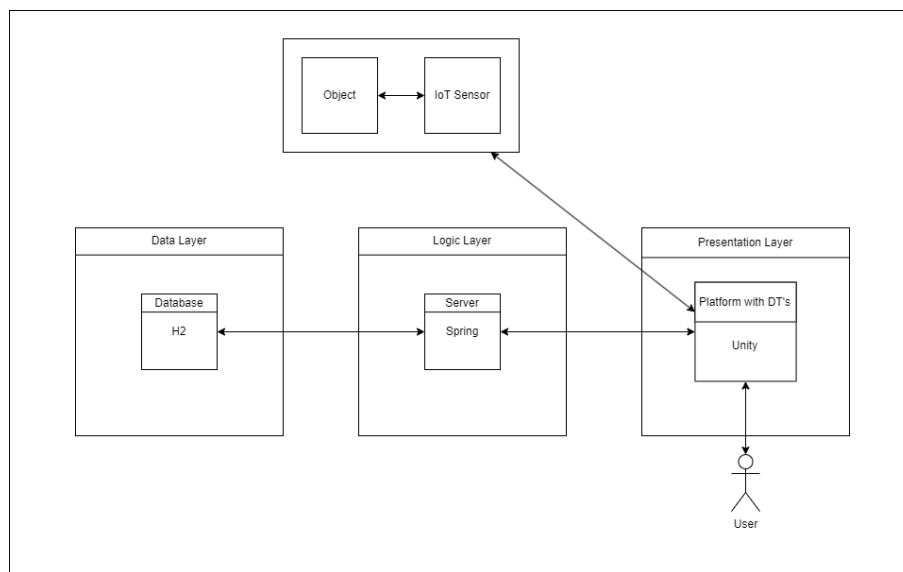


Figure 4.1: Full system architecture

By achieving this bi-directional data flow, the platform would be empowered to deliver real-time insights, foster data-driven decision-making, and provide users with a deeper understanding of the heritage environment.

In Figure 4.2, we can explore a more detailed view of the Digital Twin ecosystem, illustrating the bi-directional information flow between it and the IoT sensors.

The data collected by the IoT sensors serve as valuable input for the Digital Twin system. This continuous flow of information ensures that Digital Twin's virtual representation

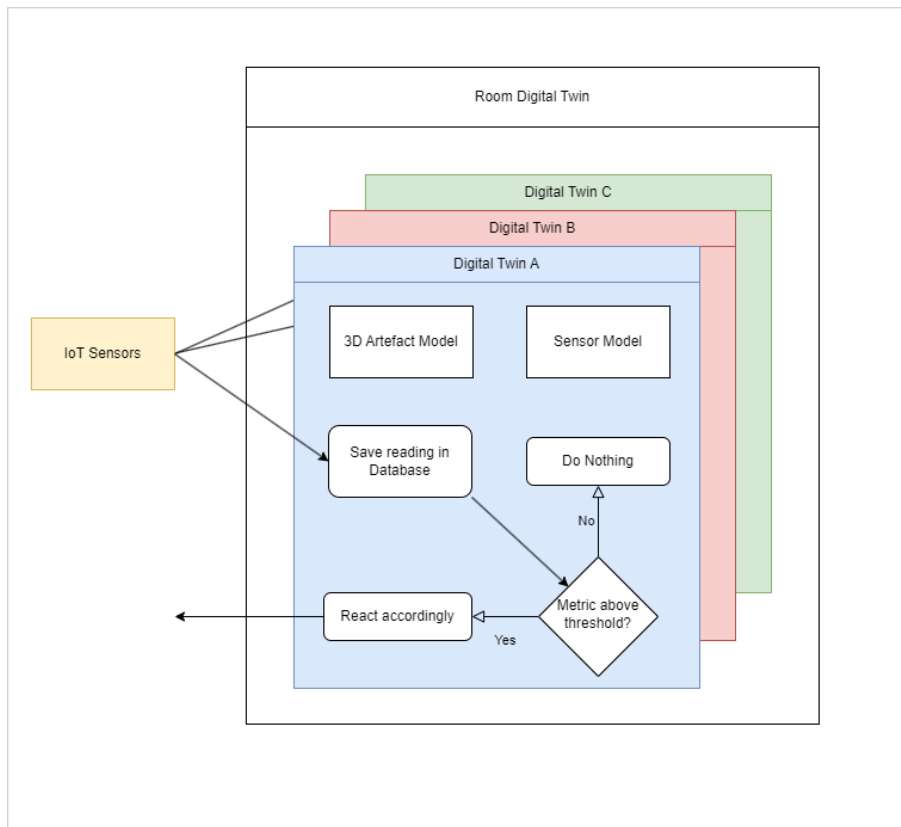


Figure 4.2: Digital Twin environment functionality

remains accurate and dynamic, reflecting the current state of the heritage environment. In response, the Digital Twin system processes this data in real time to generate a synchronized and interactive virtual representation. The bi-directional flow of data fosters a meaningful feedback loop, where the Digital Twin’s response can influence the physical environment through control mechanisms or recommendations.

Figure 4.3 presents an illustrative representation of the platform’s functionalities, providing a comprehensive overview of its capabilities.

First and foremost, the platform enables users to import OBJ files of heritage artefacts that have been previously 3D scanned. This seamless import process ensures that the digital representation of the artefacts is accurately integrated into the platform, preserving their intricate details and historical significance. Once the artefacts are imported, users can interact directly with the 3D models. The platform facilitates intuitive and user-friendly interactions, allowing users to perform various actions like moving, rotating, resizing, and selecting different visualizations of the imported artefacts.

In addition to interacting with the 3D models, the platform allows users to place sensors in the environment. These sensors act as data collectors, capturing real-time information from the physical surroundings or the artefacts. Considering this project’s scope, the user can add temperature and humidity data to each sensor. To facilitate data visualization and analysis, the platform offers tools to visualize the added sensor data

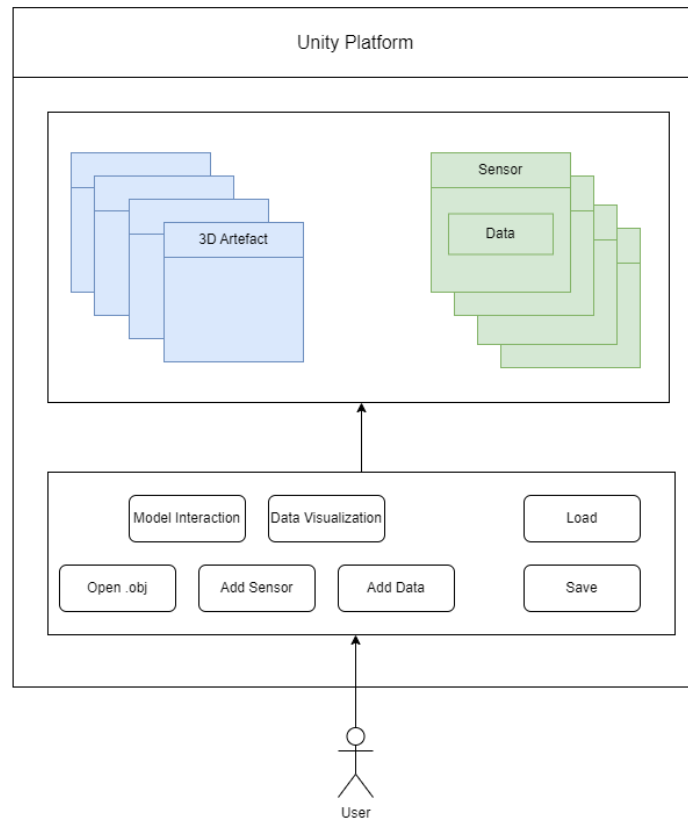


Figure 4.3: Unity platform's features and user interactions

effectively. This feature empowers users to gain insights into the heritage environment, identifying patterns, trends, or anomalies that may enhance their understanding of the artefacts or the broader historical context.

Furthermore, the platform allows users to save and load their progress, ensuring the continuity of their work and facilitating collaboration with other stakeholders. This feature enables users to seamlessly revisit previous sessions, compare data, or continue exploring.

4.4 Conclusions

The current chapter presents the necessary process that took part in defining this project's solution. The main objective is to create an interactive platform where users can add, manipulate and explore 3D models of artefacts, sensors and data related to these. The work was done keeping in mind that the end goal is to provide support for creating and handling Digital Twins in conservation.

With these considerations in mind, the functional and non-functional requirements are defined in Section 4.1 and the technologies used and some of the reasoning for the selections made are presented in Section 4.2.

Finally, the system architecture, the Digital Twin environment, how it would operate, and the platform's desired features and interactions are explored in [Section 4.3](#).

IMPLEMENTING THE SOLUTION

Based on the solution's conceptual framework, the implementation phase of the system was initiated. This phase has been subdivided into four primary modules, each comprehensively elucidated in this chapter's subsequent sections.

Each module was systematically developed with a methodical approach, contributing to establishing a cohesive and integrated system.

The start and, arguably, the most important feature of the project - 3D Model Importer, is described in Section 5.1. Section 5.2 then explains the process taken to complement the previous feature to allow the user to interact and explore the imported 3D models.

Subsequently, the mechanism of Save and Load is elucidated in Section 5.3, followed by the depiction of the process for adding sensors and related data to facilitate visualisation in Section 5.4.

The current dissertation's work is divided into four significant parts, each one of them explored and analysed in the subsequent sections. All the work completed in each phase was developed in accord with the knowledge obtained through the studied literature in Chapter 2 and Chapter 3, the specifications defined in Chapter 4 and to accommodate for the future full functionality of several Digital Twins integrated into the system.

After this study took place and the requirements and architecture of the system were defined, the implementation work began.

5.1 3D Model Importer

Unfortunately, Unity Engine does not offer any tools to import 3D objects or manage them in runtime, and, due to this, there were several frequent challenges throughout the project - the first one being the actual import of the 3D objects.

As mentioned in Chapter 4, the plugin OBJLoader^{1,2} from the Unity Asset Store was used to fill this critical need. As can be seen in the documentation, it provides the following features:

¹<https://github.com/PhalanxHead/UnityRuntimeOBJLoaderDocs>

²<https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547>

- Ability to load OBJ file from a stream;
- MTL file loading, and texture loading (dds, crn, jpg, jpeg, png, tga, bmp);
- Triangle, Quad, and Ngon import;
- Fast loading times;
- Works in both the editor and runtime;
- Usable in the editor.

For context, an OBJ file³ is a text-based format that encodes details about the geometry of a 3D object, including its vertices, faces, and textures of a 3D model. Also, an MTL file⁴ is a material settings file used in conjunction with an OBJ file that describes how a program should apply textures to that object.

The vital prerequisite for using this plugin entails navigating to the "Edit > Project Settings > Graphics" section within Unity; here lies the need to append "Standard (Specular Setup)" to the collection of shaders that are always incorporated. While this addition may lead to a slightly prolonged build time for the subsequent phase, it is an essential step to establish the necessary shader and shader variants for the plugin to function optimally.

The plugin offers a selection of functions, varying in the input taken. These are the options it provides:

- Load OBJ from an input stream. If there is no MTL file referenced in the input stream (i.e. by the OBJ file), the default Specular Shader (White) will be attached to the OBJ.;
- Load OBJ from an input stream and load the associated MTL from the MTL input stream.;
- Load an OBJ from the given path and the associated MTL from the given MTLPath;
- Load an OBJ from a given path.

The process of importing a 3D model of an artefact starts with the user pressing the "Import OBJ" button. That button is programmed to run the method "OnClick", which, in turn, triggers the start of a Coroutine featuring the *LoadModel* method, as illustrated in the Code Listing 5.15.

The *LoadModel* method is tasked with the management of the entire import process. It receives, as arguments, the paths obtained in the OnClick function through the *StandaloneFileBrowser*⁵ and reads the content located in the designated path into an array of bytes. This array will be turned into a *MemoryStream*, which allows the file's data

³<https://jinolo.com/blog/obj-file-overview>

⁴<https://www.loc.gov/preservation/digital/formats/fdd/fdd000508.shtml>

⁵<https://github.com/gkngkc/UnityStandaloneFileBrowser>

to be treated as a stream that can be processed by subsequent operations. The streams (containing the data from the OBJ and the MTL files) are then passed into the Load method of the OBJLoader class. This method will return a GameObject representing the loaded 3D model.

It is necessary to save the path from which the artefact is being imported so that later, when saving the progress of the application, the original files can be copied to avoid new file generations, which are more difficult and inconsistent. This will be detailed in Section 5.3.1.

```
1 public static void LoadModel(string url, string mtl)
2 {
3     byte[] objBytes = File.ReadAllBytes(url);
4     MemoryStream objStream = new MemoryStream(objBytes);
5
6     byte[] mtlBytes = File.ReadAllBytes(mtl);
7     MemoryStream mtlStream = new MemoryStream(mtlBytes);
8
9     GameObject model = new OBJLoader().Load(objStream, mtlStream);
10
11     model.name += counter++;
12     model.tag = tagName;
13
14     AddComponents(model);
15     int lastIndex = url.LastIndexOf('\\');
16     string result = "";
17     if (lastIndex >= 0)
18     {
19         result = url.Substring(0, lastIndex);
20     }
21     model.GetComponent<InteractionController>().path = result;
22 }
```

Listing 5.1: "LoadModel" method from "LoadModels" file.

The *AddComponents*, called by the *LoadModel* method, is responsible for assigning some components and performing some logic to the artefact model necessary for the platform's functionality. These are:

- *Rigidbody*: gravity set to false and all constraints frozen;
- *MeshCollider*: this component is added to all children to allow accurate collision detection;
- *InteractionController*: this component is responsible for all the user interactions with the artefact. It will be discussed in detail in Section 5.2;
- *ModelVisualizations*: this component serves as a manager of the different visualizations that the artefact can have. It will also be discussed in detail in Section 5.2;

- Resize: resize the object in case it is over the maximum size defined;

The method *ResizeObject*, which can be seen in the Code Listing 5.2, is called by *AddComponents*. This function serves the purpose of dynamically adjusting the proportions of the imported model. This adjustment is based on the *GameObject*'s bounding limits, ensuring that its dimensions do not break the predetermined upper and lower thresholds. This measure is essential to guarantee that the object comfortably fits within the spatial constraints of the environment while also avoiding being too small for eye detection. *GetBound(model)* (line 3) retrieves the bounding box of the model and *bounds.size* (line 4) extracts its dimensions. It is worth noting that particular imported objects might have negative scales. In this case, symmetrical values must be assigned to account for future calculations and manipulations. The function calculates a scaling multiplier upon determining that an object's dimensions are not within the defined range. This multiplier is then used to either amplify or diminish the object's dimensions based on the most pertinent dimension, aligning with the pre-established maximum or minimum size parameters.

```

1  public static void ObjectResize(GameObject model)
2  {
3      Bounds bounds = GetBound(model);
4      var size = bounds.size;
5      var maxSize = 100;
6      var minSize = 10;
7
8      Vector3 currentScale = model.transform.localScale;
9      Vector3 positiveScale = new Vector3(
10         Mathf.Abs(currentScale.x),
11         Mathf.Abs(currentScale.y),
12         Mathf.Abs(currentScale.z)
13     );
14     model.transform.localScale = positiveScale;
15     if (size.x > maxSize || size.y > maxSize || size.z > maxSize)
16     {
17         float maxDim = Mathf.Max(size.x, size.y, size.z);
18         float scaleMultiplier = maxSize / maxDim;
19
20         Vector3 newScale = model.transform.localScale * scaleMultiplier;
21
22         model.transform.localScale = newScale;
23     }
24
25     if (size.x < minSize || size.y < minSize || size.z < minSize)
26     {
27         float minDim = Mathf.Max(size.x, size.y, size.z);
28         float scaleMultiplier = minSize / minDim;
29
30         Vector3 newScale = model.transform.localScale * scaleMultiplier;
31
32         model.transform.localScale = newScale;

```

```
33     }  
34     model.transform.position = new Vector3(0, 0, 0);  
35 }
```

Listing 5.2: "ObjectResize" method from "LoadModels" file.

5.2 Application Interactions

It is a critical aspect of this dissertation's work to have adequate, responsive and intuitive ways to interact with the simulated environment, comprised of the imported 3D artefact models, sensors, sensor data and the heritage environment.

When users open the application, they will be greeted with an organized view of all the user controls on the left, as seen in Figure 5.1. The button on top of the control list can be clicked to toggle the menu, and each button on the menu has its own functionality. The usability of the "Open .obj" button was showcased in Section 5.1. The subsequent discussions of the "Save" and "Load" buttons will take place in Section 5.3, and the "Add Sensor" and "Open Data" in Section 5.4.



Figure 5.1: Initial view of the application.

The "Help" and "Reset" buttons are straightforward. The first one displays on screen a few lines of instructions for controlling and managing the objects, sensors and camera. If clicked again, the instructions disappear. These instructions are fundamental when dealing with new users who are not yet accustomed to the application controls. These controls will be discussed throughout this section. As the name suggests, the "Reset" button resets all application contents. This means that any objects, sensors or data that were added will be deleted, and the application will return to its initial state like the one seen in Figure 5.1. If any progress was saved, it can still be loaded at any time using the "Load" button.

5.2.1 Imported Models

First, we will explore possible ways to interact with the imported artefact's 3D models. These are:

- Selecting an object;
- Moving an object;
- Rotating an object;
- Scaling an object;
- Changing the object's visualization.

All these interactions are managed by a component called *InteractionController* that is attached to all imported game objects that represent a heritage artefact by the *AddComponents* method from the *LoadModel* script, part of which is showcased in Code Listing 5.15 of Section 5.1.

In addition to this, all objects can be deleted by pressing the "Delete" key on the keyboard after selecting it.

5.2.1.1 Selection

Selecting an object is a pre-requirement to perform any other listed actions. This imposes order and clearance on the user's workflow. The user action that defines the select feature is a double-click. A double click is detected by checking the interval on which two consecutive clicks happen through the *OnMouseDown*⁶ function from Unity's *MonoBehaviour* class. If these happen within a pre-defined threshold, it is considered a double click. Since all objects share the *InteractionController* component and only one object can be selected at a time, a static variable of the type *InteractionController* is used to represent the currently selected instance, if any. A static variable is a powerful mechanism that allows a variable to be shared among all instances of the same class [34]. By assigning the variable to itself on a double click, it will enable any object, at any time, to check if it is the one that is selected. This behaviour also helps in the implementation of some of the other interactions mentioned previously.

To distinguish a selected object from a non-selected object, a red outline is applied to the selected one, as can be seen in Figure 5.2. This outline was created using Unity's Asset Store plugin: *Quick Outline*⁷. This plugin serves as a tool to apply customizable outlines to game objects. Its shader offers a more optimal and effective way to create outlined edges in a 3D scene; it is designed to be faster and capable of supporting *Multisample anti-aliasing (MSAA)*⁸, while also eliminating the issue of gaps that can occur in world

⁶<https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseDown.html>

⁷<https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

⁸<https://www.digitaltrends.com/computing/what-is-anti-aliasing/>

space outline shaders. To use this component, we need to add its premade and publicly available script to the game object, configure the highlight's colour and thickness, and toggle it via a reference in the *InteractionController* script.



Figure 5.2: View of a selected artefact.

5.2.1.2 Movement

Moving the object is the most complex user action of the ones listed previously, especially when the different camera views that will be discussed in Section 5.2.3 are accounted for. Because of this, it was the most volatile part of the code throughout this dissertation work, having suffered several changes.

An object can only be moved if it is selected and if it is being dragged by the user. To check if the object is selected, the value of the static variable discussed previously is read, and to check if the object is being dragged, the *OnMouseDown* function from Unity's *Monobehaviour* class is used. With the requirements defined, we can move on to the object-moving logic.

The methods that comprise the core functionalities of the moving logic are *OnMouseDown*, *OnMouseDown* and *GetMouseWorldPosition*. The first two methods, as said before, are part of Unity's *Monobehaviour* class and can be seen in Code Listing 5.3 and Code Listing 5.4, respectively. *GetMouseWorldPosition* is a very important piece of code that suffered several changes throughout this time and can be seen in Code Listing 5.5.

The *OnMouseDown* method from Code Listing 5.3 starts by checking for the double click, and in case it is detected, it marks the current instance as selected. It calculates

an offset based on the current position that will be applied when updating the object's position (line 9). This offset is crucial for achieving smooth and natural dragging behaviour when moving the objects. It helps maintain the relative position between the mouse cursor and the dragged object.

```

1  private void OnMouseDown()
2  {
3      if (curtain.GetOpacity() == 0)
4      {
5          if (Time.time - lastClickTime < doubleClickTime) // Double-click
6              detected
7              {
8                  _currentSelected = this;
9                  isDragging = true;
10                 offset = transform.position - GetMouseWorldPosition();
11             }
12             else
13             {
14                 _currentSelected = null;
15             }
16             lastClickTime = Time.time;
17     }
18 }

```

Listing 5.3: "OnMouseDown" method from "InteractionController" file.

Then, the *OnMouseDrag* method from Code Listing 5.4 checks a few necessary requirements. In case these are cleared, it updates the object's position to the result of the *GetMouseWorldPosition* plus the offset calculated previously to account for the inconsistency between the mouse and object position while dragging the object.

```

1  private void OnMouseDrag()
2  {
3      if (_currentSelected == this && isDragging && !cameraController.
4          rotating && curtain.GetOpacity() == 0)
5      {
6          transform.position = GetMouseWorldPosition() + offset;
7      }
8  }

```

Listing 5.4: "OnMouseDrag" method from "InteractionController" file.

Finally, the *GetMouseWorldPosition* from Code Listing 5.5 casts a ray with a layer mask that includes only the layer named "Model". This layer mask is used to filter out objects that are not on the "Model" layer during raycasting. The ray is cast from the mouse cursor into the scene to determine the point of intersection with the object being clicked. It then retrieves the mouse cursor's current position on the screen in pixel coordinates on line 8. The intersection point is then converted from world space to camera space, its local coordinates are retrieved in relation to the camera, and the "mousePoint" z-coordinate is adjusted on line 10. This is a crucial step to ensure that the correct depth

value is used in the world position calculation. This line adjusts the z coordinate of the mouse and calculates the difference between the world position of the point where the raycast hit an object and the camera's position in world space. This is achieved using *mainCamera.transform.InverseTransformPoint()*⁹, which converts the world space point to a local space point relative to the camera. This adjustment is made to ensure that the resulting world position is correctly aligned with the camera's depth. Finally, it converts the adjusted "mousePoint" value from screen coordinates (2D) to a three-dimensional world position using the current camera's perspective. This world position can then be used to accurately interact with objects in the 3D scene.

```
1  private Vector3 GetMouseWorldPosition()
2  {
3      Camera mainCamera = Camera.main;
4      Ray mouseRay = mainCamera.ScreenPointToRay(Input.mousePosition);
5      int layerMask = 1 << LayerMask.NameToLayer("Model");
6      Physics.Raycast(mouseRay, out RaycastHit mouseHit, 800f, layerMask);
7
8      Vector3 mousePoint = Input.mousePosition;
9
10     mousePoint.z = mainCamera.transform.InverseTransformPoint(mouseHit.
11         point).z;
12
13     return mainCamera.ScreenToWorldPoint(mousePoint);
14 }
```

Listing 5.5: "GetMouseWorldPosition" method from "InteractionController" file.

At first, the movement logic was done according to an orthographic projection. For context, in an orthographic projection, objects maintain their relative sizes regardless of their distance from the camera. In contrast, perspective projection simulates how our eyes perceive objects in the real world. Objects that are closer to the camera appear larger, and as they move away, they appear smaller and converge towards a vanishing point. This convergence of lines and diminishing size of objects with distance creates a sense of depth and realism. As such, the projection had to be changed into perspective, and the code completely changed.

Later, multiple cameras with different rotations and viewing angles were implemented, and as such, the code had to suffer changes again. This was one of the most challenging aspects of the movement implementation. The biggest problem was when using non-frontal camera views - the mouse position is calculated in two dimensions while the position of the object is in three dimensions. In a frontal view, this does not pose that big of an issue because the object, even though has a three-dimensional position, only moves in the same two dimensions as the mouse. When using a non-frontal view, the object will move in two dimensions that are different from the mouse's, causing issues.

⁹<https://docs.unity3d.com/ScriptReference/Transform.InverseTransformPoint.html>

After several changes to Code Listing 5.5 and a thorough study on the matter, the movement was finally working correctly on any camera orientation.

An essential aspect, not directly related to the movement logic but crucial for achieving the desired outcome, is handling collisions. Naturally, objects need to be able to stop when colliding with other objects. To accommodate for this, *MeshColliders*¹⁰ were added to the game objects of the imported 3D artefacts. *MeshCollider* is a type of collider that adapts to a particular object's mesh so that it takes the object's shape. If it can be used, it is usually the best option because it provides the most accurate and precise collider shape. This was a relatively simple process, but it is worth noting that since many objects have children or sub-objects, it is essential to assign the mesh colliders to all the object's children.

For the object-room collisions, some options were analyzed. The objective is to place a collider on the room game object so that the objects which represent the imported 3D artefacts can be moved freely inside the room, but stop when touching the room's boundaries.

At first, a *BoxCollider*¹¹, which is a box-shaped collider, was considered, but Unity cannot properly deal with collisions that happen inside the collider. As such, we tried using *MeshColliders*, but this was not possible because, frequently, the room object had children that represented more than just one of its walls. This was a problem because applying a mask to the Raycast¹² to interact with objects inside a confined space, did not work as intended. While the ray would hit the room collider and successfully ignore it, it would not check what exists behind it. As a result, a dilemma emerged: some colliders needed to be deactivated to allow interaction with the objects inside the room. However, this solution introduced a new problem. If one of these colliders were connected to multiple walls of the room, it would enable objects to cross certain room boundaries unexpectedly.

To address this situation, the following solution was devised: using separate *BoxColliders* for each wall of the room. By deactivating the collider that was facing the interaction area, it became possible to interact with the inner objects. Whenever the camera shifted to one of its predetermined positions, the previously deactivated collider was reactivated. In contrast, the new front collider was deactivated - more information about these camera positions can be found in Section 5.2.3. This approach guaranteed the intended collision behaviour between the objects and the room.

5.2.1.3 Rotation

Rotation is a crucial interaction for managing and exploring the 3D models of the artefacts. It allows the user to position the objects in angles and positions that would not be possible otherwise.

¹⁰<https://docs.unity3d.com/ScriptReference/MeshCollider.html>

¹¹<https://docs.unity3d.com/ScriptReference/BoxCollider.html>

¹²<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>

Code Listing 5.7 showcases the rotation logic, which is part of the *Update*¹³ function - a function that runs every frame.

To rotate an object, the user must click the right button of the mouse and drag it in the desired direction. To detect the mouse right-click, we use Unity's *Input* class, which can be seen on line 1. The moment the user releases the mouse's right button, the rotation is over. We use a *bool* variable to control this behaviour. If the user presses the right button, the rotation is then applied to the game object by calculating the horizontal and vertical movement of the mouse (lines 11 and 12) and applying it on the correct rotation axis. The rotation is multiplied by the *rotationSpeed* and adjusted for frame rate using *Time.deltaTime* to ensure smooth and consistent behaviour.

One thing to notice is the use of *Space.World* as the relative space. This ensures that no matter the current rotation level of the object, the subsequent rotation operations will remain consistent, i.e. constantly rotating the object around the same axis no matter its current rotation. Directly changing the object's rotation using "*transform.rotation = transform.rotation * xRotation * yRotation*" would also have the same undesired effect.

```
1      if (Input.GetMouseButtonDown(1) && _currentSelected == this)
2      {
3          rotating = true;
4      }
5      if (Input.GetMouseButtonUp(1))
6      {
7          rotating = false;
8      }
9      if (rotating && _currentSelected == this && curtain.GetOpacity() == 0)
10     {
11         float x = Input.GetAxis("Mouse X") * rotationSpeed * Time.
12         deltaTime;
13         float y = Input.GetAxis("Mouse Y") * rotationSpeed * Time.
14         deltaTime;
15
16         transform.Rotate(0f, -x, 0f, Space.World);
17         transform.Rotate(y, 0f, 0f, Space.World);
18     }
```

Listing 5.6: Rotation logic from "Interaction Controller" file.

At first, the rotation was done using *Torque*¹⁴, but after adding collisions to the application, rotations had to be locked to avoid movement when entering in a collision. this made *Torque* no longer a viable option, and the code had to be changed, the final result being the code in Code Listing 5.7.

¹³<https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html>

¹⁴<https://docs.unity3d.com/ScriptReference/Rigidbody.AddTorque.html>

5.2.1.4 Scaling

Scaling the 3D models of the artefacts may be necessary in certain scenarios. If the imported model is not true to its real dimension, or its proportion with the rest of the environment is not appropriate, the user can scale the object as he sees fit.

To scale the object, the user needs to select it first. After selecting it, they can use the mouse wheel to rescale it - an upward turn of the mouse wheel to increase the scale and a downward one to decrease it.

We can see the scaling logic on Code Listing 5.7, which is part of the *Update* function. It starts by retrieving the amount the mouse wheel has been scrolled. It assigns this value to the variable *scrollDelta*, which can be positive or negative based on the scrolling direction. In case of this value being different than zero, meaning the user is performing a scroll action, it calculates a percentage value based on the magnitude of the current scale of the object *transform.localScale*. The purpose of the percentage value is to serve as a scaling factor that influences the amount by which the object's size changes when the user scrolls the mouse wheel. Then, a new scale vector is calculated based on the previous scale of the object, the calculated *percentageValue* and the direction and amount of scrolling the user applied. This will increase or decrease the scale of the object based on the scrolling direction.

```

1  float scrollDelta = Input.GetAxis("Mouse ScrollWheel");
2  if (scrollDelta != 0f && _currentSelected == this && curtain.GetOpacity()
   == 0)
3  {
4      float percentageValue = transform.localScale.magnitude *
       scalePercentage / 100.0f;
5
6      Vector3 scale = transform.localScale + (Vector3.one * percentageValue
       * scrollDelta);
7
8      scale.x = Mathf.Max(scale.x, 0f);
9      scale.y = Mathf.Max(scale.y, 0f);
10     scale.z = Mathf.Max(scale.z, 0f);
11
12     transform.localScale = scale;
13 }

```

Listing 5.7: Rotation logic from "Interaction Controller" file.

5.2.1.5 Changing Visualization

The final interaction available with the imported 3D models of the artefacts is changing their appearance. In the future, this could potentially serve as a means to present users with a temporal assessment of the historical evolution and visual transformations of a heritage artefact using a slider to facilitate seamless transitions on these various visualizations.

For now, it only shows the artefact's original texture and visual aspect and a wireframe view¹⁵ as is controlled by a dropdown menu. A wireframe view is a visual representation of a three-dimensional object or scene. It displays the object's structure using lines and vertices, giving a simplified outline that highlights its basic form without showing surfaces or details. In a wireframe view, only the edges and connecting lines of the object are shown, in this case, using the colour red. This was done by creating a new shader and material that fit these characteristics.

In Figure 5.3, we can see how this wireframe representation looks. By analysing the image, it can be seen that the amount of vertices and edges depends on the model structure and complexity. On the left side, we have a table object with a rather simple 3D structure, while on the right side, we have a set of tableware with a rather more complex 3D structure.

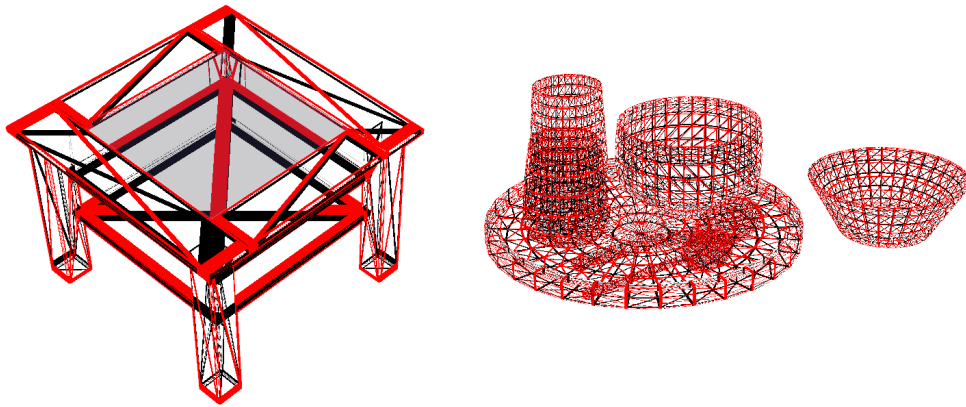


Figure 5.3: Wireframe visualization of two artefact representations.

5.2.2 Sensors

The sensor interactions are probably the simplest of the ones explored in this project. After importing a sensor, it can be selected by double-clicking on it, displaying a red outline like the imported 3D model artefacts.

Sensors can have data attached to them, as will be seen in detail in Section 5.4. After selecting a sensor, the user can click on the "Open Graph" button that will open the graph visualization, and the attached sensor data can be visualized if it has been added. An example of the graph displaying sensor data can be seen in Figure 5.4.

The default data shown is temperature, but it can be changed using the dropdown menu on the right corner of the graph visualization. When the value is changed, the dropdown menu is programmed to call the function in Code Listing 5.8. As will be discussed and reasoned in Section 5.4, there are three different graphs: one for each

¹⁵<https://help.autodesk.com/view/ACDLT/2024/ENU/?guid=GUID-84E193D7-A18D-4EE2-B978-19E4AFBCAEEC>

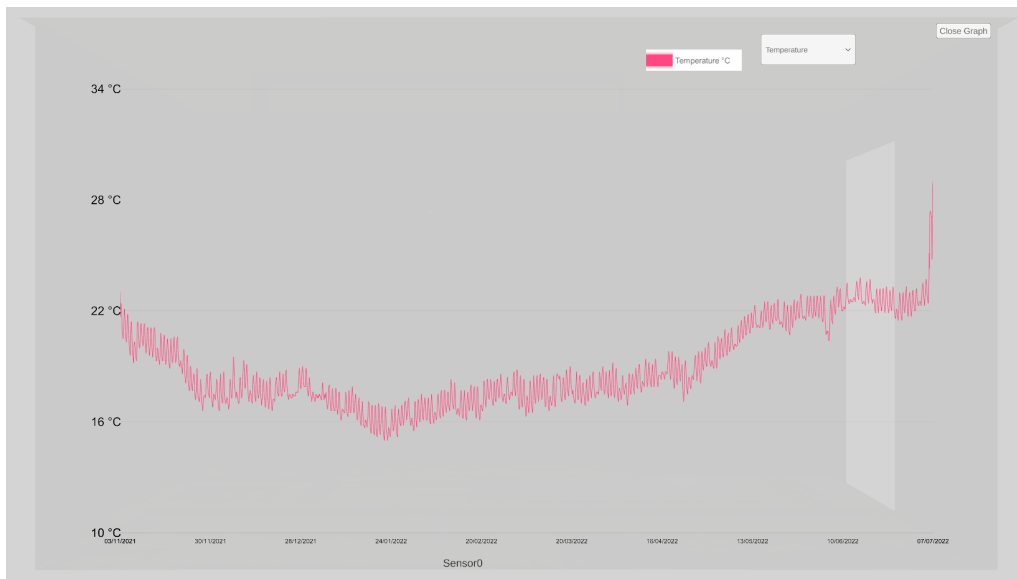


Figure 5.4: Default sensor data graph visualization.

category on the dropdown menu. This function will receive the index of the desired graph, manage the visibility of all the graph and legend components, and update the data of the selected graph.

```

1  public void ChangeVisualization(int index)
2  {
3      switch (index)
4      {
5          case 0:
6              graphH.SetActive(false);
7              legendH.SetActive(false);
8              graphHOpposite.SetActive(false);
9              graphT.SetActive(true);
10             legendT.SetActive(true);
11             graphTMaker.UpdateGraph(time, temperature, true);
12             break;
13         case 1:
14             graphT.SetActive(false);
15             legendT.SetActive(false);
16             graphHOpposite.SetActive(false);
17             graphH.SetActive(true);
18             legendH.SetActive(true);
19             graphHMaker.UpdateGraph(time, humidity, false);
20             break;
21         case 2:
22             graphH.SetActive(false);
23             graphT.SetActive(true);
24             legendT.SetActive(true);
25             legendH.SetActive(true);
26             graphHOpposite.SetActive(true);
27             graphTMaker.UpdateGraph(time, temperature, true);

```

```
28         graphH0ppositeMaker.UpdateGraph(time, humidity, false);
29         break;
30     default:
31         break;
32     }
33 }
```

Listing 5.8: "ChangeVisualization" method from "Sensor Controller" file.

5.2.3 Camera

As previously mentioned in Section 5.2.1.2, the movement logic was first done according to an orthographic projection and later changed to fit a perspective projection. Again, in an orthographic projection, objects maintain their relative sizes regardless of their distance from the camera, while perspective projection simulates how our eyes perceive objects in the real world. Objects that are closer to the camera appear larger, and as they move away, they appear smaller and converge towards a vanishing point. This convergence of lines and diminishing size of objects with distance creates a sense of depth and realism. As such, the projection had to be changed into perspective, and the code completely changed.

When referring to the camera logic, this projection change entails changing the camera's game object projection in Unity's inspector window, unlike the several and more complex changes that had to be done to adapt the movement logic.

Several camera positions were implemented with the objective of allowing the user to have different views and angles on the environment and its content. There are five pre-determined cameras with their respective positions and rotations, each placed in a different facet of an imaginary cube that surrounds the heritage environment, except for the bottom face. To change the camera position, the user needs to press the keys: "1", "2", "3", "4", and "5". The default camera position is position number one. In terms of code, this mechanic was pretty simple to implement. Some vectors for positions and rotations were set up, and when the camera position was changed, we just needed to apply those vectors to the camera's position and rotation.

The application also allows the user to freely move the camera around using the arrow keys. One thing to notice is that when doing this, the user cannot interact with the application content, only observe. This limitation had to be applied to prevent dragging objects outside the defined boundaries when looking from an angle that affects more than one collider. This issue will be discussed in Section 5.5 when discussing some of the system limitations.

As can be seen in Code Listing 5.9, the arrow keys input is detected using *Input.GetAxis* on line 1; if this value is not zero, it means the user is pressing the arrow keys, and we can then store the correspondent value (with some factoring to improve the rate of movement) and apply it to the camera.

For this particular feature, a change had to be made to the camera's game object. As can be seen on line 6, the rotation is not being applied to the camera but to a variable

called *camPivot*. *camPivot* is an empty game object created to serve, as the name suggests, as a pivot for the camera. This object is placed at the centre of the environment, and the camera becomes its child and is placed at the desired distance from it. By applying the rotation to the pivot, the camera also rotates and moves, always keeping the same distance from its pivot. This is important because if not for this, a lot of geometrical calculations would have to be done in order to achieve the circular movement of the camera around the environment. Rotating everything else but the camera was also a solution, but it would be very computationally expensive and much more complex.

```

1  if (Input.GetAxis("Vertical") != 0 || Input.GetAxis("Horizontal") != 0)
2  {
3      rotating = true;
4      headingX += Input.GetAxis("Horizontal") * Time.deltaTime * 180;
5      headingY += Input.GetAxis("Vertical") * Time.deltaTime * 180;
6      camPivot.rotation = Quaternion.Euler(headingY, -headingX, 0);
7  }

```

Listing 5.9: Free camera movement logic from "Camera Controller" file.

5.3 Data Management

The next building block of the platform is the ability to Save and Load the session's progress. This is very important because it allows for a consistent, efficient and non-data loss work environment.

Opposite to the 3D Model Importer feature described in Section 5.1, Unity Engine offers native ways to solve part of this challenge.

With this project's context and goals, both the objects and sensors need to be saved to restore the progress on load. Not only does the object itself need to be saved, but also its associated data like position, rotation, scale, and in the sensor's case, the imported sensor data if it exists. This particular feature can be divided into four fundamental phases:

- Exporting the 3D models into an OBJ file;
- Saving the previously described associated model data in a BIN file;
- Importing the saved OBJ file as game objects;
- Read the data from the BIN file and apply it to the imported game objects.

It was decided to save the related data in BIN format for privacy and security reasons since BIN data is not human-readable.

For context, a BIN file is a type of data file that solely or mainly holds information in binary format. Several apps utilize the .bin extension for a variety of types of files. Data backups, disk images, resource libraries, and firmware updates are a few examples. Unlike text-based files, BIN files can be modified with a text editor and are human-readable.

Binary data cannot usually be changed and is not human-readable. While the majority of BIN files exclusively hold data in binary format, some BIN files can include text data. For instance, a header containing a plain text string describing the game version and when the game last edited the file may be present in a BIN file that stores saved game data in binary format.

All the relevant data is stored within a persistent data directory. Unity's *Application.persistentDataPath* returns the file path to a directory where the application can store data that should persist across runs of the application. This directory can be used to store user preferences, game save data, and other data that needs to be retained even if the application is closed. The path returned by *Application.persistentDataPath* can be read and written into at runtime, allowing the application to save and load data to and from the directory. This data can be kept in a location apart from the game data so that a player will not lose it if they uninstall or update the application.

The actual location where the data is stored differs based on the operative system used, and the list can be observed in the "Application.persistentDataPath" section of Unity's website¹⁶.

The *System Manager* game object handles all button presses and part of the subsequent logic, including the Save and Load buttons.

The *SaveSystem* class is called by *System Manager* to handle the rest of the logic, and it has four methods that need to be discussed:

- *SaveModels*
- *SaveSensors*
- *LoadModels*
- *LoadSensors*

As the names suggest, these methods are responsible for the saving and loading of models and sensors. The main difference between the model and sensor methods is the data class they use. As such, the data classes will be explored in addition to the model's save and load methods in the following sections.

5.3.1 Save

This section of the project suffered a big and important change late in the development. At first, to save the game objects' 3D models and texture files, these were generated from the information present in Unity. This is a complex and inconsistent project, and it was producing a bug: the textures were loaded but not in the correct order. It was derived that the problem relied on how the textures were being exported on the save action. We believed the root cause to be the uncontrollable incorrect order in which the textures were

¹⁶<https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>

exported, which resulted in misassigned textures on the load. Another solution was found and will be discussed in the next paragraphs.

As previously said, the Save functionality starts in the *System Manager* game object. Its *Save* function can be seen in Code Listing 5.10.

The first step is fetching all game objects that contain one of the designated tags that represent the models and the sensors. In case this is the first time the object is being saved, the original model and texture files will be copied to the *Application.persistentDataPath* directory from its source directory (this directory was saved when the model was imported - Section 5.1), renamed and modified to account for this directory change and the name changes. This procedure, which can be seen in lines 22-72, exists to avoid generating OBJ and MTL files from the Unity game objects, which is more complex and inconsistent. The mentioned file modifications are crucial for this procedure because the directories and name mapping for the object-texture pairs are present in the MTL file. The method *ModifyFileContent* is called on line 69 and is responsible for this - to update the file names and mapping paths on the MTL file.

In case of any object being outlined (selected), the outline is temporarily removed to avoid problems in texture saving.

After this, the *SaveModels* and *SaveSensors* methods from the *SaveSystem* class are invoked.

```

1  public void Save()
2  {
3      GameObject[] models = GameObject.FindGameObjectsWithTag(modelTag);
4      GameObject[] sensors = GameObject.FindGameObjectsWithTag(sensorTag);
5
6      if (models.Length > 0)
7      {
8          GameObject o = models[0];
9          bool selected = false;
10         string destinationDirectory = Application.persistentDataPath;
11         foreach (GameObject obj in models)
12         {
13             Outline outline = obj.GetComponent<Outline>();
14             if (outline != null && outline.enabled)
15             {
16                 outline.enabled = false;
17                 o = obj;
18                 selected = true;
19             }
20             obj.GetComponent<ModelVisualizations>().SetVisualizationMode
21             (0);
22
23             if (obj.GetComponent<InteractionController>().firstLoad)
24             {
25                 string sourceDirectory = obj.GetComponent<
26                 InteractionController>().path;

```

```
25
26     string[] files = Directory.GetFiles(sourceDirectory);
27     foreach (string filePath in files)
28     {
29         bool found = false;
30         string fileName = Path.GetFileName(filePath);
31         string fileExtension = Path.GetExtension(fileName);
32         string newFileName = "";
33         if (fileExtension == ".mtl" || fileExtension == ".MTL"
34     )
35         {
36             newFileName = obj.name + ".mtl";
37             found = true;
38         }
39         if (fileExtension == ".obj" || fileExtension == ".OBJ"
40     )
41         {
42             newFileName = obj.name + ".obj";
43             found = true;
44         }
45         if (found)
46         {
47             string destinationPath = Path.Combine(
48 destinationDirectory, newFileName);
49             File.Copy(filePath, destinationPath, true);
50             found = false;
51         }
52     }
53     string extensionToSearch = ".mtl";
54     string[] files2 = Directory.GetFiles(destinationDirectory,
55 obj.name + extensionToSearch);
56     string mtlPath = "";
57     foreach (string filePath in files2)
58     {
59         mtlPath = filePath;
60     }
61     int textureCounter = 0;
62     foreach (string filePath in files)
63     {
64         string fileName = Path.GetFileName(filePath);
65         string filenameNoExtension = Path.
66 GetFileNameWithoutExtension(filePath);
67         string fileExtension = Path.GetExtension(fileName);
68         string newFileName = "";
69         if (fileExtension == ".png" || fileExtension == ".PNG"
70 || fileExtension == ".JPG" || fileExtension == ".JPEG" || fileExtension ==
71 ".jpg" || fileExtension == ".jpeg")
72         {
73             newFileName = obj.name + "_tex" + textureCounter++
74 + fileExtension;
```

```

67         string destinationPath = Path.Combine(
destinationDirectory, newFileName);
68         File.Copy(filePath, destinationPath, true);
69         ModifyFileContent(mtlPath, fileName,
destinationDirectory, newFileName);
70     }
71 }
72 }
73 }
74 SaveSystem.SaveModels();
75 Debug.Log("Saved Models.");
76 if (selected)
77 {
78     o.GetComponent<Outline>().enabled = true;
79 }
80 }
81 else
82 {
83     SaveSystem.ClearModels();
84 }
85
86 if (sensors.Length > 0)
87 {
88     GameObject s = sensors[0];
89     bool selected = false;
90     foreach (GameObject obj in sensors)
91     {
92         Outline outline = obj.GetComponent<Outline>();
93         if (outline != null && outline.enabled)
94         {
95             outline.enabled = false;
96             s = obj;
97             selected = true;
98         }
99     }
100     SaveSystem.SaveSensors();
101     Debug.Log("Saved Sensors.");
102     if (selected)
103     {
104         s.GetComponent<Outline>().enabled = true;
105     }
106 }
107 else
108 {
109     SaveSystem.ClearSensors();
110 }
111 saveCompleteText.SetActive(true);
112 StartCoroutine(Waiter(2, () =>
113 {
114     saveCompleteText.SetActive(false);

```

```
115     });  
116 }
```

Listing 5.10: "Save" method from "SystemManager" file.

The goal of the *SaveModels* method, which can be seen in Code Listing 5.11, is to clear all previously saved files and create new ones.

It starts by collecting all the names of objects on the scene. It then iterates through the files in the "save" directory (*Application.persistentDataPath*), looking for files which do not belong to the current objects on the scene; these are files from previous saves, which are deleted. This is particularly important due to the potential scenario where an object exists in a previous save file and is removed from the scene before the next save operation. In such cases, residual data could inadvertently be loaded in future loads without the complete deletion of its corresponding file, leading to inconsistencies.

Following the deletion of previous save files, a File Stream is created on the Unity's *Application.persistentDataPath* with the *FileMode* on "Create" to create a new file and a Binary Formatter¹⁷ to allow serialization and deserialization of data. For context, serialization is the mechanism through which an object's state is transformed into a coherent byte stream. The complementary process, deserialization, involves recreating the original object in memory from the corresponding byte stream. This symbiotic relationship between serialization and deserialization gives objects persistence, ensuring their sustained existence across different instances and sessions. Scenes, assets, and asset bundles are loaded and saved to and from the device's memory via serialization in Unity.

```
1     public static void SaveModels()  
2     {  
3         string directoryPath = Application.persistentDataPath;  
4         GameObject[] models = GameObject.FindGameObjectsWithTag(modelTag);  
5         List<string> names = new List<string>();  
6         foreach (GameObject model in models)  
7         {  
8             string modelName = model.name;  
9             names.Add(modelName);  
10        }  
11        string[] files = Directory.GetFiles(directoryPath);  
12        for (int i = 0; i < files.Length; i++)  
13        {  
14            files[i] = files[i].Replace("/", "\\");  
15        }  
16        List<string> filesToDelete = new List<string>();  
17        foreach (string filePath in files)  
18        {  
19            bool found = false;  
20            foreach (string modelName in names)  
21            {
```

¹⁷<https://learn.microsoft.com/en-us/dotnet/api/system.runtime.serialization.formatters.binary.binaryformatter?view=net-7.0>

```

22         if (filePath.Contains(modelName))
23         {
24             found = true;
25             break;
26         }
27     }
28     if (!found)
29     {
30         filesToDelete.Add(filePath);
31     }
32 }
33 foreach (string file in filesToDelete)
34 {
35     if (!file.Contains(".log"))
36     {
37         File.Delete(file);
38     }
39 }
40 BinaryFormatter formatter = new BinaryFormatter();
41 string path = Application.persistentDataPath + "/modelsInfo.bin";
42 FileStream stream = new FileStream(path, FileMode.Create);
43 ModelData data = new ModelData();
44 formatter.Serialize(stream, data);
45 stream.Close();
46 }

```

Listing 5.11: "SaveModels" method from "SaveSystem" file.

The data that the Binary Formatter will serialize is retrieved from a class called *ModelData*, showcased in Code Listing 5.12.

There are some general and specific requisites and limitations while serializing and deserializing data in Unity that need to be followed¹⁸. The most important rule is that only primitive types are allowed to be used. This means all data not originally in the form of a primitive type needs to be converted to it before being serialized. In this specific case, since we will be serializing the *ModelData* class, it needs to have the *Serializable* identifier and not be static; this can be seen in line 1 of Code Listing 5.12.

The *ModelData* class is responsible for creating and holding all the relevant related model data, like position, scale and rotation values; it also calls a method from another class to handle the export of the game object into an OBJ file.

The *ModelData* class starts by retrieving all game objects present on the scene that have the designated identifying tag; these are the objects that hold the relevant associated data.

```

1  [System.Serializable]
2  public class ModelData
3  {
4      public float[,] position;
5      public float[,] rotation;

```

¹⁸<https://docs.unity3d.com/Manual/script-Serialization.html>

```
6      public float[,] scale;
7
8      private string tagName = "Model";
9
10     public ModelData()
11     {
12         GameObject[] allObjects = GameObject.FindGameObjectsWithTag(
13         tagName);
14         allObjects = allObjects.OrderBy(obj => obj.name).ToArray();
15         position = new float[allObjects.Length, 3];
16         rotation = new float[allObjects.Length, 3];
17         scale = new float[allObjects.Length, 3];
18         int counter = 0;
19         foreach (GameObject obj in allObjects)
20         {
21             string path = Application.persistentDataPath + "/savedModel" +
22             counter + ".obj";
23             ObjExporter.ExportToObj(obj, path);
24
25             Transform transform = obj.transform;
26             position[counter, 0] = transform.position.x;
27             position[counter, 1] = transform.position.y;
28             position[counter, 2] = transform.position.z;
29
30             rotation[counter, 0] = transform.rotation.eulerAngles.x;
31             rotation[counter, 1] = transform.rotation.eulerAngles.y;
32             rotation[counter, 2] = transform.rotation.eulerAngles.z;
33
34             scale[counter, 0] = transform.localScale.x;
35             scale[counter, 1] = transform.localScale.y;
36             scale[counter, 2] = transform.localScale.z;
37             counter++;
38         }
39     }
40 }
```

Listing 5.12: "ModelData" class.

The *ModelData* class calls the method *ExportToOBJ* from the *ObjExporter* class, which is responsible for exporting the specified game object into an OBJ file which will be saved on Unity's `Application.persistentDataPath`. The method is very long, so it will not be displayed in a Code Listing.

The Wavefront OBJ file format is a standard for presenting 3D models and is used in conjunction with Material (MTL) files.

The *ExportToObj* function, which is at the centre of the code, requires two crucial inputs: the *GameObject* to be exported and the selected export directory (path). The procedure is divided into several distinct steps.

First, the code locates each *MeshFilter* component in the *GameObject* hierarchy. These elements keep the 3D model's basic geometry in storage. Two lists are started concurrently

to track material names and texture routes for the MTL file. The code uses `StringBuilder`¹⁹ instances for effective content generation. Repeatedly changing a string can result in a severe performance hit for algorithms that undertake heavy string manipulation, hence using `StringBuilder`, a mutable string class. These builders gather the information needed for later export.

The core of the export process is geometry and material processing. The algorithm collects crucial details, including vertex locations, UV coordinates, normals, and triangle indices for each `MeshFilter` in the `GameObject`. After being appropriately formatted, these elements are added to the object's `StringBuilder`.

Additionally taken into account are the materials and textures linked to each `MeshFilter`. The primary texture, as well as material characteristics like ambient, diffuse, and specular colours, are extracted by the code. Each material is given a distinct name, and the corresponding texture is saved as a PNG picture. The material `StringBuilder` is updated with all material-related data.

The data collected for the OBJ and MTL files are written into separate files after all `MeshFilter` components have been processed.

The `SensorData` class follows the same logic as the `ModelData`, but instead of holding position, scale and rotation values, it has the position and the values originating from the temperature and humidity sensors. As mentioned previously, the data structures that have this data need to be based on primitive types; in the model saving, there is no obstacle in that regard, but while saving the sensor data, there is data from the `DateTime` type that needs to be converted into a string, hence the extra loop on line 36 of Code Listing 5.13. In this class, three matrixes were used to store all sensors' temperature, humidity and time values. Each row represents a sensor, while the values of the designated categories are represented in their attached columns.

The data matrix is initially initialized with the maximum length of all sensors and with placeholder values. These placeholders will be immediately replaced with the actual values because different sensors may have varied data lengths, leading to unreal data. This guarantees that in sensors with fewer data, values that were never replaced will not be misinterpreted as real values and will be treated appropriately afterwards. This process can be seen in Code Listing 5.13.

```

1  public SensorData()
2  {
3      GameObject[] allObjects = GameObject.FindGameObjectsWithTag(tagName);
4      allObjects = allObjects.OrderBy(obj => obj.name).ToArray();
5      position = new float[allObjects.Length, 3];
6
7      int counter = 0;
8      int maxSize = 0;
9      foreach (GameObject obj in allObjects)
10     {

```

¹⁹<https://learn.microsoft.com/en-us/dotnet/api/system.text.stringbuilder?view=net-7.0>

```
11         SensorController sensorController = obj.GetComponent<  
SensorController>();  
12         List<float> temp = sensorController.GetTemperature();  
13         if (temp.Count > maxSize)  
14         {  
15             maxSize = temp.Count;  
16         }  
17     }  
18     temperature = new float[allObjects.Length, maxSize];  
19     humidity = new float[allObjects.Length, maxSize];  
20     time = new string[allObjects.Length, maxSize];  
21     temperature = InitializeMatrix(temperature);  
22     humidity = InitializeMatrix(humidity);  
23     time = InitializeMatrix(time);  
24     foreach (GameObject obj in allObjects)  
25     {  
26         SensorController sensorController = obj.GetComponent<  
SensorController>();  
27         Transform transform = obj.transform;  
28         position[counter, 0] = transform.position.x;  
29         position[counter, 1] = transform.position.y;  
30         position[counter, 2] = transform.position.z;  
31         float[] tempTemp = sensorController.GetTemperature().ToArray();  
32         float[] tempHum = sensorController.GetHumidity().ToArray();  
33         DateTime[] tempTime = sensorController.GetTime().ToArray();  
34         string[] finalTempTime = new string[tempTemp.Length];  
35         int timeCounter = 0;  
36         foreach (DateTime temp in tempTime)  
37         {  
38             finalTempTime[timeCounter++] = temp.ToString();  
39         }  
40         int localCounter = 0;  
41         foreach (float temp in tempTemp)  
42         {  
43             temperature[counter, localCounter] = temp;  
44             humidity[counter, localCounter] = tempHum[localCounter];  
45             time[counter, localCounter] = finalTempTime[localCounter];  
46             localCounter++;  
47         }  
48         counter++;  
49     }  
50 }
```

Listing 5.13: "SensorData" class.

5.3.2 Load

As previously said, the Load functionality starts in the *System Manager* game object. Its *Load* function can be seen in Code Listing 5.14 and is responsible for loading the models

and sensors from the saved files and assigning the associated data to them.

The first step is to fetch all game objects containing one of the designated tags that represent the models and the sensors and delete them from the scene. This is needed so that we can load the content of the saved files. One thing to note is that since the sensor's game objects are instantiations of prefabs²⁰, which is a fully configured reusable game object, and the prefab object also has the same tag, there is the need to check if the game object is present on the scene before deleting it to avoid deleting the prefab itself. This logic is demonstrated in lines 4-10 of Code Listing 5.14.

The next part is repeated for models and sensors, and, as such, only the model part will be detailed in this section. The method *LoadModels* from the *SaveSystem* class is then called, and in case the associated model data returned not being null, the models' data assignments take place.

```

1  public void Load()
2  {
3      GameObject[] gameObjects = Resources.FindObjectsOfTypeAll<GameObject>();
4      foreach (GameObject obj in gameObjects)
5      {
6          if (IsInScene(obj) && (obj.CompareTag(modelTag) || obj.CompareTag(modelAddOnsTag) || obj.CompareTag(sensorTag)))
7          {
8              GameObject.DestroyImmediate(obj);
9          }
10     }
11     objectInfoText.SetActive(false);
12     ModelData data = SaveSystem.LoadModel();
13     if (data != null)
14     {
15         gameObjects = GameObject.FindGameObjectsWithTag(modelTag);
16         gameObjects = RemoveDuplicates(gameObjects);
17         int counter = 0;
18         foreach (GameObject obj in gameObjects)
19         {
20             Transform transform = obj.transform;
21             transform.position = new Vector3(data.position[counter, 0],
22 data.position[counter, 1], data.position[counter, 2]);
23             transform.localScale = new Vector3(data.scale[counter, 0],
24 data.scale[counter, 1], data.scale[counter, 2]);
25             transform.rotation = Quaternion.Euler(data.rotation[counter,
26 0], data.rotation[counter, 1], data.rotation[counter, 2]);
27             counter++;
28         }
29         Debug.Log("Loaded Models.");
30     }
31     sensorInfoText.SetActive(false);

```

²⁰<https://docs.unity3d.com/Manual/Prefabs.html>

```

30     openGraphButton.SetActive(false);
31     SensorData sensorData = SaveSystem.LoadSensors();
32     if (sensorData != null)
33     {
34         sensorCounter = 0;
35         for (int i = 0; i < sensorData.position.GetLength(0); i++)
36         {
37             GameObject sensor = Instantiate(sensorPrefab, new Vector3(
38                 sensorData.position[i, 0], sensorData.position[i, 1], sensorData.position[i
39                 , 2]), Quaternion.identity);
40             SensorController s = sensor.AddComponent<SensorController>();
41             List<float> tempT = new List<float>();
42             List<float> tempH = new List<float>();
43             List<DateTime> tempTime = new List<DateTime>();
44             for (int j = 0; j < sensorData.temperature.GetLength(1); j++)
45             {
46                 tempT.Add(sensorData.temperature[i, j]);
47                 tempH.Add(sensorData.humidity[i, j]);
48                 DateTime dateTime;
49                 if (DateTime.TryParseExact(sensorData.time[i, j], "dd/MM/
50                 yyyy HH:mm:ss", CultureInfo.InvariantCulture, DateTimeStyles.None, out
51                 dateTime))
52                 {
53                     tempTime.Add(dateTime);
54                 }
55                 s.AddValues(tempTime, tempT, tempH);
56                 sensor.name = "Sensor" + sensorCounter++;
57             }
58             Debug.Log("Loaded Sensors.");
59         }
60     }

```

Listing 5.14: "Load" method from "SystemManager" file.

The method *LoadModel* from the *SaveSystem* class can be seen in Code Listing 5.15. This method is responsible for loading the saved OBJ files into the scene and returning their associated data.

The method checks for the designated files in Unity's *Application.persistentDataPath*, and in case it exists, it creates a *BinaryFormatter* to deserialize the associated model data that was serialized into binary. It then checks for the obj files, and while these exist, they are loaded into the scene using the plugin mentioned in Section 4.2, *OBJLoader*^{21,22}. Finally, it returns the data that was deserialized so that it can be applied to the loaded game objects.

```

1     public static ModelData LoadModel()
2     {

```

²¹<https://github.com/PhalanxHead/UnityRuntimeOBJLoaderDocs>

²²<https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547>

```

3      int counter = 0;
4      string objpath = Application.persistentDataPath + "/WavefrontObject" +
counter + ".obj";
5      string mtlpath = Application.persistentDataPath + "/WavefrontObject" +
counter + ".mtl";
6      string path = Application.persistentDataPath + "/modelsInfo.bin";
7      if (File.Exists(path))
8      {
9          BinaryFormatter formatter = new BinaryFormatter();
10         FileStream stream = new FileStream(path, FileMode.Open);
11         ModelData data = formatter.Deserialize(stream) as ModelData;
12         stream.Close();
13         while (File.Exists(objpath))
14         {
15             if (File.Exists(mtlpath))
16             {
17                 LoadModels.LoadModel(objpath, mtlpath, counter);
18             }
19             else
20             {
21                 LoadModels.LoadModel(objpath, counter);
22             }
23
24             counter++;
25             objpath = Application.persistentDataPath + "/WavefrontObject"
+ counter + ".obj";
26             mtlpath = Application.persistentDataPath + "/WavefrontObject"
+ counter + ".mtl";
27         }
28         return data;
29     }
30     else
31     {
32         Debug.Log("Save file not found in " + path);
33         return null;
34     }
35 }

```

Listing 5.15: "LoadModel" method from "SaveSystem" file.

5.4 Sensor Data

An essential part of this dissertation solution's implementation is related to the sensor data. Data is crucial when monitoring and managing heritage artefacts in a cultural environment. It is also a critical piece of a Digital Twin environment, as studied in Chapter 3 and Chapter 4.

Ideally, the Sensor Data would come from a real-time feed directly connected to the sensor in the real environment, but this part was moved into future works due to practical

and time constraints.

As defined in Chapter 4, the platform allows the user to add data files related to them and visualize them. Both of these functionalities will be explored in detail in the following sections.

Before the user can import data, sensors need to exist. For this, the user can also add or place sensors on the scene that mirror the actual heritage environment. To add a sensor, the user must click the "Add Sensor" button and click where he would like to place it. If the place the user clicked is part of the enclosing environment, a grey sphere sensor prefab that was constructed previously will be instantiated in that position, with a small positional offset to avoid it being placed through surfaces. *SensorController* is the only component that needs to be added to the prefab since it handles all the logic related to the sensor and the data it may eventually hold.

5.4.1 Importing Data

Both of these functionalities start with pressing the respective button on the application's menu. The *AddData* button opens the respective Operative System explorer windows and allows the user to select one CSV (Character Separated Values)²³ file that should represent the sensor data. At the moment, the platform is ready to accept data containing temperature, humidity and time values with the structure that can be seen in Figure 5.5.

	A	B	C	D	E	F	G
1	[##]Device Description						
2	Data 5						
3							
4	[##]Device Information						
5	Version = V2.0-0						
6	Serial Number = A16030206						
7	File Name = n5_Registo-HR-T_CineJunior_VitrineExposicao.LOG						
8	Log Interval = 02:00:00						
9							
10	[##]						
11	Humidity = %rh						
12	Low Alarm Threshold = 20						
13	High Alarm Threshold = 80						
14							
15	[##]						
16	Temperature = °C						
17	Low Alarm Threshold = 10						
18	High Alarm Threshold = 50						
19							
20	[##]						
21	Date	Time	Humidity	Temperature			
22							
23	[##]						
24	[##]Measurements						
25	#####	08:00:00	58.79	23.4			
26	#####	10:00:00	57.88	22.1			
27	#####	12:00:00	55.98	22.6			
28	#####	14:00:00	52.3	22.8			
29	#####	16:00:00	51.24	22.7			
30	#####	18:00:00	50.59	22.4			
31	#####	20:00:00	50.76	21.9			
32	#####	22:00:00	50.78	21.5			
33	#####	00:00:00	50.81	21.2			
34	#####	02:00:00	50.8	21.1			
35	#####	04:00:00	50.98	20.9			
36	#####	06:00:00	51.93	20.9			

Figure 5.5: Sensor data file provided by DCR.

²³<https://www.howtogeek.com/348960/what-is-a-csv-file-and-how-do-i-open-it/>

This structure accounts for a header that provides information about the file and device. The files used to create and test the code were provided by teachers from DCR and are files generated from Data Loggers²⁴, a piece of technology that combines a sensor for information collection and a memory chip for information storage. This is the device that is most frequently used in these cultural environments to capture data.

After the user selects the CSV file to be added, a method from the class *CSVParser* is called by *SystemManager* to handle the parsing of the CSV file and save the data encoded in it. This method, which can be seen in Code Listing 5.16, starts by creating a *StreamReader* to receive the file stream and access the file contents. It then goes through the file, ignoring lines that do not follow the defined structure (meaning they are part of the header or have missing values) and parsing and adding the values of the lines that do follow the requirements to local data structures. One thing to notice is that if a value is not parsed successfully to its type, the row is skipped to avoid invalid data.

```

1  private bool ParseCSV()
2  {
3      StreamReader reader;
4
5      if (csvFileStream != null)
6      {
7          reader = new StreamReader(csvFileStream);
8      }
9      else
10     {
11         Debug.LogError("No CSV file MemoryStream provided.");
12         return false;
13     }
14     Clear();
15     while (!reader.EndOfStream)
16     {
17         string line = reader.ReadLine();
18         string[] values = line.Split(',');
19
20         // Check if the row has the required number of columns
21         if (values.Length < 4)
22         {
23             //Debug.LogWarning("Skipping row with missing values: " + line
24         );
25             continue;
26         }
27
28         // Parse the date and time columns
29         string dateTimeString = values[0] + " " + values[1];
30
31         DateTime dateTime;

```

²⁴https://www.geminidataloggers.com/info/what_is_a_data_logger

```
31         if (!DateTime.TryParseExact(dateTimeString, "dd/MM/yyyy HH:mm:ss",  
CultureInfo.InvariantCulture, DateTimeStyles.None, out dateTime))  
32         {  
33             //Debug.LogWarning("Skipping row with invalid date and time: "  
+ line);  
34             continue;  
35         }  
36  
37         // Parse the temperature and humidity columns  
38         float humidity;  
39         if (!float.TryParse(values[2], out humidity))  
40         {  
41             //Debug.LogWarning("Skipping row with invalid humidity: " +  
line);  
42             continue;  
43         }  
44  
45         float temperature;  
46         if (!float.TryParse(values[3], out temperature))  
47         {  
48             //Debug.LogWarning("Skipping row with invalid temperature: " +  
line);  
49             continue;  
50         }  
51  
52         // Store the values in respective lists  
53         timeValues.Add(dateTime);  
54         temperatureValues.Add(temperature);  
55         humidityValues.Add(humidity);  
56  
57         //Debug.Log("Parsed values: Date: " + dateTime.ToString("dd/MM/  
yyyy") + ", Time: " + dateTime.ToString("HH:mm:ss") + ", Temperature: " +  
temperature + ", Humidity: " + humidity);  
58     }  
59     reader.Close();  
60     return true;  
61 }
```

Listing 5.16: "ParseCSV" method from "CSVParser" file.

The values saved in the local data structures are then fetched by the System Manager using the usual "Get" methods. These data will be held on until the user clicks on a sensor present on the scene to which he wishes to add the data. When this happens, and in case the clicked object is, in fact, a sensor, the values are added to its *SensorController* component. As the name suggests, this component is responsible for handling all operations and data related to the sensor.

One important detail that, at first, caused every data import to overwrite previous ones is that Lists²⁵ are reference types in C# [43]. This means that instead of actually

²⁵<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=net-7.0>

containing data like its counterpart value types, reference types store references to the data in memory. The method seen in Code Listing 5.16 writes data to its local list data structures. The values in these lists were previously being cleared on new data arrivals using the List method *Clear*²⁶ as can be seen in Code Listing 5.17.

```

1  public void Clear()
2  {
3      timeValues.Clear();
4      temperatureValues.Clear();
5      humidityValues.Clear();
6  }
```

Listing 5.17: Old "Clear" method from "CSVParser" file.

The problem is that this sets the list counter to zero, but the list's internal array and capacity remain the same. As such, when new items arrive, the position in memory is the same and all the previous data content changes to the new one. The solution used, which is very simple, can be seen in Code Listing 5.18. This solution prevents the issue since it creates new lists with newly allocated memory space.

```

1  public void Clear()
2  {
3      temperatureValues = new List<float>();
4      humidityValues = new List<float>();
5      timeValues = new List<DateTime>();
6  }
```

Listing 5.18: New "Clear" method from "CSVParser" file.

5.4.2 Visualizing Data

After the data is imported and assigned to one of the sensors present on the scene, it is essential that it can be visualised in a consistent and informative manner.

For this project, as referenced before in Chapter 4, it was decided to use one of Unity's Asset Store plugins, Graph and Chart²⁷. This is a very powerful plugin that is extensively documented and provides plenty of pre-built and highly customizable 2D and 3D charts like Graph, Bar, Pie, Torus, and many more. It also offers several interaction modes and events and integration with several popular assets that are used in this project, like *TextMeshPro*²⁸.

In this project, the Graph Chart was used to represent all the relevant data: temperature, humidity and both at the same time (always in function of the time). Figure 5.6 and 5.7 show the end result of these graphs with one of the data files provided.

To build these graphs, each individual category had to be created in the inspector, only one for the graphs shown in Figure 5.6 and two for the graph shown in Figure 5.7. In

²⁶<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1.clear?view=net-7.0#system-collections-generic-list-1-clear>

²⁷<https://assetstore.unity.com/packages/tools/gui/graph-and-chart-78488>

²⁸<https://docs.unity3d.com/Manual/com.unity.textmeshpro.html>



Figure 5.6: (a) Temperature and (b) Humidity graph using sensor data.

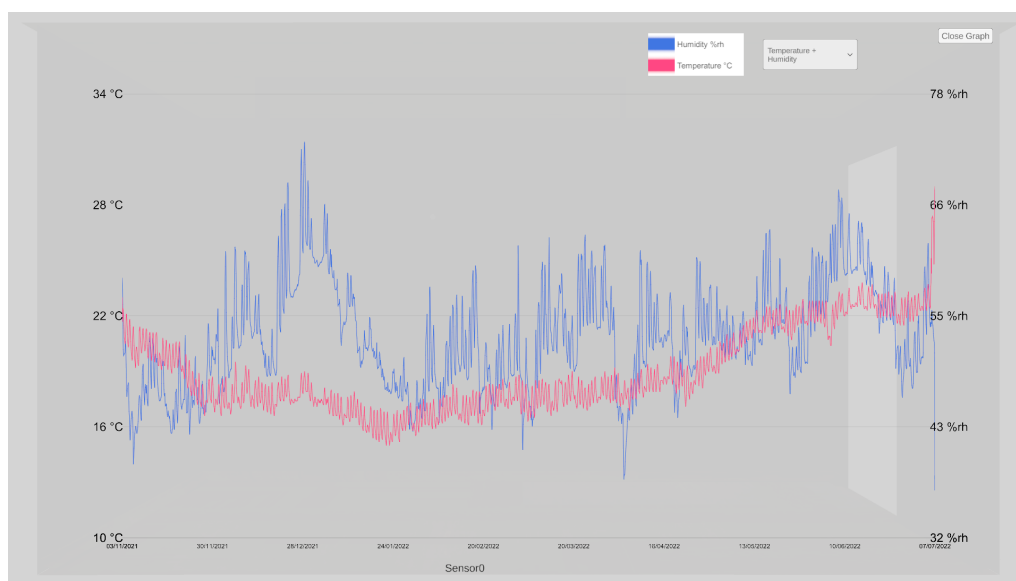


Figure 5.7: Temperature and humidity merged graph using sensor data.

addition, all the visual aspects, like the colours and sizes, were also set in the inspector window. Each graph label had to be placed and sized individually, but its colour linkage to the categories is achieved automatically by the Graph and Chart plugin's functionality. All these aspects make the creation process simpler and more efficient.

Since each graph needs prior settings to be tuned differently (like categories and axis), there need to be three separate graphs, even though only one is active at all times, instead of one graph that changes according to the data that is provided.

One thing to note is the merged graph from Figure 5.7. This graph is, in fact, the composition of two separate graphs - one with a left-aligned vertical axis and horizontal axis and another with only a right-aligned axis. This is needed to accommodate the need to visualize both temperature and humidity values together but is not supported by Unity's Asset Store plugin, Graph and Chart. As such, each of the graph counterparts receives its identified data, temperature or humidity, while only one of them receives the time values. This is not a problem since both data originate from the same sensor, the same CSV file and the same time and date values.

A script was created that receives data and populates the graph attached to the game objects. This method is called from the *SensorController* class when needed - when the graph is toggled on, or the graph visualization is changed via the dropdown menu that can be seen in Figures 5.6, 5.7, 5.6. This script has two implementations of the method *UpdateGraph*, one for the merged temperature and humidity graph that can be seen in Figure 5.7 and one for the other two. For redundancy reasons, only one of the implementations will be detailed.

The method implementation, which refers to the non-merged graph, can be seen in Code Listing 5.19 and starts by starting a new batch in line 5. Starting and ending the batch at the beginning and end of any graph content-related operations is standard for any update operation using any graph from the Graph and Chart plugin. The code then determines if this operation refers to the temperature or humidity graph by checking the provided *bool* argument (*isTemperature*) and clears the specified category accordingly. After this, the values are iterated and checked for its validity. If the data is considered valid, it is then added to the graph, as depicted in lines 23 and 27. Finally, we end the data batch using Graph and Chart's designated method invocation in line 38 - *EndBatch*. In case the sensor has no data attached to it, the message in line 42 is printed.

```

1  public void UpdateGraph(List<DateTime> timeValues, List<float> values, bool
    isTemperature)
2  {
3      if (graph != null)
4      {
5          graph.DataSource.StartBatch(); // start a new update batch
6          if (isTemperature) // clear the categories we have created in the
inspector
7          {
8              graph.DataSource.ClearCategory("Temperature");
9          }
10         else
11         {
12             graph.DataSource.ClearCategory("Humidity");
13         }
14         if (timeValues.Count != 0 && values.Count != 0)
15         {
16             int counter = 0;
17             foreach (float point in values)
18             {
19                 if (point != -1000.0f && timeValues[counter].ToString() !=
"00/00/0000 00:00:00")
20                 {
21                     if (isTemperature)
22                     {
23                         graph.DataSource.AddPointToCategory("Temperature",
timeValues[counter], point);
24                     }
25                     else

```

```
26         {
27             graph.DataSource.AddPointToCategory("Humidity",
timeValues[counter], point);
28         }
29     }
30     counter++;
31 }
32 }
33 else
34 {
35     Debug.Log("Couldn't get data. Data is empty.");
36 }
37 graph.DataSource.EndBatch();
38 }
39 else
40 {
41     Debug.LogWarning("Graph is null");
42 }
43 }
```

Listing 5.19: "UpdateGrpah" method from "GraphMaker" file used to update data values on the active graph.

With this implementation, no matter which sensor is selected, we are sure that the data that is being visualized is, in fact, the selected sensor's data and not remnants from other previous visualizations.

5.5 Limitations

Some limitations have been identified within the project, some of which were intentionally introduced to prevent specific unfavourable situations. These measures were implemented to ensure that the core functionality of the application remains intact. It is anticipated that these limitations will be addressed and resolved as part of ongoing and future efforts.

One inconvenience related to the previously discussed issue is how the used *ObjImporter* plugin interprets paths. Unfortunately, it requires the full texture path to be disclosed on the mappings of the MTL file instead of its relative path. For example, the following way to define the path (the full path starting from the Unity Assets folder) will result in the correct loading of the texture:

map_Kd Assets/Models/wooden crockery/texture/honey scoop.base color.jpg

However, by using only the relative path:

map_Kd texture/honey scoop.base color.jpg

The result will not be the intended one, as can be seen by the lack of texture of the honey scoop if the model in Figure 5.8.

In the event that an MTL file uses a relative path, it creates the need for a manual adjustment of the path directly on the MTL file. It will also require a manual change to the



Figure 5.8: Wooden crockery with no texture on the honey scoop.

texture paths, when downloading texture files or transferring them from one environment to another.

One limitation purposely installed and partially explained in Section 5.2.3 is the ability to interact with objects only when the camera position is fixed. This means that while the camera is not snapped to one of the predetermined positions (it is in free mode, being moved using the arrow keys), the user will not be able to interact with the objects on the scene. This limitation was placed to prevent the dragged object from leaving the environment wall bounds.

As mentioned and reasoned in Section 5.2.1.2, the heritage environment is surrounded by *Box Colliders*. While using the application from one of the fixed camera positions, the front collider is detected and disabled; this is a simple procedure because the same collider covers the whole interactive surface. While using the free camera mode, the interactable surface comprises more than one collider, which creates the need to disable a different collider depending on the mouse's position. Unfortunately, this does not work; ideally, we would check the collider below the mouse position and disable it. But, since the back colliders will be detected after the front one is disabled, the back collider will be enabled, and a loop of back and front colliders enable-disable will take place, not allowing for any form of consistent interaction.

Disabling both colliders does not work because, in certain angles, one or both of the disabled colliders are necessary to prevent the object from pushing through the set boundaries.

As such, the interaction in the free camera mode was limited to prevent unwanted behaviours while still adding a dynamic view to the functionality of the application.

Something that is not a limitation but is important to know is that when importing

objects with complex hulls, Unity emits the warning: "Couldn't create a Convex Mesh from source mesh "mesh" within the maximum polygons limit (256). The partial hull will be used. Consider simplifying your mesh.". This partial hull is a simplified version of the mesh that still provides a good level of collision representation, but it will not be as accurate as a full mesh collider. In this project's environment and setting, the collisions work as intended with a good precision level using the simplified hull. If this were an issue, the solution would be to alter the object's 3D composition, which does not fit the scope of this project.

EVALUATION

This chapter presents the evaluation process of the developed system; it details the participant selection and the different evaluation processes they have gone through in Sections 6.1 and 6.2. Finally, it presents an analysis of the obtained results in Section 6.3.

In this chapter, we will showcase the two evaluation tests performed. The first evaluation involves hands-on testing focusing on usability and is presented in Section 6.1. Here, users will be given specific tasks meant to cover the entire application. We encourage users to share their thought processes as they work through these tasks. Users will be closely observed and offered guidance if needed during this process. We will also keep track of the time it takes to complete the tasks, both in total and for individual sections, to help identify specific potential hindrances.

This test is open to a wide range of users, and there is no need to be an expert to participate. We have intentionally chosen a diverse group of users, including some from VICARTE¹.

Once the tasks are completed, all users are asked a series of questions regarding the application's usability. The first set of questions is based on a SUS questionnaire, which provides a standard framework for evaluating usability. SUS is a 10-question usability evaluation tool that was created by John Brooke in 1986 and is a quick and reliable way to evaluate usability based on a 5-set answer system ranging from "Strongly agree" to "Strongly disagree" [33].

Section 6.2 presents the evaluation process for our specialist users who also participated in the first evaluation. It consists of an additional set of questions designed to assess the relevance of the application's features to their specific needs and some future additions to the application. This helps us understand how well the application compares to the requirements of experts in the field by whom this application is meant to be used.

Both evaluation forms collect some basic personal information like age and gender to allow a more complete study of the results.

¹<https://vicarte.org/>

Participants were provided with an informative and consent document in Portuguese for both assessments, which can be seen in Appendix A. This document, inspired by the informed consent record created during a colleague's dissertation [42], serves a vital role in this evaluation's dynamic. It aims to educate participants about the project, ensure their understanding of the tasks involved, emphasise the confidentiality of their personal information, and secure their voluntary participation without any form of compensation.

6.1 Usability Evaluation

As mentioned before, the Usability evaluation is based on a set list of tasks to be performed by the participant and a Google Forms questionnaire, available in Appendix B, to be filled after.

The given tasks cover all major points of the application that were showcased in Chapter 5 and serve as a way for the user to get to know the application, experience its flow and indulge in its varied interactions. These are organised in groups based on their nature and objective. The tasks are as follows:

1.
 - a) Import an OBJ file model called "skull" and the MTL file of the same name from the "Filipe -> EvaluationData -> skull" folder on the Desktop.
 - b) Select the object, scale it up, rotate it and place it close to one of the room's walls.
2.
 - a) Import an OBJ file model called "wooden crockery" and the MTL file of the same name from the "Filipe -> EvaluationData -> wooden crockery" folder on the Desktop.
 - b) Select the object and change its visualisation to "wireframe".
 - c) Change its visualisation back to "original".
3.
 - a) Add a sensor to the room.
 - b) Add a data file called "sensorReads" from the "Filipe -> EvaluationData" folder on the Desktop and assign it to the added sensor.
4.
 - a) Select the sensor and open its graph view.
 - b) Change the graph view to "humidity".
 - c) Change the graph view to "temperature + humidity".
 - d) Close the graph view.
5.
 - a) Change to camera position 2.
 - b) Use the free-camera mode.
 - c) Change to camera position 1.

6. a) Save the progress.
- b) Select and move one of the existing artefacts.
- c) Load the previously saved progress.

After completing these tasks, the participant is ready to answer the usability questionnaire. This questionnaire is presented as a Google Forms to facilitate result analysis and has three sets of questions. The first set is adapted from the [SUS](#) evaluation tool and is globally oriented, while the second and third sets of questions are more specific - they target the usability and understandability of the main features of the application.

6.2 Specialist Evaluation

After conducting the Usability evaluation showcased in [Section 6.1](#), if the participant's primary area of study is Conservation and they have sufficient experience, they are also eligible to complete the Specialist Evaluation. This evaluation is comprised of a questionnaire that will tackle the relevance of the features the application provides, which is available in [Appendix C](#). This is an essential part of the evaluation since it allows for direct input from the target users who will use the application in the future.

6.3 Evaluation Results

In this section, the results from both Usability and Specialist Evaluations will be presented and discussed. This is a crucial part of the project since this result collection and analysis will serve as a foundation for future decisions regarding the usability and functionality of the developed application.

This section will start by presenting an analysis of the users who participated in the evaluation and continue with the discussion of the results, including some statistical analysis.

There is a considerable amount of questions that use a "Likert" scale in the answer options. The "Likert" scale presents information in the form of statements, rather than questions, after which participants are asked to indicate how much they agreed or disagreed with the main point. It is important to note that for "Likert" scales, averaging values should be avoided to analyse results; the median is a more appropriate way to evaluate these results [5]. As such, for questions that use the "Likert" scale, the median will be used to analyse results instead of the average.

6.3.1 User Analysis

The first part of the evaluation is a usability evaluation, and as such, it is optimal to have a varied user pool to produce the most optimal results. [Figures 6.1, 6.2, and 6.3](#) showcase

some aspects and qualities of the users who took part in this evaluation and how, in general, the participants pool is, in fact, varied.

Figure 6.1 presents the gender distribution of the participants, where one can easily see the female gender is predominant. This is due to the fact that the participants who were formed in the Conservation area of studies and consequently qualified as specialised participants all identified as the female gender.

What gender do you identify with?

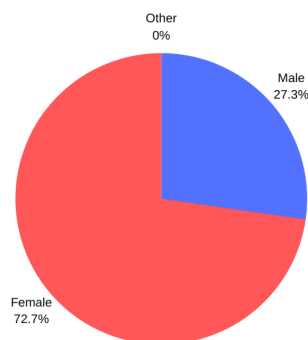


Figure 6.1: Gender distribution.

Figure 6.2 provides insight into the age distribution of the participants. Only participants who were over 18 years of age took part. This figure shows a bar graph that divides participant ages into three categories: participants under the age of twenty-five, participants between the ages of twenty-five and thirty-five, and participants over the age of thirty-five. The youngest participant was nineteen, while the oldest was 51. The average participant age was around 29.3.

Figure 6.3 shows the distribution of the level of education of the participants ranging from "High School" to "Doctorate degree". The majority of the participants had a university degree, and almost half of the participants had either a Master's degree or a Doctorate degree.

6.3.2 Usability Results

As previously mentioned, the Usability evaluation was comprised of a set list of tasks to be performed by the participant and a Google Forms questionnaire. The questionnaire is separated into the [SUS](#), a general questionnaire, and some specific questions regarding the ease of use and usability of the application's individual features.

The list of tasks presented to the participants showed in Section 6.1 was timed based on the numerate sections; in total, there were six sections. The time results are presented in Figure 6.4 ordered by the participant age. It is safe to say that age had no significant impact on how much time the participants took to complete the tasks.

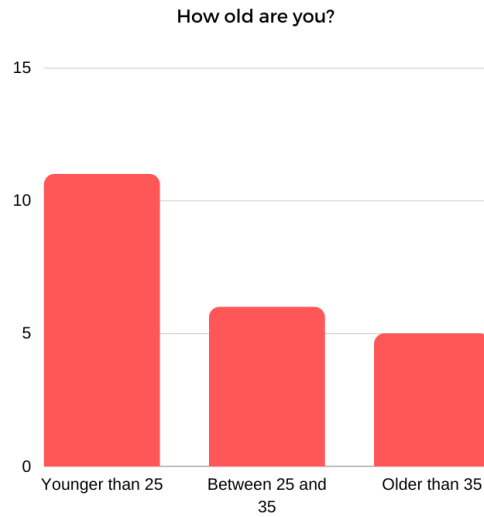


Figure 6.2: Age distribution.

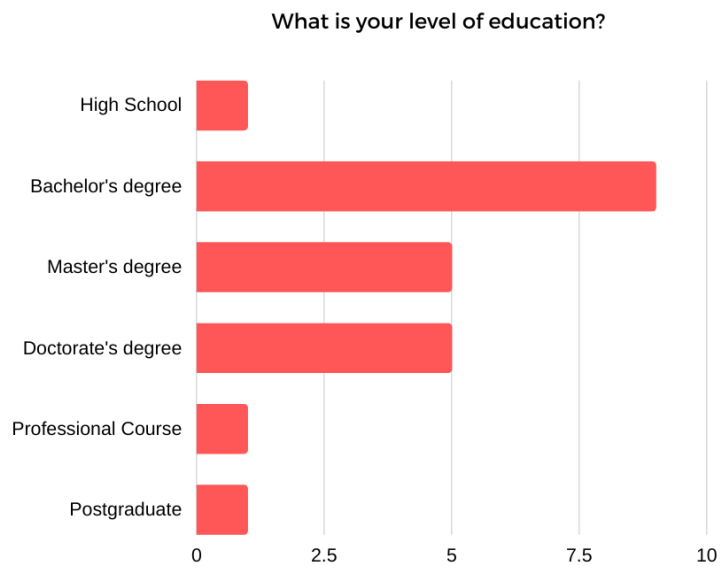


Figure 6.3: Level of education distribution.

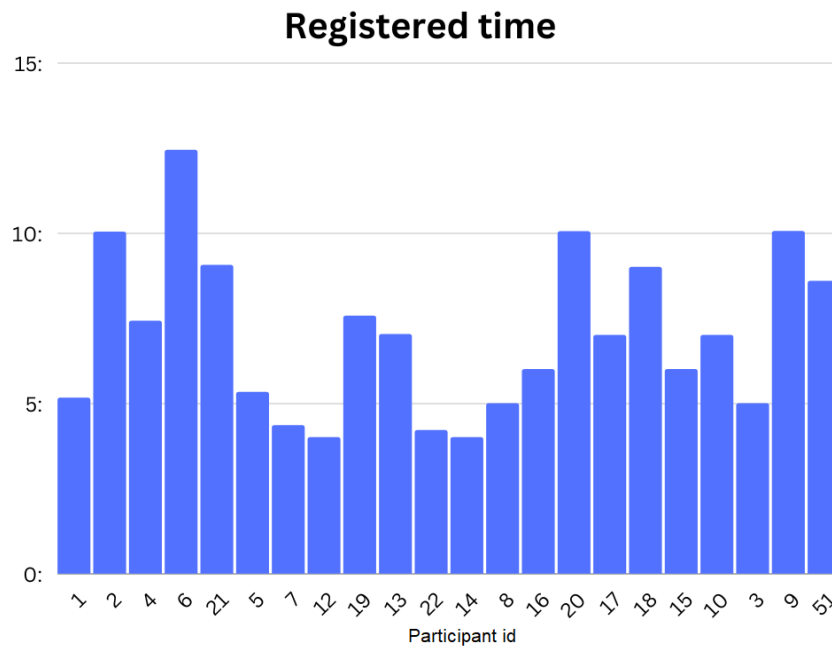


Figure 6.4: Time taken by each participant to complete the tasks, ordered by their ages.

The [SUS](#) questionnaire that was used during the evaluation sessions is based on the Portuguese translation of the following questions [33]:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The [SUS](#) score is calculated using the following formula [3]: To calculate the final result of the [SUS](#), values are assigned to each response option between zero and four, corresponding to the possible options between one and five. Thus, for statements with a positive connotation (odd-numbered statements in the enumerate above), the value of the

response is determined by the position on the scale minus one (i.e., 0-4), with the reverse calculation being performed for responses to statements with negative connotations (i.e., even-numbered statements), which leads to subtracting five from the position of the response option. The final SUS result is calculated by multiplying the sum of the response values by 2.5 so that the result falls within the range of 0-100. Based on Bangor et al. research [2], which comprises more than ten years of SUS data, a SUS score above 70 would be considered above average, and anything under 70 below average. Furthermore, products which produced scores between 75 and 89 are considered very positive, while scores above 90 are considered excellent.

The results from the SUS evaluation can be seen in Figure 6.5. The results show that more than half of the participants' answers resulted in a score above 85.

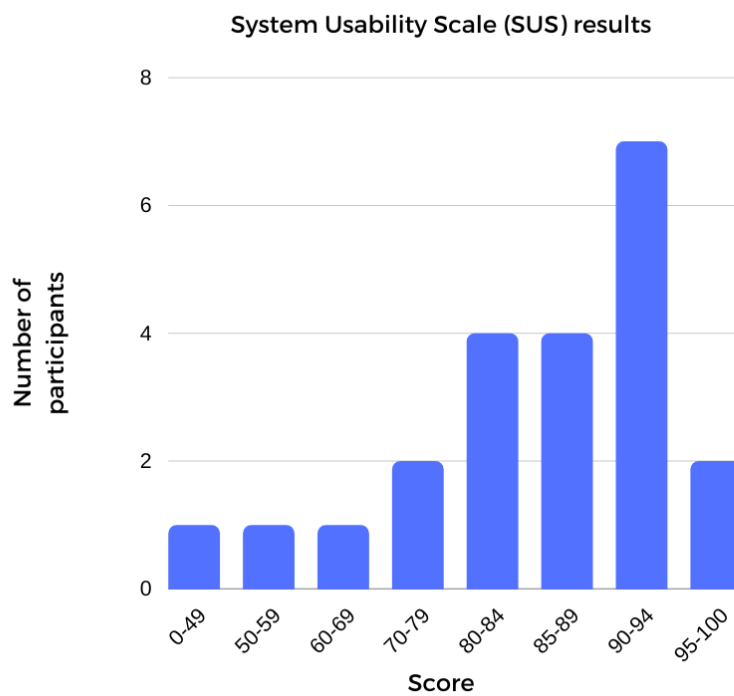


Figure 6.5: SUS results.

Table 6.1 shows a simple statistical analysis of the results and presents a very positive final average result of 82.15. The standard deviation is high, at 13, which can occur due to various factors. One reason for this is the fact that two evaluation results are very low compared to the rest. They stand at 42.5 and 52.5, while the mean score of the rest of the results stands at 86.5, a very different and positive result when compared. Another reason for this could be that the answers are answered from "Strongly Disagree" to "Strongly Agree", and both can be positively or negatively connoted based on the question; during the evaluation questionnaires and after analysing the answers, there are some cases where it seems the participant answered with the same positive/negative connotation based on the previous answer which was contrary to the current question. This would produce

some undesired results and contribute to the high standard deviation value.

Mean	Standard Deviation	Minimum	Maximum	Median
82.15	13.8	42.5	100	87.5

Table 6.1: Results from SUS.

The results of the questions that evaluated the ease of use of specific functionalities of the application were all questioned like the one in Figure 6.6 and had a median score of 5 out of 5.

This section aims to evaluate the ease-to-use of specific features of the application.

On a scale of 1 to 5, how easy to use were the listed features?

Importing a 3D model of an artefact.

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Figure 6.6: One of the feature-oriented questions.

The questions all had the same possible answer as the question in Figure 6.6, and the questions were all answered from "Very difficult" to "Very easy". The score is the median calculation of all values. A higher score means a better performance on all questions, and the maximum score is five. The questions and consequent results are:

1. Importing a 3D model of an artefact (Score: 5).
2. Interact with the imported artefact (Score: 4)
3. Adding a sensor (Score: 5).
4. Adding a data file and associating it with a sensor (Score: 5).
5. Visualizing the data on the graph (Score: 5).
6. Using the different camera positions (Score: 4).
7. Using the free-camera mode (Score: 5).
8. Using the Save/Load mechanism (Score: 5).

Figure 6.7 shows the results distribution of the set of questions referring to the ease of use of specific functionalities of the application. We can see that the mean and median values do not vary much per question, but the median value is usually slightly higher. This means the data is slightly "skewed to the left" and can indicate that, in some questions, there may exist some isolated or extremely low values when compared to the rest.

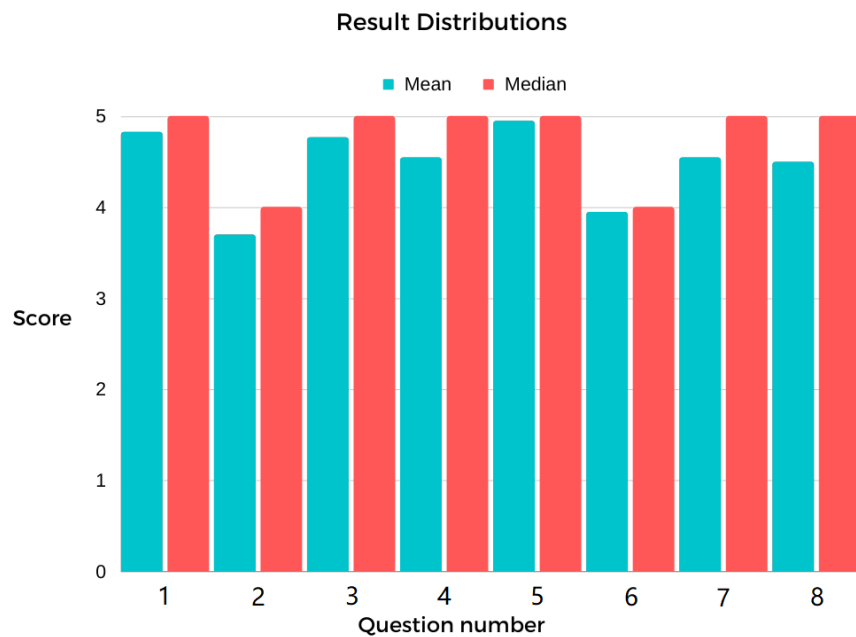


Figure 6.7: Result distribution of the usability questions referring to the ease of use of specific functionalities of the application.

One question that needs specific attention is related to the interaction with an artefact (question 2). The results of this question can be seen in Figure 6.8, and it got a median result of 4 out of 5; this was the lowest-scoring question in this section (when considering other metrics like average). It was also visible during the evaluation process that this phase of the process was where the majority of the participants struggled the most. After discussing and reading some of the suggestions left by the participants, it was clear that this was due to the way the movement of the object was implemented. As it stands, the user needs to double click and drag the object to select and move it (the second clic needs to transform into the drag motion), but most users affirmed it was better if this could be done with only one click or drag motion after the artefact was selected.

On a scale of 1 to 5, how easy to use were the listed features?

Interacting with the imported artefact.

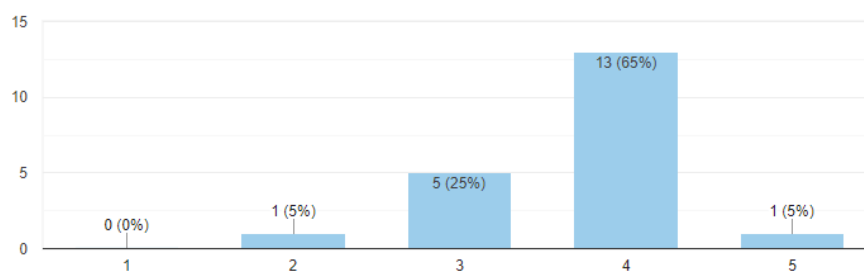


Figure 6.8: One of the feature-oriented questions.

Another point that brought some struggles to the participants was using the fixed camera positions; they can be changed using the numerical keyboard. Some of the participants who struggled stated that even though they struggled, they were happy with how this camera control was implemented. However, some suggested the camera to be changed after clicking on the desired wall for the camera to change perspective instead of using the numerical keyboard.

Based on the evaluation experience, this struggle mostly happened because the camera controls are the only ones that require the keyboard. Also, by being the last feature that was tested, the user was much more focused on the mouse and the screen content.

The next set of questions in the usability questionnaire was more focused on the understandability of the provided content and the realness of some of the user-artefact interactions. The questions were all answered with individual answering tags, depending on their content - in all cases, the score is the median calculation of all values. A higher score means better performance, and the maximum score is five. The questions and consequent results are:

1. How helpful was the help section? (Score: 5).
2. From a physical standpoint, how natural was the artefact's movement interaction? (Score: 5).
3. From a physical standpoint, how natural was the artefact's rotation interaction? (Score: 5).
4. From a physical standpoint, how natural was the artefact's scaling interaction? (Score: 5).
5. Was the content of the graph visualizations with only 1 data variable understandable? (Score: 5).
6. Was the content of the graph visualization with 2 data variables understandable? (Score: 5).
7. From a physical standpoint, how natural was the movement in the free-camera mode? (Score: 5).

Figure 6.9 shows the results distribution of the set of questions referring to the understandability and realness of specific aspects of the application. The results from this set of questions follow the same pattern as the previous one: the mean and median values are similar, but the median value is usually slightly higher. This means the data is slightly "skewed to the left" and can indicate that, in some questions, there may exist some isolated or extremely low values when compared to the rest.

The final question was an open-ended answer asking the participant to list any suggestions, problems or general observations they were willing to share. The most relevant suggestions/observations left by the participants were as follows:

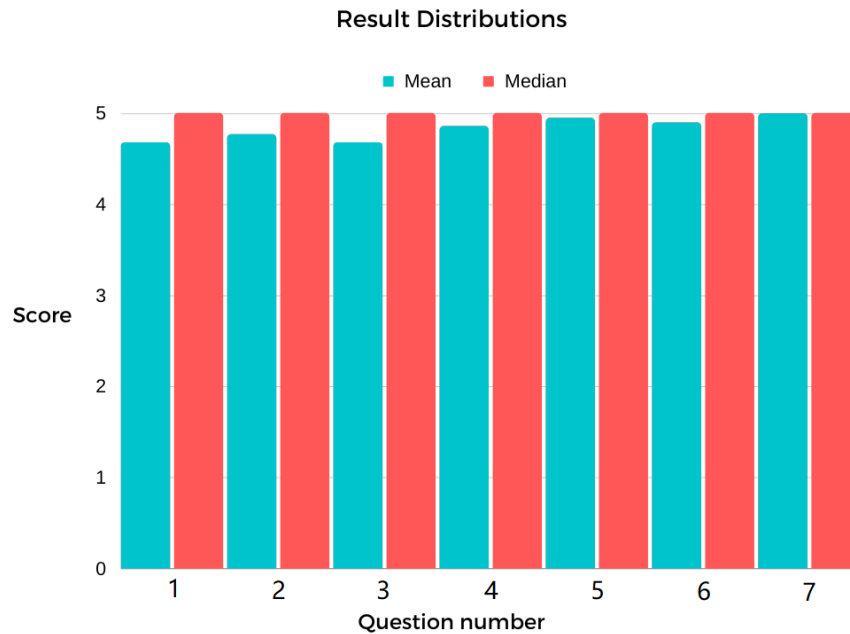


Figure 6.9: Result distribution of the usability questions referring to the understandability and realness of specific aspects of the application.

- The most commonly reported hindrance felt by the participants was the movement of the artefact, as mentioned before.
- There were also some comments on the naming of the buttons; for example, some users had issues discerning the purpose of the "Open .obj file" button and the "Load" button.
- One particular user provided some insight on a graph's standard structure and some general rules and how the application's graphs lacked some of those - axis units should not be repeated; the legend of the graph should be centred; if the data is discrete, the graph should be comprised of points instead of a line.

The results for each individual question can be seen in Appendix D.

6.3.3 Specialization Results

As mentioned in Section 6.2, if the participant's primary area of study is Conservation and has sufficient experience, they are also eligible to complete the Specialist Evaluation. As such, the participants who were qualified had to answer a new set of questions after completing the Usability evaluation. The participant ID, provided at the beginning of the session, was re-used to link all answers.

These questions aim to evaluate the relevance of the application's features, keeping in mind its goal, the environment and the conditions in which they will be used. The questions were not all of the same nature, and, as such, different answering tags were

provided depending on the question's content - in all cases, the score is the median calculation of all values. A higher score means better performance, and the maximum score is five. The questions and consequent results are:

1. How relevant is the feature of importing the OBJ models of 3D artefacts? (Score: 5).
2. How relevant is the ability to move the artefacts? (Score: 5).
3. How relevant is the ability to rotate the artefacts? (Score: 5).
4. How relevant is the ability to scale the artefacts? (Score: 4).
5. How relevant is the feature of changing the artefact visualization to "wireframe"? (Score: 2).
6. In the future, we would like to implement a feature to display a visual temporal evolution of the artefact. How relevant would this be? (Score: 5).
7. How relevant is the feature of adding sensors to the environment? (Score: 5).
8. How relevant is the feature of associating data files with sensors? (Score: 5).
9. How relevant is the feature of visualizing the multiple sensor data in a graph? (Score: 5).
10. How useful is the ability to choose from the provided fixed camera positions to better visualize the application content? (Score: 5).
11. How useful is the ability to freely change the camera's position using the provided unlimited camera movement (free-mode) to better visualize the application content? (Score: 5).
12. How relevant is the feature of saving and loading the session progress? (Score: 5).
13. How satisfied are you with the features provided by this application? (Score: 5).

Figure 6.10 shows the results distribution of the set of questions referring to the understandability and realness of specific aspects of the application. The results from this set of questions are similar to the previous ones. The mean and median values are similar, but the median value is slightly higher in some cases. This means the data is slightly "skewed to the left" and can indicate that, in some questions, there may exist some isolated or extremely low values when compared to the rest. Question four was the only case of the opposite happening; it suggests that there are some isolated or extremely high values instead.

The results were, in general, very positive and showed that almost all of this application's functionalities fit the defined objectives. One important thing to note is that question five got a median result of two out of five, which was foreseeable since, in the future, it will

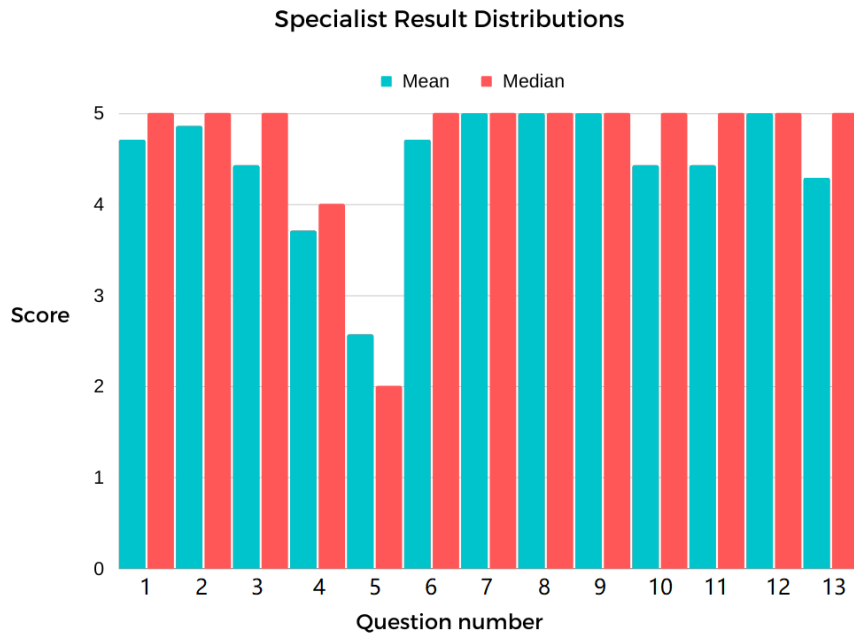


Figure 6.10: Result distribution of the specialist questions referring to the relevance of some of the application’s features.

be replaced by a visualization to display the visual evolution of the artefact, as mentioned in Section 7.2. Question six results positively support this decision as it achieved a median score of five out of five.

The final question mirrored the one in the Usability evaluation, an open-answer question that asked the participant for any suggestions or features they would like to see implemented in the future. More than half of the participants had nothing to add; these were the comments received:

- Providing lighting data in the form of sensors or by specifying the lighting conditions of the room, which play a big part in the conservation of the artefacts
- Having the ability to compare data from different sensors in the same visualization.
- Having the ability to compare data from different sensors on specific time values.

These comments are all very pertinent and fit the scope of the project. Ideally, these modifications would be made in the future.

The results for each individual question can be seen in Appendix D.

CONCLUSIONS AND FUTURE WORK

In the final chapter of the dissertation, the main conclusions of the work carried out are described (Section 7.1), followed by the proposal of certain improvements to the developed components, as well as possible starting points for future advancements of the platform and the resulting overall system (Section 7.2).

7.1 Conclusions

The produced system aims to help professionals control and manage heritage artefacts present at museums or other controlled environments. This system was designed and developed to entail specifications designed by the team at Palacio Nacional de Sintra of Parques Sintra in the early stages of work.

All the literature studied, and all the knowledge gained during the early stages of this dissertation's work was used throughout the solution's implementation to ensure everything is developed to fit the future mould that this application will define within a complex cultural and Digital Twin environment.

The application was built using Unity Engine and with two very important plugins. *ObjImporter*, which facilitates the importing of OBJ files into Unity and *Graph and Chart*, which facilitates the creation of 2/3D chart visualizations in Unity.

The application was built in a manner to ensure that users can import 3D models of artefacts to be monitored and displayed; these 3D models can originate from on-field scans using digital scanners, like the ones showcased in Section 4.2, and are imported with the help of Unity's *ObjImporter* plugin. After the models are imported, the user can interact with the artefacts to place, rotate and scale at free will; this allows the user to recreate and manage real-life environments and also experiment with them, all from within the application.

To have a complete and functional monitoring of this cultural environment filled with heritage artefacts, sensor data like temperature and humidity is imperative. With this in

mind, the ability to place sensors in the virtual environment and add data files to these entities was created. By having several sensors placed and with all the associated data being available to visualise in a graph using Unity's Graph and Chart plugin, the user can then have a clearer picture of the environmental conditions present in the environment and make adequate decisions accordingly.

To ensure that the application can be used consistently, a save and load mechanism was added. This allows the user to load previously saved scenes and continue previous works; it is crucial to have this ability when using a platform that is supposed to have a continuous flow of work.

The proposed objectives in Chapter 1 were successfully accomplished. However, due to time and material limitations, the section related to the artefacts and environment data had to be revised. The idea of establishing a real-time, bidirectional data flow between real sensors and the system had to be adapted; instead of real-time data transmission, processed data had to be manually integrated into the platform's sensor inputs.

By looking at the analysis of the evaluation phase results in Section 6.3.2, it can be concluded that the participant response to the system's usability was very positive. The evaluation results were also considered positive for the specialised users, as shown in Section 6.3.3. Nonetheless, a few key points that should be improved or changed were detected in both evaluations, which are highlighted in Sections 6.3.2 and 6.3.3.

7.2 Future Work

This project has been designed with a focus on both current and forthcoming advancements. While certain choices are explicitly aimed at future-specific developments and features, others provide a foundation for improvements.

The core purpose of this project is to lay the groundwork for the eventual integration of the Digital Twins in the application. This initiative strives to create an adaptable platform that is ready to accommodate the seamless incorporation of the Digital Twin technology studied in Chapter 2.

As elaborated upon in Section 5.5, considering time limitations, deliberate strategic choices were taken. Additionally, certain challenges were showcased, and it is anticipated that these, alongside the placed constraints, will ideally find resolution in future works.

One pivotal area of future development revolves around establishing real-time data streams sourced directly from sensors. This approach would either replace or supplement the current practice of assigning processed data to sensors within the heritage environment. This transformative shift involves harnessing the capabilities of live sensor data, leading to an implementation process that becomes finely aligned with the intricacies of sensor dynamics.

Given the intricacies involved, it becomes imperative to study and discuss with Parques Sintra the most appropriate and effective types of sensors to be employed. This will ensure

that the foundation of the implementation is laid upon a thorough understanding of the sensor landscape, enhancing the project's efficiency.

Moreover, as the process towards defining the Digital Twin's functional specifications continues, another dimension comes into play: the management of real-time data. This aspect involves not only capturing the live sensor inputs but also dynamically adapting and reacting to these inputs. This would create a bi-directional flow of data, mirroring the interactive exchanges that have been explored and documented in the reviewed literature of Chapter 3.

It is important to recognize that this progression encapsulates the fundamental milestones on the path towards realizing a fully functional Digital Twin. By effectively leveraging real-time sensor data, selecting the appropriate sensors, and implementing responsive adaptation to the live data, a fully functional Digital Twin will emerge.

Currently, the application offers a "wireframe" visualization as an alternative to the original texture of the artefact. This can only be relevant in some specific situations and is highly dependent on the object's structure file. As such, a different visualization would be more beneficial. After some discussion with the National Palace of Sintra representatives, the goal of having a visualization that displayed an evolution of the object's visual aspect was established. Artefacts that are significantly eroded or have structural damage would be the ones used in this process.

During this dissertation's work, some limitations were found, and some requirements that ideally would not exist had to be placed on specific features. These are discussed in detail in Section 5.5. Removing such limitations will significantly enhance the user experience and widen the applicability of the system.

In essence, this project seeks to pave the way for the effective integration of Digital Twin's of heritage artefacts in the system, with due consideration for technical and circumstantial Digital Twin's requirements, future expansions, real-time data integration, threshold-based responses, and immersive user interaction experience.

BIBLIOGRAPHY

- [1] S. Azhar, M. Khalfan, and T. Maqsood. In: *The Australasian Journal of Construction Economics and Building* 12.4 (2012), pp. 15–28. URL: <https://search.informit.org/doi/10.3316/informit.013120167780649> (cit. on p. 8).
- [2] Bangor et al. “The System Usability Scale (SUS): an Empirical evaluation”. In: *International Journal of Human-Computer Interaction* 24 (2008-08), pp. 574–. DOI: [10.1080/10447310802205776](https://doi.org/10.1080/10447310802205776) (cit. on p. 83).
- [3] J. Brooke. “SUS: A quick and dirty usability scale”. In: *Usability Eval. Ind.* 189 (1995-11) (cit. on p. 82).
- [4] W. Choi et al. “Smart home and internet of things: A bibliometric study”. In: *Journal of Cleaner Production* 301 (2021), p. 126908. ISSN: 0959-6526. DOI: <https://doi.org/10.1016/j.jclepro.2021.126908>. URL: <https://www.sciencedirect.com/science/article/pii/S0959652621011276> (cit. on p. 10).
- [5] F. Decker. *How to interpret Likert Surveys*. 2018-03. URL: <https://sciencing.com/interpret-likert-surveys-8573143.html> (cit. on p. 79).
- [6] Dezen-Kempter, Eloisa, Mezencio, Davi Lopes, Miranda, Erica De Matos, De Sá, Danilo Pico and Dias, Ulisses. “Towards a Digital Twin for Heritage Interpretation - from HBIM to AR visualization”. UNICAMP, 2020. DOI: <https://doi.org/10.52842/conf.caadria.2020.2.183> (cit. on pp. 21, 22).
- [7] D. Essex. *What is a Digital Twin?* 2022-11. URL: <https://www.techtarget.com/searcherp/definition/digital-twin> (cit. on p. 7).
- [8] A. Fuller et al. “Digital Twin: Enabling Technologies, Challenges and Open Research”. In: *IEEE Access* 8 (2020), pp. 108952–108971. DOI: [10.1109/ACCESS.2020.2998358](https://doi.org/10.1109/ACCESS.2020.2998358) (cit. on pp. 6, 7, 9, 10, 19, 20, 29).
- [9] A. Al-Fuqaha et al. “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. In: *IEEE Communications Surveys Tutorials* 17.4 (2015), pp. 2347–2376. DOI: [10.1109/COMST.2015.2444095](https://doi.org/10.1109/COMST.2015.2444095) (cit. on p. 1).

- [10] E. Glaessgen and D. Stargel. “The digital twin paradigm for future NASA and U.S. air force vehicles”. In: *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*. 2012-04. ISBN: 978-1-60086-937-2. DOI: [10.2514/6.2012-1818](https://doi.org/10.2514/6.2012-1818) (cit. on p. 5).
- [11] M. Grieves. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*. 2015-03 (cit. on p. 5).
- [12] W. Hu et al. “Digital twin: a state-of-the-art review of its enabling technologies, applications and challenges”. In: *Journal of Intelligent Manufacturing and Special Equipment* 2.1 (2021), pp. 1–34. DOI: [10.1108/JIMSE-12-2020-010](https://doi.org/10.1108/JIMSE-12-2020-010) (cit. on p. 1).
- [13] IBM. *What is a digital twin?* URL: <https://www.ibm.com/topics/what-is-a-digital-twin> (cit. on pp. 5–7).
- [14] IBM. *What is Artificial Intelligence (AI) ?* URL: <https://www.ibm.com/topics/artificial-intelligence> (cit. on p. 10).
- [15] IBM. *What is deep learning?* URL: <https://www.ibm.com/topics/deep-learning> (cit. on p. 11).
- [16] IBM. *What is machine learning?* URL: <https://www.ibm.com/topics/machine-learning> (cit. on p. 11).
- [17] IBM. *What is natural language processing?* URL: <https://www.ibm.com/topics/natural-language-processing> (cit. on p. 11).
- [18] ICOM-CC. *Environmental guidelines ICOM-CC and IIC declaration*. URL: <https://www.icom-cc.org/en/environmental-guidelines-icom-cc-and-iic-declaration> (cit. on p. 17).
- [19] ICOM-CC. *Terminology for Conservation*. URL: <https://www.icom-cc.org/en/terminology-for-conservation> (cit. on pp. 13, 25).
- [20] P. Jouan and P. Hallot. “Digital Twin: Research Framework to Support Preventive Conservation Policies”. In: *ISPRS International Journal of Geo-Information* 9.4 (2020). ISSN: 2220-9964. URL: <https://www.mdpi.com/2220-9964/9/4/228> (cit. on pp. 26, 27).
- [21] G. A. Lee et al. “CityViewAR: A mobile outdoor AR application for city visualization”. In: *2012 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH)*. 2012, pp. 57–64. DOI: [10.1109/ISMAR-AMH.2012.6483989](https://doi.org/10.1109/ISMAR-AMH.2012.6483989) (cit. on p. 12).
- [22] J. M. Lourenço. *The NOVAthesis L^AT_EX Template User’s Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (cit. on pp. ii, iii).

-
- [23] Z. Lv and S. Xie. *Artificial intelligence in the digital twins: State of the art, challenges, and future research topics*. 2021. DOI: [10.12688/digitaltwin.17524.1](https://doi.org/10.12688/digitaltwin.17524.1). URL: <https://doi.org/10.12688/digitaltwin.17524.1> (cit. on p. 11).
 - [24] A. Marra et al. "Combining Integrated Informative System and Historical Digital Twin for Maintenance and Preservation of Artistic Assets". In: *Sensors* 21.17 (2021). ISSN: 1424-8220. DOI: [10.3390/s21175956](https://doi.org/10.3390/s21175956). URL: <https://www.mdpi.com/1424-8220/21/17/5956> (cit. on pp. 8, 22–25).
 - [25] M. Murphy, E. McGovern, and S. Pavía. "Historic building information modelling (HBIM)". In: *Structural Survey* 27 (2009-08), pp. 311–327. DOI: [10.1108/02630800910985108](https://doi.org/10.1108/02630800910985108) (cit. on p. 8).
 - [26] T. Navman. *The Ultimate Guide to IOT - Transport & Logistics Edition*. URL: <https://www.teletracnavman.com/fleet-management-software/telematics/resources/iot-in-transportation-logistics-the-ultimate-guide> (cit. on p. 10).
 - [27] F. Niccolucci, A. Felicetti, and S. Hermon. "Populating the Data Space for Cultural Heritage with Heritage Digital Twins". In: *Data* 7.8 (2022). ISSN: 2306-5729. DOI: [10.3390/data7080105](https://doi.org/10.3390/data7080105). URL: <https://www.mdpi.com/2306-5729/7/8/105> (cit. on pp. 26, 28–30).
 - [28] Oracle. *What is the internet of things (IOT)?* URL: <https://www.oracle.com/pt/internet-of-things/what-is-iot/> (cit. on p. 9).
 - [29] J. Otero. "Heritage Conservation Future: Where We Stand, Challenges Ahead, and a Paradigm Shift". In: *Global Challenges* 6.1 (2022), p. 2100084. DOI: <https://doi.org/10.1002/gch2.202100084>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/gch2.202100084>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/gch2.202100084> (cit. on p. 1).
 - [30] T. D. Palma. *Architectural surveys: What are they for and how are they prepared?* 2023-07. URL: <https://biblus.accasoftware.com/en/architectural-surveys-what-are-they-for-and-how-are-they-prepared/> (cit. on p. 8).
 - [31] Parques de Sintra. *Palácio Nacional de Sintra*. URL: <https://www.parquesdesintra.pt/pt/parques-monumentos/palacio-nacional-sintra/> (cit. on p. 15).
 - [32] C. Pazzanese. *Ethical concerns mount as AI takes bigger decision-making role*. 2020-12. URL: <https://news.harvard.edu/gazette/story/2020/10/ethical-concerns-mount-as-ai-takes-bigger-decision-making-role/> (cit. on p. 11).
 - [33] A. S. f. Public Affairs. *System Usability Scale (SUS)*. 2013-09. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (cit. on pp. 77, 82).
 - [34] A. Saini. *SainiStaticKeyword*. 2021-05. URL: <https://www.geeksforgeeks.org/static-keyword-in-c-sharp/> (cit. on p. 45).

- [35] South West Museums, Libraries, and Archives Council. *Signposts to Collections Care: Factsheet 4 - Preventive Conservation*. 2006-03. URL: <https://collectionstrust.org.uk/wp-content/uploads/2017/01/South-West-Museums-Libraries-and-Archives-Council-Signposts-to-Collections-Care-factsheet-4-Preventive-Conservation-March-2006.pdf> (cit. on p. 17).
- [36] Techtarget. *What is Augmented Reality*. URL: <https://www.techtarget.com/whatis/definition/augmented-reality-AR> (cit. on p. 12).
- [37] UNESCO. *Charter on the Preservation of the Digital Heritage*. 2009. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000179529> (cit. on p. 14).
- [38] UNESCO. *Concept of digital heritage*. 2019-04. URL: <https://en.unesco.org/themes/information-preservation/digital-heritage/concept-digital-heritage> (cit. on p. 14).
- [39] UNESCO. *Concept of digital preservation*. 2019-04. URL: <https://en.unesco.org/themes/information-preservation/digital-heritage/concept-digital-preservation> (cit. on p. 14).
- [40] UNESCO. *Conservation of cultural heritage*. 2022-09. URL: <https://uis.unesco.org/node/3080254> (cit. on p. 14).
- [41] K. Van Balen. "Challenges that Preventive Conservation poses to the Cultural Heritage documentation field". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W5* (2017-08), pp. 713–717. DOI: [10.5194/isprs-archives-XLII-2-W5-713-2017](https://doi.org/10.5194/isprs-archives-XLII-2-W5-713-2017) (cit. on p. 25).
- [42] J. Vilar. "Sistema para Criação e Exploração de Narrativas Interativas e Jogos". MSc Dissertation in Computer Science. Universidade NOVA de Lisboa, 2022 (cit. on p. 78).
- [43] B. Wagner. *Reference types - C# reference*. 2023-06. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/reference-types> (cit. on p. 70).
- [44] S. Zahoor and R. N. Mir. "Resource management in pervasive Internet of Things: A survey". In: *Journal of King Saud University - Computer and Information Sciences* 33.8 (2021), pp. 921–935. ISSN: 1319-1578. DOI: <https://doi.org/10.1016/j.jksuci.2018.08.014>. URL: <https://www.sciencedirect.com/science/article/pii/S1319157818305858> (cit. on p. 10).

A

CONSENT AND INFORMATIVE DOCUMENTS

Informed Consent for Research

NOVA LINCS, Computer Science Department, Faculdade de Ciências e Tecnologias,
Universidade NOVA de Lisboa

Developing Support for Cultural Digital Twin in Conservation - 3D

RESEARCHERS

Filipe Miranda - Computer Science Department, FCT-NOVA

fm.miranda@campus.fct.unl.pt

Prof. Armanda Rodrigues - Computer Science Department, FCT-NOVA

a.rodrigues@fct.unl.pt

Prof. Nuno Correia - Computer Science Department, FCT-NOVA

nmc@fct.unl.pt

Prof. Márcia Vilarigues - Conservation and Restoration Department, FCT-NOVA

mgv@fct.unl.pt

DESCRIPTION OF THE RESEARCH WORK

The purpose of this project was to develop an application designed to support professionals in the administration, exploration, and supervision of artifacts present in a real environment, recreating them digitally. Upon its completion, this application will be used by professionals at the National Palace of Sintra to investigate different approaches to displaying their artifacts in a room. The choices will be guided by data captured by sensors in the room. For example, a delicate piece, whose chemical composition is more sensitive to humidity, should be placed in an area of the room with reduced humidity levels.

STUDY PURPOSE

The current study aims to conduct a thorough analysis of the usability and relevance of the developed application. The participant will use the application following a predefined set of tasks. After exploring the features, the participant will be asked to respond to a set of questions accessible through an online form. In the event that the participant is identified as an expert, they will then be invited to fill out a second online questionnaire.

CONFIDENTIALITY

During the questionnaire completion, refrain from providing any information that would reveal your identity. The researchers will make every effort to ensure the confidentiality of participants by assigning numerical codes that will be used in all notes and research documents instead of personal identification.

Participants' personal data will be kept strictly confidential during data analysis and in the presentation of results in research documents and/or scientific conferences. Only the researchers mentioned at the beginning of this document will have access to the data and questionnaire responses, which will be securely stored. Storage will be conducted to

preserve confidentiality, using the numerical reference assigned at the beginning of participation instead of personal identification.

The study session will be conducted in person. During the session, notes may be taken, and recording will be done solely for reference during the data analysis process and promptly deleted upon completion of that process.

VOLUNTEER PARTICIPATION

Participation in this study is strictly voluntary, and deciding to participate is entirely your choice. You will be asked to sign the consent form only if you choose to participate in the study. After signing the consent form, you still have the freedom to withdraw at any time without the need to provide a justification. If you decide to discontinue your participation before the data collection is complete, any data collected up to that point will be promptly deleted.

It is important to note that participation in the study will not result in personal or financial benefits.

CONSENT

I have read and understood the information provided. I had the opportunity to ask questions, and I am aware that my participation is voluntary. I have the freedom to discontinue my participation at any time without the need to provide a reason and without associated costs. I voluntarily agree to participate in this study.

The signature below indicates that you are voluntarily giving your consent to participate in this study.

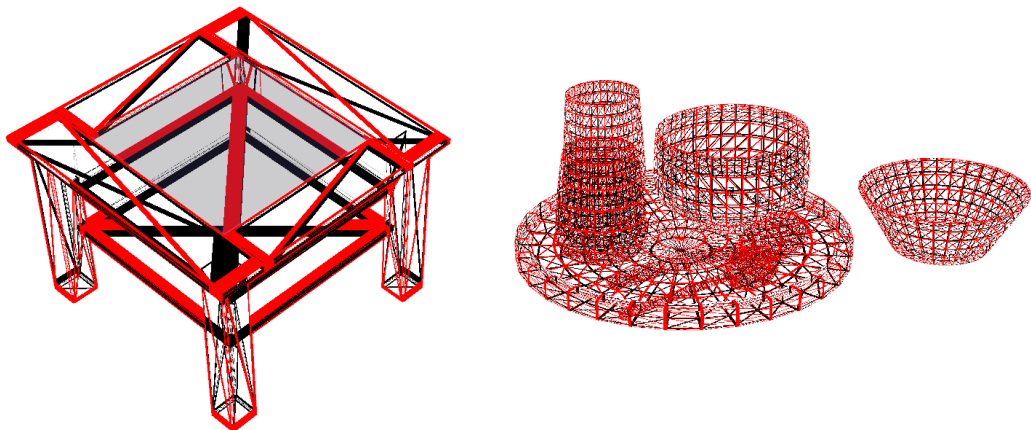
Signature: _____

Date: _____

Important things to know throughout the evaluation:

The goal of the application is to help professionals manage, explore, and monitor artefacts from a real environment by recreating it in a digital manner. Ideally, it will be used by personnel from cultural institutions to explore different ways to display their artefacts in a room. Sensor data from the room would influence the decisions. For example, a delicate piece whose chemical composition is more susceptible to humidity should stay in a place in the room with low humidity values.

Wireframe is a visual representation of a three-dimensional (3D) physical object used in 3-D computer graphics. A 3D wireframe model is a skeletal representation of any real-world object. It consists of various points, arcs, lines, circles, and curves to denote object edges and depth cleanly. The term “Wireframe” is derived from the origin when designers made use of metal wires to represent an object in three-dimensional space and shape.



The **camera's free mode** is how the user can move the camera around the room with no restrictions using the arrow keys on the keyboard.

| B

USABILITY TEST



Usability Test - Developing support for the Heritage Digital Twin in Conservation 3D

This form is designed to assess the usability of the developed application in the scope of the "Developing support for the Heritage Digital Twin in Conservation - 3D" dissertation. You will be asked to provide some personal information and prior to completing the technical part of this form, you will be given given specific tasks to perform within the application. After, we invite you to provide feedback and insights based on your interactions, helping us gain valuable insights into the application's strengths and areas for improvement. Your input is highly valuable, and we appreciate your time and effort in contributing to our research.

The goal of the application you will test is to help professionals manage, explore, and monitor artefacts from a real environment by recreating it in a digital manner. After it is finished, this application will be used by professionals from Sintra's National Palace to explore different ways to display their artefacts in the museum's rooms. The decisions would be influenced by sensor data from the room. For example, a delicate piece whose chemical composition is more susceptible to humidity should stay in a place in the room with low humidity values.

Personal Information

In this section we ask you to provide some of your personal information.

What is your user ID? *

A sua resposta

What gender do you identify with? *

- ☐ Female
- ☐ Male
- ☐ Other
- ☐ Prefer not to say

How old are you? *

A sua resposta

What is your level of education? *

A sua resposta

What is your current occupation/area of studies? *

A sua resposta

Anterior

Seguinte



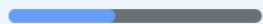
Página 2 de 7

Limpar
formulário

Task List

In this section you should minimize the form and complete the following task list using the provided application. The tasks are grouped based on their nature for research purposes.

1.
 - a) Import an OBJ file model called "skull" and the MTL file of the same name from the "Filipe -> EvaluationData -> skull" folder on the Desktop.
 - b) Select the object, scale it up, rotate it and place it close to one of the room's walls.
2.
 - a) Import an OBJ file model called "wooden crockery" and the MTL file of the same name from the "Filipe -> EvaluationData -> wooden crockery" folder on the Desktop.
 - b) Select the object and change its visualization to "wireframe".
 - c) Change its visualization back to "original".
3.
 - a) Add a sensor to the room.
 - b) Add a data file called "sensorReads" from the "Filipe -> EvaluationData" folder on the Desktop and assign it to the added sensor.
4.
 - a) Select the sensor and open its graph view.
 - b) Change the graph view to "humidity".
 - c) Change the graph view to "temperature + humidity".
 - d) Close the graph view.
5.
 - a) Change to camera position 2.
 - b) Use the free-camera mode.
 - c) Change to camera position 1.
6.
 - a) Save the progress.
 - b) Select and move one of the existing artefacts.
 - c) Load the previously saved progress.

[Anterior](#)[Seguinte](#)

Página 3 de 7

[Limpar
formulário](#)

System Usability Scale (SUS)

This section consists of a 10 item questionnaire with five response options ranging from Strongly disagree to

Strongly agree. Answer each question based on your experience with the application. You should respond according to this scale:

1. Strongly Disagree
2. Disagree
3. Neutral
4. Agree
5. Strongly Agree

I think that I would like to use this system frequently. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I found the system unnecessarily complex. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I thought the system was easy to use. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I think that I would need the support of a technical person to be able to use this system. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I found the various functions in this system were well integrated. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I thought there was too much inconsistency in this system. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I would imagine that most people would learn to use this system very quickly. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I found the system very cumbersome to use. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I felt very confident using the system. *

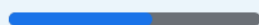
	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

I needed to learn a lot of things before I could get going with this system. *

	1	2	3	4	5	
Strongly Disagree	☐	☐	☐	☐	☐	Strongly Agree

[Anterior](#)

[Seguinte](#)



Página 4 de 7

[Limpar
formulário](#)

This section aims to evaluate the ease-to-use of specific features of the application.

On a scale of 1 to 5, how easy to use were the listed features?

Importing a 3D model of an artefact. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Interacting with the imported artefacts. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Adding a sensor. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Adding a data file and associating it with a sensor. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Visualizing the data on the graph. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Using the different camera positions. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Using the camera without restrictions (free-camera mode). *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

Save and Load the progress. *

	1	2	3	4	5	
Very difficult	☐	☐	☐	☐	☐	Very easy

[Anterior](#)

[Seguinte](#)

Página 5 de 7

[Limpar formulário](#)

This section aims to evaluate how natural and understandable some parts of the application are.

On a scale of 1 to 5, answer each of the following questions in accordance with their content:

How helpful was the help section? *

	1	2	3	4	5	
Very unhelpful	☐	☐	☐	☐	☐	Very helpful

From a physical standpoint, how natural was the artefact's movement interaction? *

	1	2	3	4	5	
Very unnatural	☐	☐	☐	☐	☐	Very natural

From a physical standpoint, how natural was the artefact's rotation interaction? *

	1	2	3	4	5	
Very unnatural	☐	☐	☐	☐	☐	Very natural

From a physical standpoint, how natural was the artefact's scaling interaction? *

	1	2	3	4	5	
Very unnatural	☐	☐	☐	☐	☐	Very natural

Was the content of the graph visualizations with only 1 data variable understandable? *

1 2 3 4 5

Incomprehensible ☐ ☐ ☐ ☐ ☐ Very understandable

Was the content of the graph visualization with 2 data variables understandable? *

1 2 3 4 5

Incomprehensible ☐ ☐ ☐ ☐ ☐ Very understandable

From a physical standpoint, how natural was the movement in the free-camera mode? *

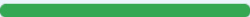
1 2 3 4 5

Very unnatural ☐ ☐ ☐ ☐ ☐ Very natural

[Anterior](#) [Seguinte](#)  Página 6 de 7 [Limpar formulário](#)

Do you have any suggestions or final considerations? *

A sua resposta

[Anterior](#) [Enviar](#)  Página 7 de 7 [Limpar formulário](#)

| C

SPECIALIST TEST

Specialist Test - Developing support for the Heritage Digital Twin in Conservation 3D

This form should only be completed after completing the "Usability Test" form.

This evaluation is comprised of a questionnaire that will tackle the relevance of the features the application provides. This is an essential part of the evaluation since it allows for direct input from the target users who will use the application in the future.

While answering these questions keep in mind that the goal of the application you tested is to help professionals manage, explore, and monitor artefacts from a real environment by recreating it in a digital manner.

After it is finished, this application will be used by professionals from Sintra's National Palace to explore different ways to display their artefacts in a room. The decisions would be influenced by sensor data from the room. For example, a delicate piece whose chemical composition is more susceptible to humidity should stay in a place in the room with low humidity values.

Personal Information

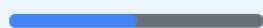
In this section we ask you to provide some of your personal information.

What is your user ID? *

A sua resposta

Anterior

Seguinte

 Página 2 de 4

Limpar
formulário

This section will tackle the relevance of the features provided by the application.

On a scale of 1 to 5, answer each of the following questions in accordance with their content:

How relevant is the feature of importing the OBJ models of 3D artefacts? *

	1	2	3	4	5	
Very irrelevant	☐	☐	☐	☐	☐	Very relevant

How relevant is the feature of moving the artefacts? *

	1	2	3	4	5	
Very irrelevant	☐	☐	☐	☐	☐	Very relevant

How relevant is the feature of rotating the artefacts? *

	1	2	3	4	5	
Very irrelevant	☐	☐	☐	☐	☐	Very relevant

How relevant is the feature of scaling the artefacts? *

	1	2	3	4	5	
Very irrelevant	☐	☐	☐	☐	☐	Very relevant

How relevant is the feature of changing the artefact visualization to wireframe? *

1

2

3

4

5

Very irrelevant

☐

☐

☐

☐

☐

Very relevant

In the future, we would like to implement a feature to display a visual temporal evolution of the artefact. How relevant would this be? *

1

2

3

4

5

Very irrelevant

☐

☐

☐

☐

☐

Very relevant

How relevant is the feature of adding sensors to the environment? *

1

2

3

4

5

Very irrelevant

☐

☐

☐

☐

☐

Very relevant

How relevant is the feature of associating data files with sensors? *

1

2

3

4

5

Very irrelevant

☐

☐

☐

☐

☐

Very relevant

How relevant is the feature of visualizing the multiple sensor data in a graph? *

1

2

3

4

5

Very irrelevant

☐

☐

☐

☐

☐

Very relevant

How useful is the ability to choose from the provided fixed camera positions to better visualize the application content? *

	1	2	3	4	5	
Not Very Useful	☐	☐	☐	☐	☐	Very Useful

How useful is the ability to freely change the camera's position using the provided unlimited camera movement (free-mode) to better visualize the application content? *

	1	2	3	4	5	
Not Very Useful	☐	☐	☐	☐	☐	Very Useful

How relevant is the feature of saving and loading the session progress? *

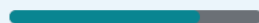
	1	2	3	4	5	
Very irrelevant	☐	☐	☐	☐	☐	Very relevant

How satisfied are you with the features provided by this application? *

	1	2	3	4	5	
Very Dissatisfied	☐	☐	☐	☐	☐	Very Satisfied

[Anterior](#)

[Seguinte](#)



Página 3 de 4

[Limpar
formulário](#)

Can you list any features you feel are missing from the application's functionality? *

A sua resposta

[Anterior](#)

[Enviar](#)



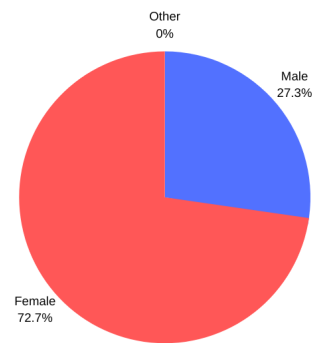
Página 4 de 4

[Limpar
formulário](#)

D

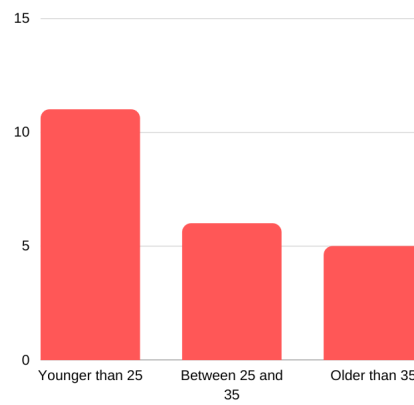
EVALUATION RESULTS

What gender do you identify with?



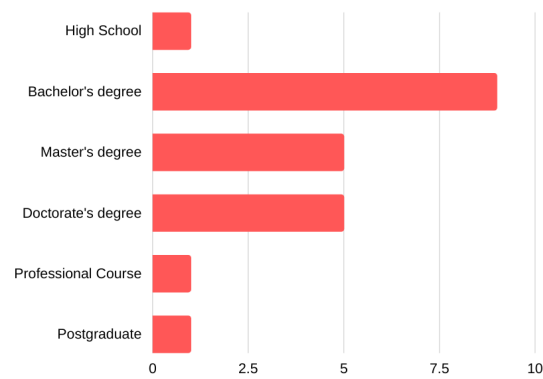
(a) Gender distribution of the participants.

How old are you?



(b) Age distribution of the participants.

What is your level of education?



(c) Level of education distribution of the participants.

Figure D.1: User Analysis.

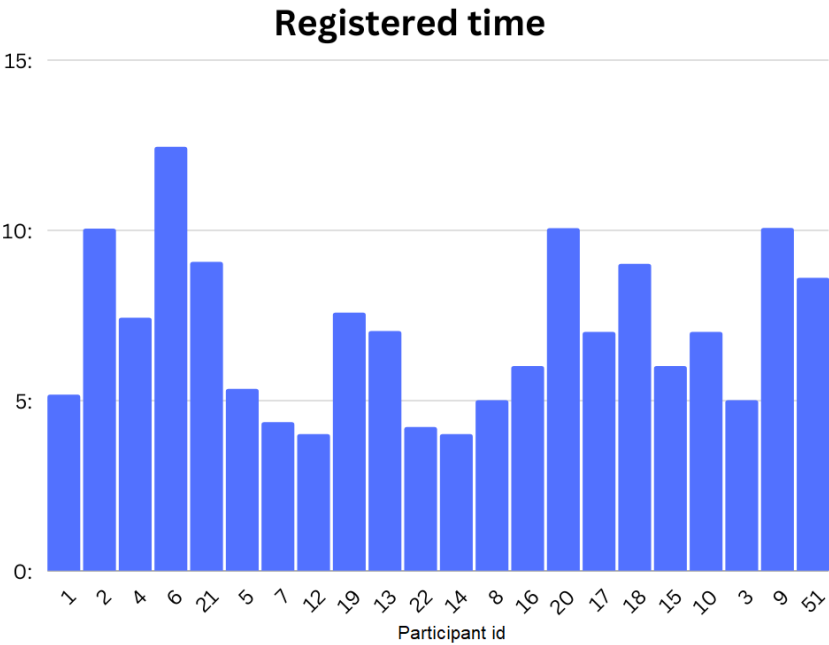
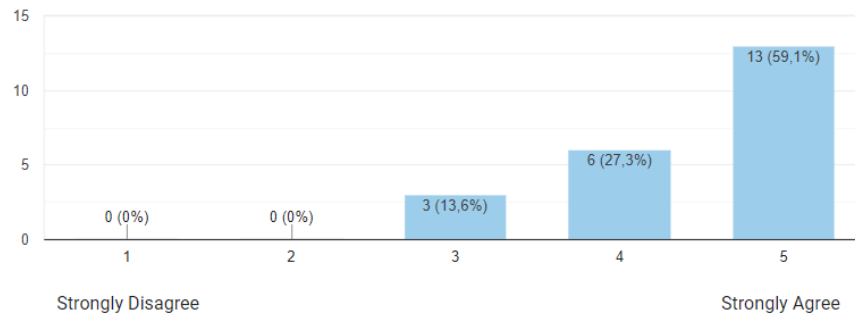


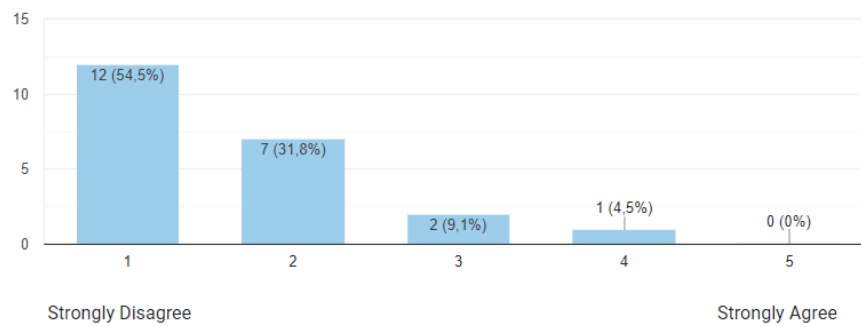
Figure D.2: Time taken to complete the list of tasks, ordered by age.

I think that I would like to use this system frequently.



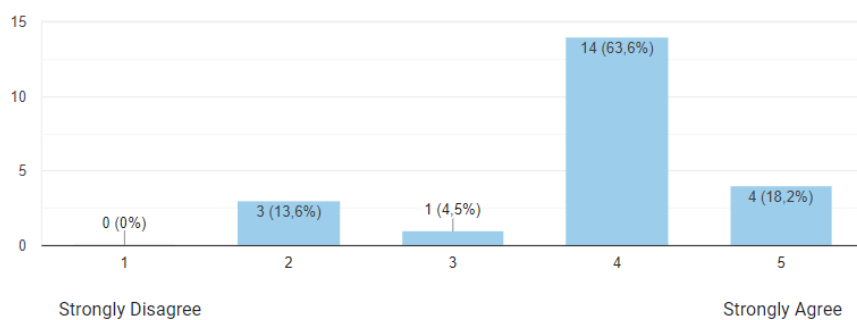
(a)

I found the system unnecessarily complex.



(b)

I thought the system was easy to use.

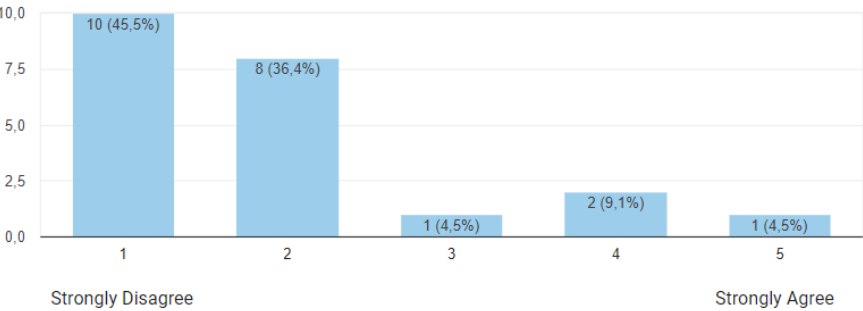


(c)

Figure D.3: Usability section 1 (SUS) results, Part 1.

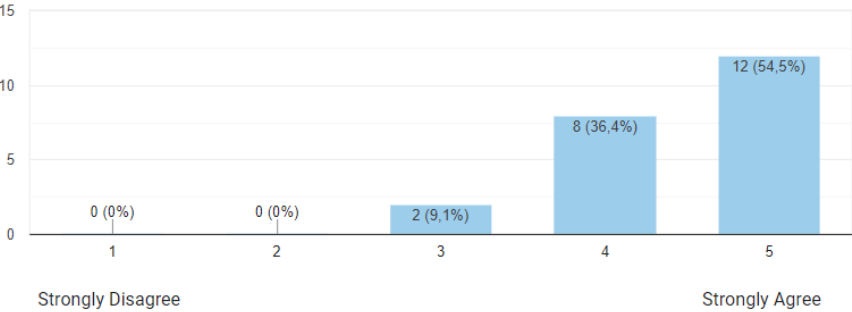
APPENDIX D. EVALUATION RESULTS

I think that I would need the support of a technical person to be able to use this system.



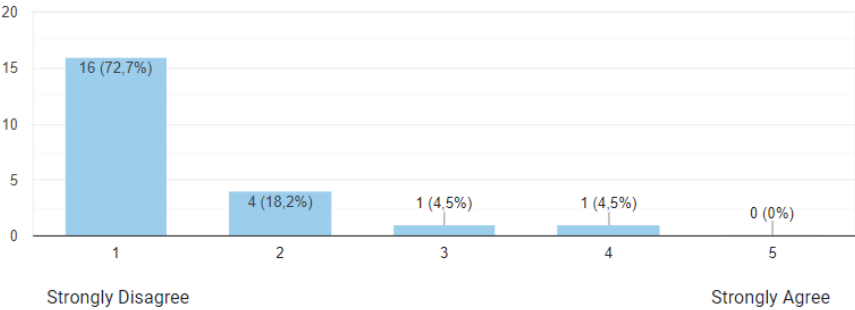
(a)

I found the various functions in this system were well integrated.



(b)

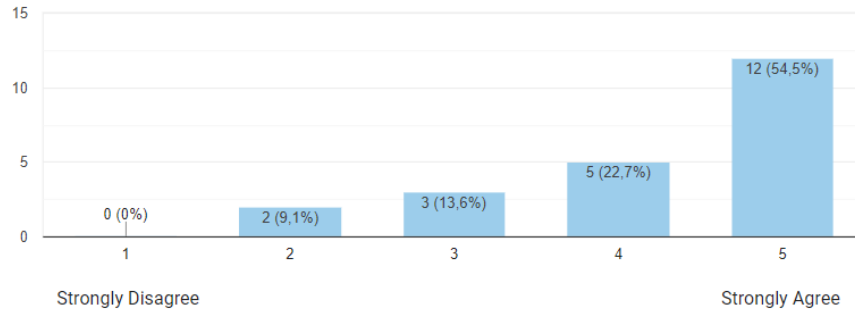
I thought there was too much inconsistency in this system.



(c)

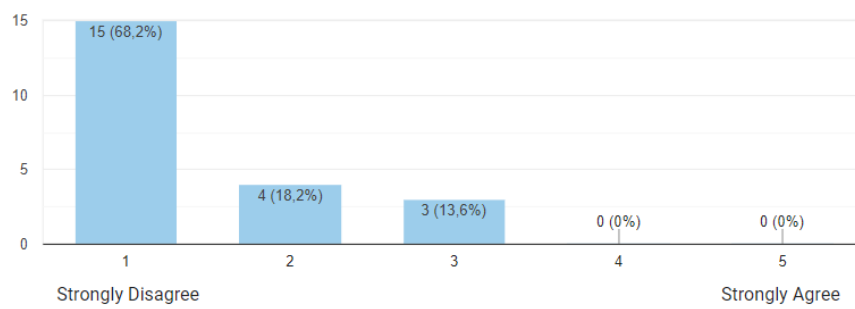
Figure D.4: Usability section 1 (SUS) results, Part 2.

I would imagine that most people would learn to use this system very quickly.



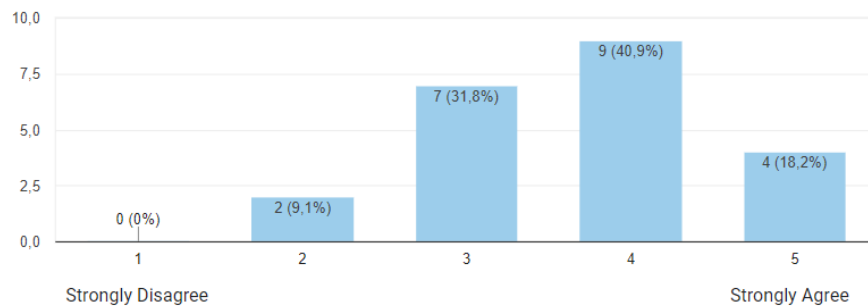
(a)

I found the system very cumbersome to use.



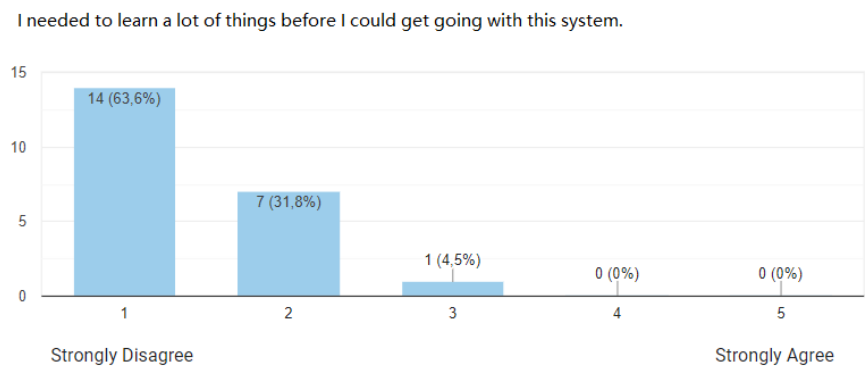
(b)

I felt very confident using the system.



(c)

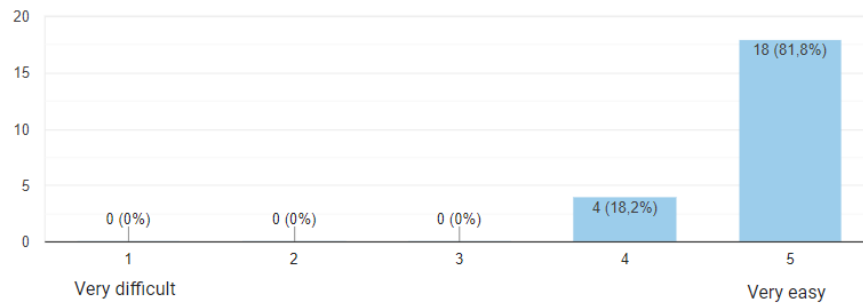
Figure D.5: Usability section 1 (SUS) results, Part 3.



(a)

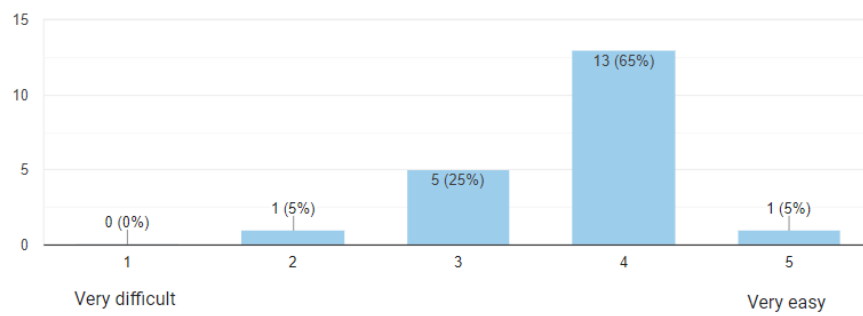
Figure D.6: Usability section 1 (SUS) results, Part 4.

On a scale of 1 to 5, how easy was to import a 3D model of an artefact?



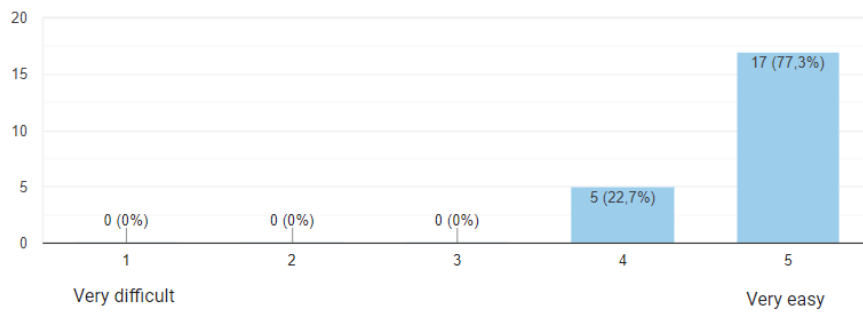
(a)

On a scale of 1 to 5, how easy was to interact with the imported artefacts?



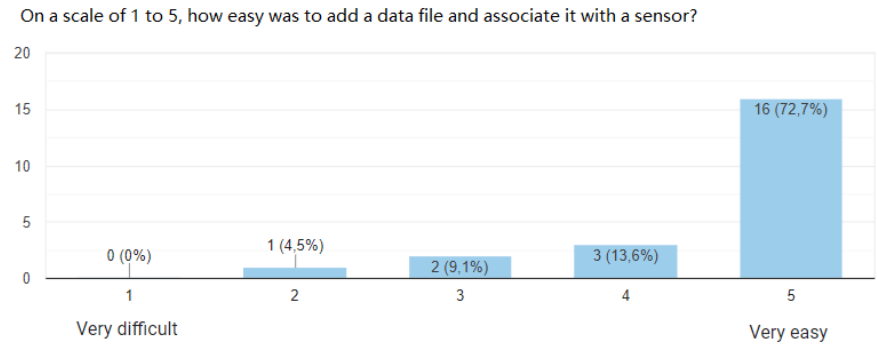
(b)

On a scale of 1 to 5, how easy was to add a sensor?

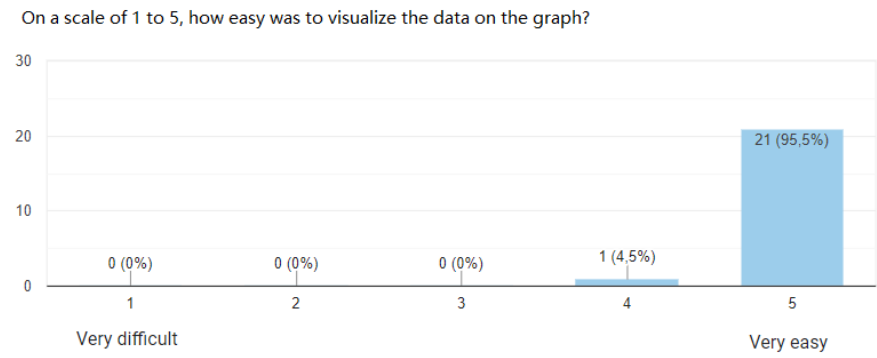


(c)

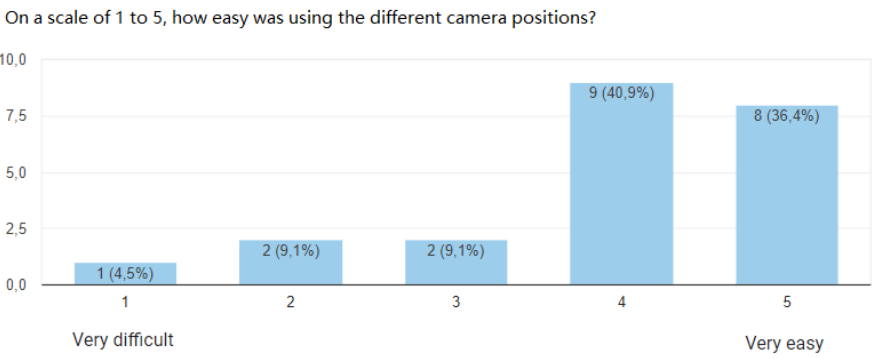
Figure D.7: Usability section 2 results, Part 1.



(a)



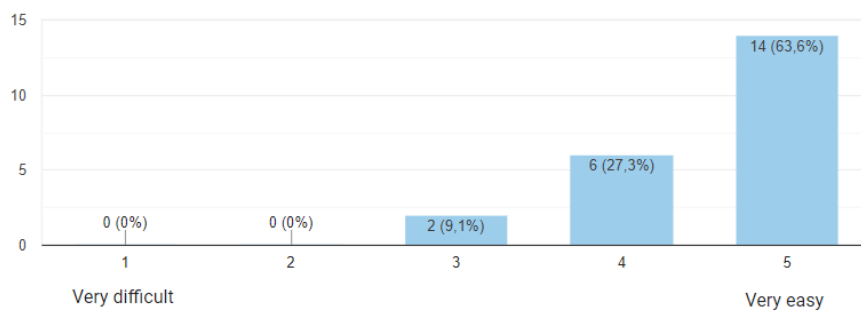
(b)



(c)

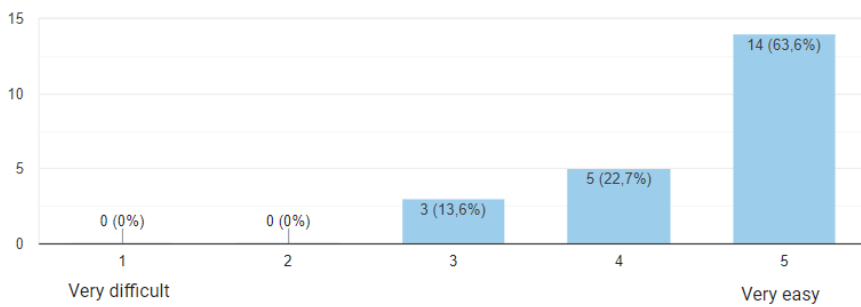
Figure D.8: Usability section 2 results, Part 2.

On a scale of 1 to 5, how easy was using the camera without restrictions (free-camera mode)?



(a)

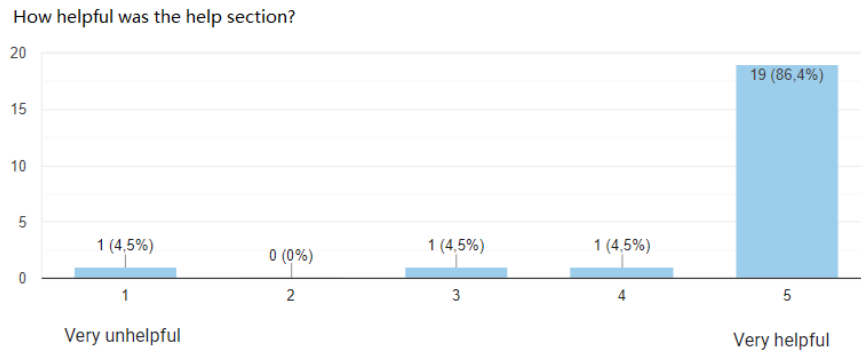
On a scale of 1 to 5, how easy was to Save and Load the progress?



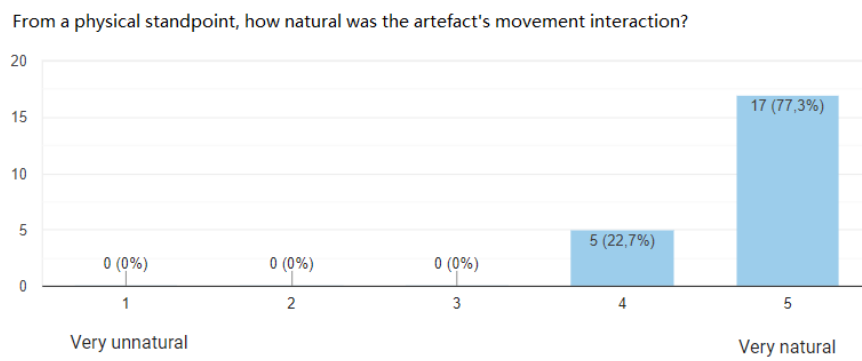
(b)

Figure D.9: Usability section 2 results, Part 3.

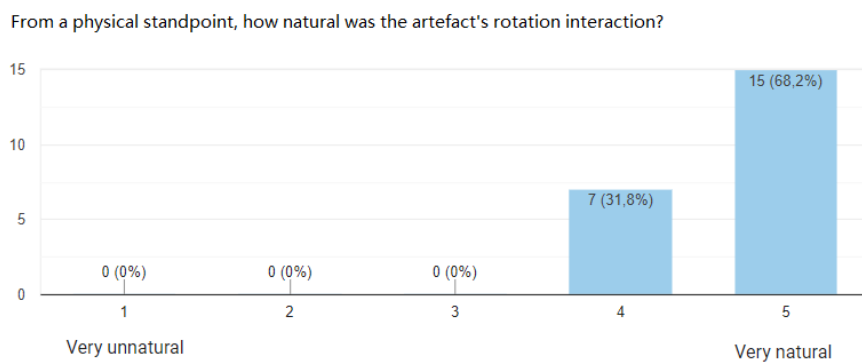
APPENDIX D. EVALUATION RESULTS



(a)



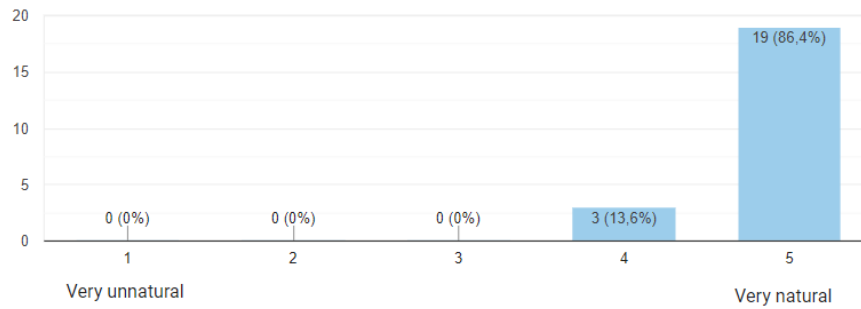
(b)



(c)

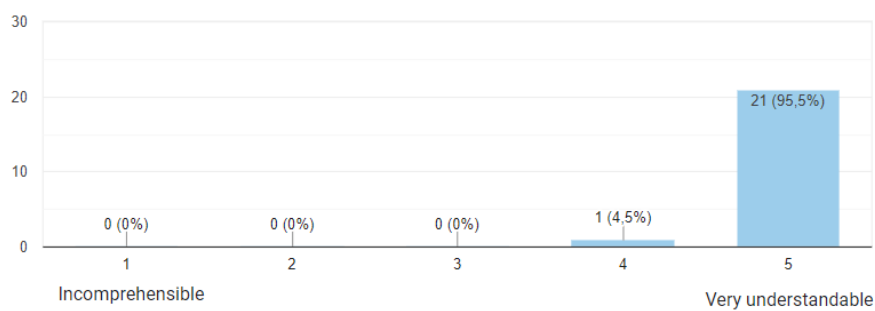
Figure D.10: Usability section 3 results, Part 1.

From a physical standpoint, how natural was the artefact's scaling interaction?



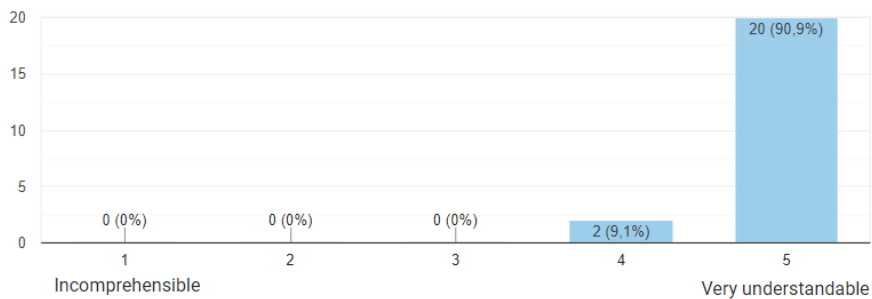
(a)

From a scale of 1 to 5, how understandable was the content of the graph visualizations with only 1 data variable?



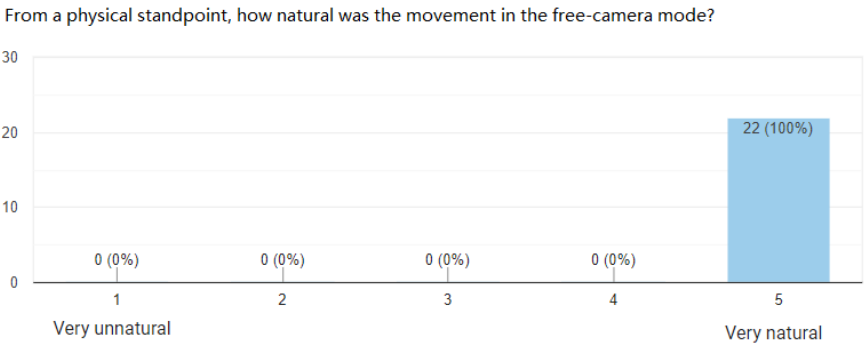
(b)

From a scale of 1 to 5, how understandable was the content of the graph visualization with 2 data variables?



(c)

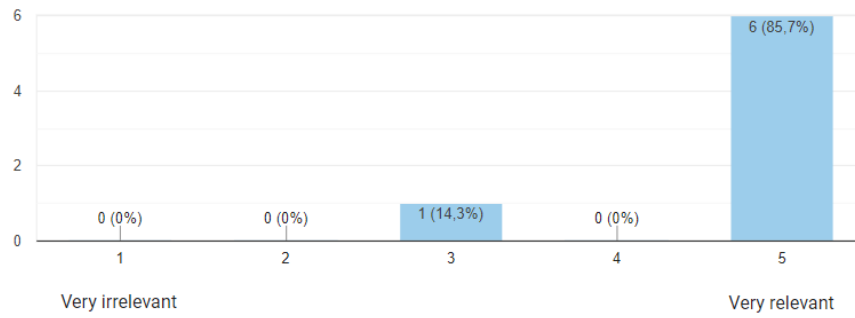
Figure D.11: Usability section 3 results, Part 2.



(a)

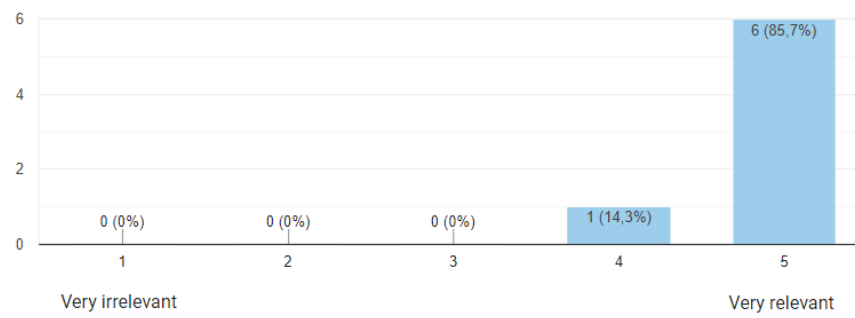
Figure D.12: Usability section 3 results, Part 3.

How relevant is the feature of importing the OBJ models of 3D artefacts?



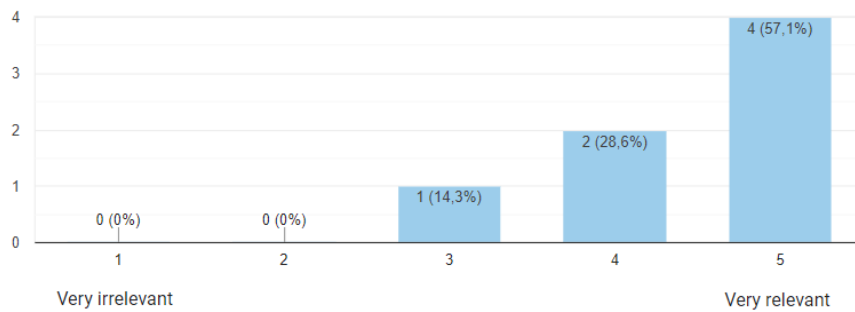
(a)

How relevant is the feature of moving the artefacts?



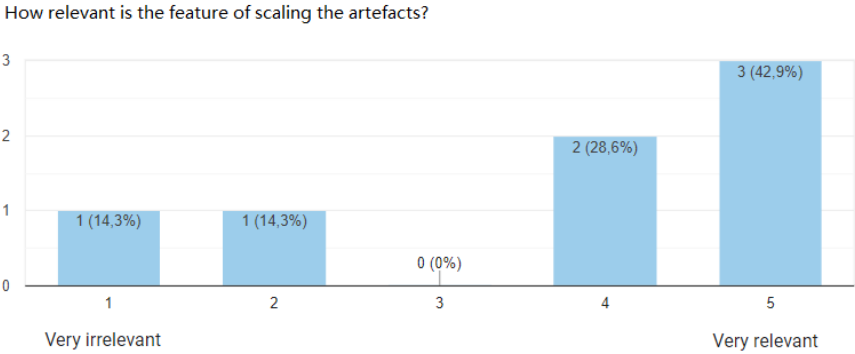
(b)

How relevant is the feature of rotating the artefacts?

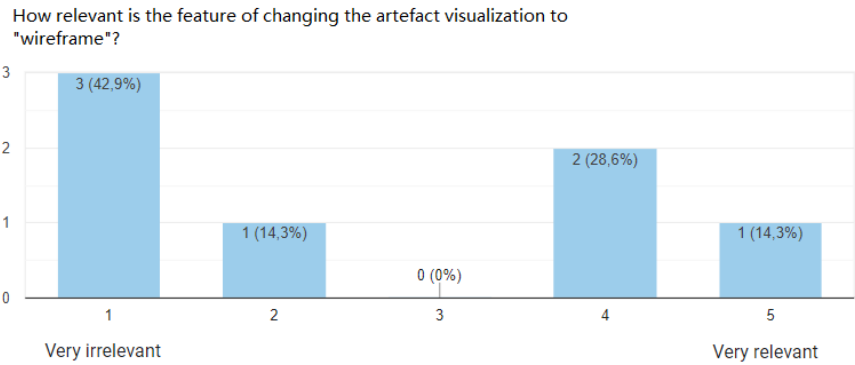


(c)

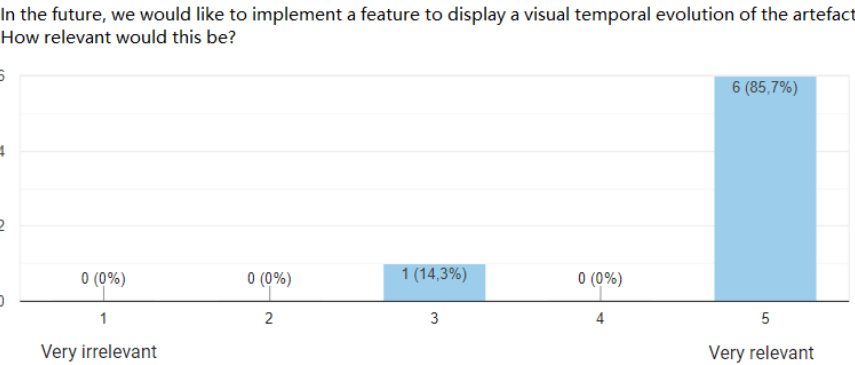
Figure D.13: Specialist's results, Part 1.



(a)



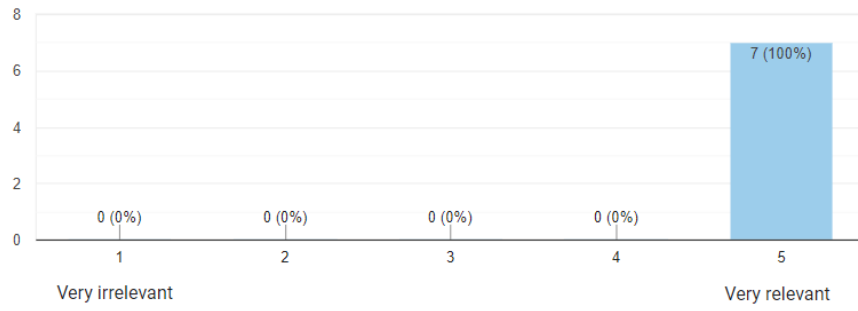
(b)



(c)

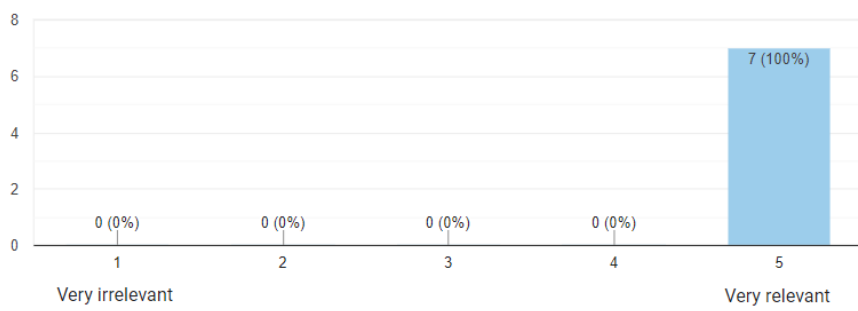
Figure D.14: Specialist's results, Part 2.

How relevant is the feature of adding sensors to the environment?



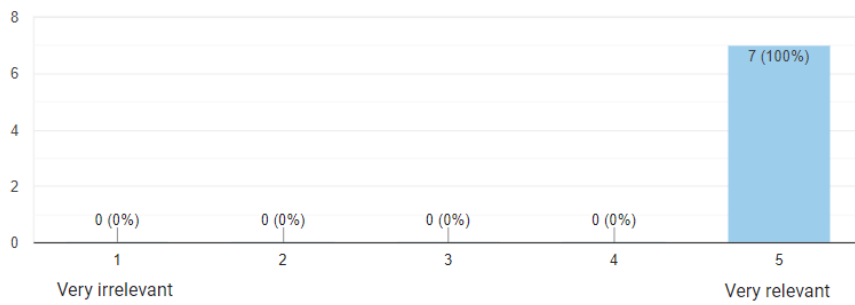
(a)

How relevant is the feature of associating data files with sensors?



(b)

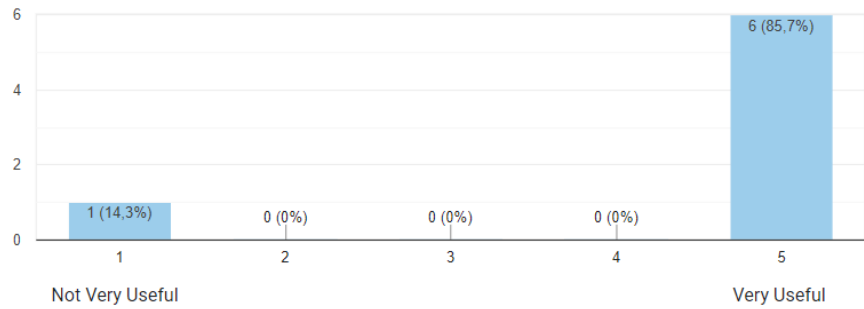
How relevant is the feature of visualizing multiple sensor data in a graph?



(c)

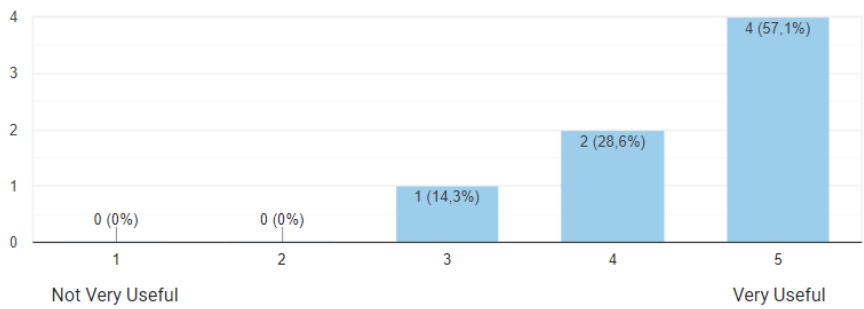
Figure D.15: Specialist's results, Part 3.

How useful is the ability to choose from the provided fixed camera positions to better visualize the application content?



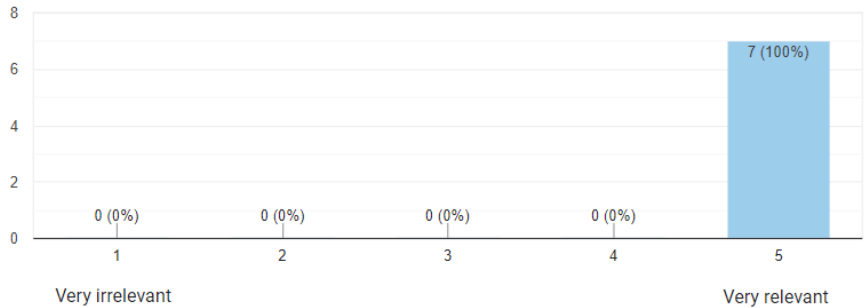
(a)

How useful is the ability to freely change the camera's position using the provided unlimited camera movement (free-mode) to better visualize the application content?



(b)

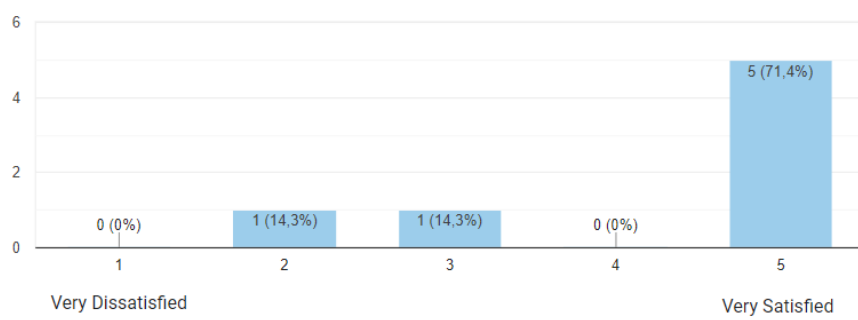
How relevant is the feature of saving and loading the session progress?



(c)

Figure D.16: Specialist's results, Part 4.

How satisfied are you with the features provided by this application?



(a)

Figure D.17: Specialist's results, Part 5.



2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing

2023 Digital Transformation 3D Printing and Manufacturing