



DIOGO NUNO DA GAMA BRÁS

Bachelor in Computer Science and Informatics Engineering

WEB OF STORIES

**STORIES AND GAMES WITH MIXED REALITY
ON MOBILE DEVICES**

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
June, 2024

WEB OF STORIES

STORIES AND GAMES WITH MIXED REALITY
ON MOBILE DEVICES

DIOGO NUNO DA GAMA BRÁS

Bachelor in Computer Science and Informatics Engineering

Adviser: Nuno Manuel Robalo Correia

Full Professor, NOVA University Lisbon

Co-adviser: Teresa Isabel Lopes Romão

Associate Professor, NOVA University Lisbon

Examination Committee

Chair: Nuno Manuel Ribeiro Perguiça

Full Professor, NOVA University Lisbon

Rapporteur: António Fernando Castro Coelho

Associate Professor, University of Porto

Adviser: Nuno Manuel Robalo Correia

Full Professor, NOVA University Lisbon

Web of Stories

Stories and games with mixed reality on mobile devices

Copyright © Diogo Nuno da Gama Brás, 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.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisors, Professor Nuno Correia and Professor Teresa Romão, whose expertise, patience, and guidance were invaluable throughout this thesis. Their availability, feedback, and support helped maintain this thesis on track, and I am deeply grateful for their mentorship.

I am also grateful to all other professors and fellow colleagues from the department of computer science at NOVA SST who helped me grow and enrich my knowledge. Special thanks to Duarte Guerreiro, Gabriel Marques, João Palmeiro, and Tiago Mendes, who were the most present during the thesis, for their friendship, comradeship, constructive criticism, suggestions, and overall mutual assistance. I want to additionally thank everyone who volunteered to take part in the user tests.

Last but definitely not least, I would like to thank my mother for all the support she has given, always prioritizing my education by being permissive and giving me full accountability for how I allocate my time.

ABSTRACT

Cultural heritage includes all the cultural assets, traditions, and knowledge of a group of people, collected and inherited over generations. It is up to those in the present to preserve and transmit their communities' tangible and intangible assets. With the advancements in technologies such as augmented reality, systems focused on perpetuating cultural heritage while providing new levels of accessibility to the public have emerged.

As a way to progress towards improving the region of Trafaria in Portugal and avoiding its possible gentrification, a gamified augmented reality storytelling platform has been developed to help preserve and transmit Trafaria's rich cultural heritage. The platform revolves around located interactive narratives which users can create and follow while being rewarded. The objective of this work focuses on the overall expansion of this platform by exploring features that could draw the attention of users and maintain their engagement. These features also aim to offer users a fun and effective learning experience by weaving in some game characteristics into augmented reality activities that incorporate cultural information. The features are further unified through a mechanism that seeks to offer a rewarding overlaying storyline.

User studies were conducted to evaluate whether the implemented features fulfilled their objectives by assessing various usability and experience aspects. The results were positive and also revealed areas with improvement potential.

The solution tries to ensure the preservation of the cultural legacy of Trafaria but it is also generalized and can be applied to any other region. It also tries to contribute as a great educational resource, encouraging its users to seek and discover more information both within and outside the platform in order to develop knowledge on other communities' cultures.

Keywords: Cultural Heritage, Augmented Reality, Gamification, Storytelling

RESUMO

O património cultural inclui todos os bens culturais, tradições e conhecimentos de um grupo de pessoas que são coletados e herdados ao longo de gerações. Cabe àqueles no presente preservar e transmitir estes bens tangíveis e intangíveis das suas comunidades. Com os avanços em tecnologias como realidade aumentada, surgiram sistemas focados em perpetuar o património cultural, proporcionando novos níveis de acessibilidade ao público.

Como forma de progredir no sentido de melhorar a região da Trafaria em Portugal e evitar a sua possível gentrificação, foi desenvolvida uma plataforma gamificada de narrativas e realidade aumentada para ajudar a preservar e transmitir o rico património cultural da Trafaria. A plataforma gira em torno de narrativas interativas localizadas que os utilizadores podem criar e seguir enquanto são recompensados. O objetivo deste trabalho centra-se na expansão global desta plataforma, explorando funcionalidades que possam atrair a atenção dos utilizadores e manter o seu envolvimento. Estas funcionalidades visam também proporcionar aos utilizadores uma experiência de aprendizagem divertida e eficaz, integrando algumas características de jogo nas atividades de realidade aumentada que incorporam informação cultural. As funcionalidades são ainda unificadas através de um mecanismo que procura oferecer uma história sobreposta.

Foram realizados estudos com utilizadores para avaliar se as funcionalidades implementadas cumpriram os seus objetivos, através da análise de vários aspetos de usabilidade e experiência. Os resultados foram positivos e também revelaram áreas com potencial de melhoria.

A solução procura garantir a preservação do legado cultural da Trafaria, mas também é generalizada e pode ser aplicada a qualquer outra região. Além disso, tenta contribuir como um bom recurso educacional, incentivando os utilizadores a procurar e descobrir mais informações tanto dentro como fora da plataforma, a fim de desenvolver conhecimentos sobre as culturas de outras comunidades.

Palavras-chave: Herança Cultural, Realidade Aumentada, Gamificação, Narrativas

CONTENTS

List of Figures	viii
List of Tables	x
Acronyms	xi
1 Introduction	1
1.1 Context	1
1.2 Motivation and Problem Definition	1
1.3 Solution	2
1.4 Contributions	3
1.5 Document Structure	3
2 Related Work	5
2.1 Digital Heritage	5
2.2 Augmented Reality (AR)	6
2.2.1 Technology	6
2.2.2 Mobile AR	9
2.2.3 Cultural Heritage AR	10
2.3 3D Modelling	11
2.3.1 Scanning Techniques	12
2.3.2 Heritage Recording	14
2.4 Gamification & Storytelling	15
2.4.1 Challenge Design	17
3 Solution Design	20
3.1 Case Study	20
3.2 Requirements	21
3.3 Feature Design	23
3.3.1 AR Interactions	23

3.3.2	Mission System	30
3.4	Technology and Architecture	34
4	Implementation Details	38
4.1	AR Interactions	38
4.1.1	Puzzles	38
4.1.2	Quiz	44
4.1.3	Balloon Pop	47
4.1.4	Geospatial Flyers	51
4.1.5	Miscellaneous	54
4.2	Mission System	57
4.2.1	Mission Graph Creation Tool	58
4.2.2	Mission Graph Interpreter	61
4.2.3	Interface	62
5	Evaluation and Results	65
5.1	Evaluation Methods	65
5.1.1	Protocol	65
5.2	Main Usability Tests	68
5.2.1	Participants Demographic Profile	69
5.2.2	Tests and Results Analysis	70
5.3	Tool Usability Tests	84
5.3.1	Participants Demographic Profile	84
5.3.2	Tests and Results Analysis	85
6	Conclusions and Future Work	89
6.1	Conclusions	89
6.2	Future Work	90
	Bibliography	92
	Appendices	
A	Usability Tests Consent Form	99
B	User Experience Questionnaire (UEQ) Results	100
C	Created Flyers	101
	Annexes	
I	System Usability Scale (SUS)	106
II	User Experience Questionnaire (UEQ)	107

LIST OF FIGURES

2.1	Types of Augmented Reality Display Technologies - In [33]	7
2.2	AR using SLAM on the left and Occlusion-Aware AR on the right - In [20] .	8
2.3	Mobile computational system from [43] on the left and Google Glass on the right	10
2.4	Drone survey - Photogrammetry on the left and LIDAR Laser Scanner on the right - Image Source	13
2.5	General overview of a data fusion pipeline using Structure from Motion (SfM) technique - In [50]	15
2.6	Organic mission graph layout on the left and Mission adaptation rules on the right - In [16]	19
3.1	Base platform's Entity Relationship Diagram	21
3.2	AR Jigsaw Puzzle Design Drawing	25
3.3	AR Sliding Puzzle Design Drawing	25
3.4	AR Quiz Design Preview	26
3.5	AR Balloon Pop Design Drawing	28
3.6	First Iteration of Flyers' Cover and Interior Design	29
3.7	Pokemon GO Missions Interface Components	33
3.8	3D View of Trafaria inside Unity's Editor	36
3.9	High-level Representation of the System's Architecture	37
4.1	AR Jigsaw Puzzle Interaction	42
4.2	AR Sliding Puzzle Interaction and Final Reward	43
4.3	AR Quiz Interaction - distant view	47
4.4	AR Balloon Pop Interaction	51
4.5	3D flyer model created, animated, and UV mapped using Blender	52
4.6	AR Geospatial Flyers Interaction	55
4.7	Information and Instructions Foldouts - Puzzle / Quiz / Balloon Pop	56
4.8	Tasks Manager Interface	58
4.9	Task Creation Interface - filled out with an example task	60

4.10	Reward Creation Interface - filled out with an example reward	60
4.11	Mission Graph Creation Interface - filled out with an example questline . .	61
4.12	Missions Interface Pages	64
5.1	Participants' Demographic Profile Graphs - Main	69
5.2	AR Interactions Preference	80
5.3	SUS Questionnaire Scores	83
5.4	User Experience Questionnaire Benchmark	84
5.6	SUS Questionnaire Scores - Tool	88

LIST OF TABLES

4.1	Puzzle pieces' UV coordinates positioning	40
5.1	Section S1 Questionnaire Results	71
5.2	Section S2 Questionnaire Results	73
5.3	Section S3 Questionnaire Results	75
5.4	Section S4 Questionnaire Results	77
5.5	Section S5 Questionnaire Results	78
5.6	Section S6 Questionnaire Results	80
5.7	Section S7 Questionnaire Results	81
5.8	Main SUS Questionnaire Answers - Shortened Affirmations	82
5.9	User Experience Questionnaire Scales Results	83
5.10	Participants' Demographic Profile Tables - Tool	84
5.11	Tool SUS Questionnaire Answers - Shortened Affirmations	88

ACRONYMS

API	application programming interface (<i>pp.</i> 35–37 , 53)
AR	augmented reality (<i>pp.</i> 2 , 6–11 , 22–24 , 26 , 27 , 29 , 30 , 35 , 36 , 38 , 44–47 , 51 , 65 , 67–69 , 71 , 79 , 82 , 91)
GUI	Graphical User Interface (<i>pp.</i> 31 , 35)
IAT	Trafaria Institute of Art and Technology (<i>pp.</i> 1 , 20)
POI	point of interest (<i>pp.</i> 20 , 21 , 23 , 27–29 , 51–53)
SDK	software development kit (<i>pp.</i> 7 , 8)
URP	Universal Render Pipeline (<i>p.</i> 24)

INTRODUCTION

This chapter clarifies the context in which this work is embedded, the motivation and the problem that underlie it, and concludes with a brief presentation of the solution and its expected contributions.

1.1 Context

This work is linked with an initiative led by T-Factor¹, a European project that aims to inspire innovative approaches to urban regeneration while encouraging co-creation and avoiding potential gentrification. One of their programs intends to grow the Trafaria neighborhood into a cultural hub. This includes the conversion of the Fort Nossa Senhora da Saúde da Trafaria into the [Trafaria Institute of Art and Technology \(IAT\)](#)².

In collaboration with NOVA University³, as a way to progress towards improving Trafaria as a place, a augmented reality storytelling platform has been created to help preserve Trafaria's cultural heritage. The platform currently offers users the possibility to create and follow point-by-point on-site narratives, providing a range of interaction types. This work focused on the overall improvement and expansion of this platform by exploring various other possible forms of interaction and, most importantly, finding a way to globally attach narratives making them resemble a cohesive storyline.

1.2 Motivation and Problem Definition

By cultural heritage, we denote the passing on of a non-natural asset to future generations, whether it is tangible such as monuments, or intangible such as traditions. Therefore, it is up to those in the present to preserve and transmit their communities' culture so it is not lost. We are all forced heirs of the culture that is presented to us, whether it is incorporated by the community where we were born or it is later recognized or even adopted, from other adventitious communities. However, culture evolves, making it inevitable that part

¹T-Factor website

²More about the Institute of Art and Technology

³NOVA University website

of these assets are lost over time, causing the necessity to use technological means to document or regenerate them.

With the transformation of Trafaria's neighborhood into a cultural hub, some problems might arise. The location could start to attract investors and higher income residents causing a noticeable demographic change, commonly labeled as gentrification. Although gentrification can lower the exposure to poverty for all residents by creating a mixed income neighborhood [11], it can also severely reduce the quality of life experienced by previous residents [12]. To avoid gentrification, it is important to increase local participation and collaboration with the present community during the transformation as well as subsequently guarantee improvement of their access to resources and opportunities [54].

As new assets start to emerge and the community evolves, it is inevitable that its culture suffers changes and some of its heritage might even be lost forever. This gives rise to the opportunity of creating a platform that can manage the documentation and exposure of a communities' cultural heritage and stories while encouraging their own cooperation so others can experience the communities' background.

1.3 Solution

The main objective of this work was to ensure the preservation of the cultural heritage of Trafaria, as well as to serve as an excellent educational resource by inspiring its users to seek out and discover more information both inside and outside the platform in order to gain knowledge about the cultures of other communities. With this in mind, mechanisms to increase users' interest and the platform's compelling nature were implemented.

This work started by including new patterns of interaction that made use of [augmented reality \(AR\)](#) and gamification in order to keep users motivated and engaged with the platform. Some of these were also made to be customizable and easy to incorporate into a user's narrative while still providing an interesting experience for the user who is interacting with them. Puzzles, quizzes, and mini-games were the base examples used for the types of patterns that were added, as they are simple to generate when creating a narrative while also allowing the introduction of cultural information into them, with puzzles only requiring images and quizzes only requiring text prompts.

Another overcome challenge involved finding a way to globally attach these interactions as if they were part of a whole overlaying storyline. This was achieved by implementing a mission system based on graphs that also used content unlocking patterns to increase user motivation. Each node of the graphs represents a mission, which contains various tasks related to the aforementioned interactions and rewards the users upon completing them. In order to create these graphs, a built-in tool was also developed.

Finally, the mission system and interactions that have the possibility of being difficult to scale use methods to try and adapt to the users' abilities as a means to further increase their satisfaction. Some other minor work was also done, including the usability testing of all implemented features.

1.4 Contributions

The solution developed considering the aforementioned problem produced the following contributions:

- **Cultural Heritage Preservation Platform** - Advancement on a platform that allows users to create and follow interactive narratives concerning the community and heritage of Trafaria and possibly other locations. Idealizing the preservation, maintenance, and documentation of cultural heritage while taking advantage of technological advances.
- **Gamified Storytelling with Augmented Reality** - Exploration of augmented reality appliances in storytelling and ways to unify user's narratives, by adding game like features, as a way to improve the experience of following these narratives, giving the feel of a cohesive storyline.
- **Heritage Digitalization** - Research and usage of data capture and 3D modelling methodologies, contributing to the documentation of heritage through the creation of its digital representation.
- **Mission System Tool for Unity** - Development of a built-in Unity tool that allows users to easily incorporate missions into their games by using various interfaces and writing minimal code.
- **Result Analysis** - The analysis of all the evaluations and tests carried out during and after the development of the platform will be useful for future projects that intend to take advantage of the same ideas and technologies.

1.5 Document Structure

The structure of this document is composed of the following six chapters:

- **Chapter 1** - The first chapter presents the context of this work, the problem and motivation behind it, and a summary of the solution along its possible contributions.
- **Chapter 2** - The second chapter contains a study overview of the state of the art, technologies and useful case studies required for the development of this work. These include augmented reality, especially on mobile devices, 3D modelling and the creation of games and stories, all while considering their possible connection with cultural heritage preservation.

- Chapter 3 - The third chapter gives a brief overview of the case study, followed by the requirements of the solution. It then goes on to breakdown the design of features that were going to fulfill the requirements and ends with a description of the used technologies and the system's architecture.
- Chapter 4 - The fourth chapter proceeds to explain in detail the implementation process of the various features described in the previous chapter.
- Chapter 5 - The fifth chapter starts by describing the evaluation methods used, along with some studies that helped to shape them. After that, the results of the usability tests performed are presented, followed by their analysis and discussion.
- Chapter 6 - The sixth chapter reflects on the developed work during this dissertation and finalizes by summarizing some future investigations that may arise from it.

RELATED WORK

This chapter contains a study overview of the state of the art, technologies and useful case studies required for the development of this work. These include augmented reality, especially on mobile devices, 3D modelling and the creation of games and stories, all while considering their possible connection with cultural heritage preservation.

2.1 Digital Heritage

Cultural heritage includes all the cultural assets, traditions, and knowledge of a group of people, collected and inherited over generations. This can include physical, and therefore tangible, objects and artifacts, such as historic buildings, monuments, and works of art, as well as intangible cultural expressions, such as traditional music, dance, and stories. Cultural heritage can provide a sense of identity, continuity, and belonging for the concerned communities and is therefore considered to be very valuable for a country. However, sites all over the world are facing some sort of deterioration in the interest of aggressive urban expansion and sometimes even just broad neglect. Nonetheless, there are various ways to protect and conserve cultural heritage, starting with national and international laws, dedicated programs, and educational initiatives.

UNESCO¹ carries a key dedicated program, denominated the *World Heritage Site Program*, which was established in 1972 and is an international initiative that seeks to identify and help protect cultural and natural heritage sites, preserving them for future generations. The program starts by designating and voting on sites based on their universal value and significance, such as their technological innovation or scientific interest, historical events or purely their exceptional beauty. Once a site is considered worthy, it joins the over 1100 other world heritage sites scattered around 167 countries and becomes qualified to receive technical and financial support when required.

With the great advancements in 3D documentation technologies, from electronic surveying instruments to laser scanners, photogrammetric cameras, and even CAD modellers,

¹[United Nations Educational, Scientific and Cultural Organization website](#)

there has been an increasing amount of heritage data brought into the digital world, developing the notion of digital heritage. Some of these 3D technologies are further described in section 2.3.

The concept of digital heritage is recognized by UNESCO¹, which defines it as any digital material of human knowledge or expression deemed significant enough to be passed on to future generations. This includes content created directly on, or converted to, a digital platform [56].

Some might not find the idea of digitizing, documenting, and possibly reconstructing some tangible assets all that appealing, because, in their opinion, although these procedures help preserve and immortalize these assets, they also completely destroy the uniqueness aura and essence of the original. Others argue that this is not a concern since even the most flawless duplicate would never have the traces of existence, in time and space, of the original, since the first can only be revealed chemically or physically and the second can be traced from previous ownership [32].

2.2 Augmented Reality (AR)

AR technology aims to improve people's perceptions and experiences of the real world by overlaying digital information on top of the physical environment. Unlike virtual reality (VR), which completely emerges people in a computer generated digital environment, AR allows for simultaneous interaction with both digital and physical elements in real-time, providing innovative ways to access information. The digital content is usually aligned with the physical world in such a way that it appears to be a natural part of the environment [3].

AR can be experienced in 3 different groups of ways, either over a device which provides a display and a camera, over a head mounted optical or video see-through display or over light projection into the real environment, these methods can be better seen in (fig 2.1). D. van Krevelen and R. Poelman in [26], present exhaustive research on several methods for producing AR and existing devices for experiencing AR, including early-stage prototype approaches. They also state their individual limitations and possible applications.

2.2.1 Technology

There are several types of approaches to generate AR experiences and, although there does not exist a concrete definition on those types, they are most commonly divided in two categories: marker-based AR and markerless AR.

Marker-based AR: This form of AR, sometimes also referred to as image recognition AR, instantiates the AR experience based on the location of specific detected markers, such as images, QR codes, or 3D objects. The AR device, which needs to be equipped with a

camera, searches for and captures the position and orientation of an expected marker and, after snatching that information, displays the digital content around it, as planned.

Markerless AR: This form of AR does not require markers to instantiate the AR experience. It instead uses sensors, such as the accelerometer, GPS², or gyroscope to determine the position and orientation of the AR device, making it able to correctly place the AR content in the environment. A combination of the information from location and mapping, denominated (SLAM)³ [27] is a commonly used algorithm, which is widely adopted by most AR software development kit (SDK)s.

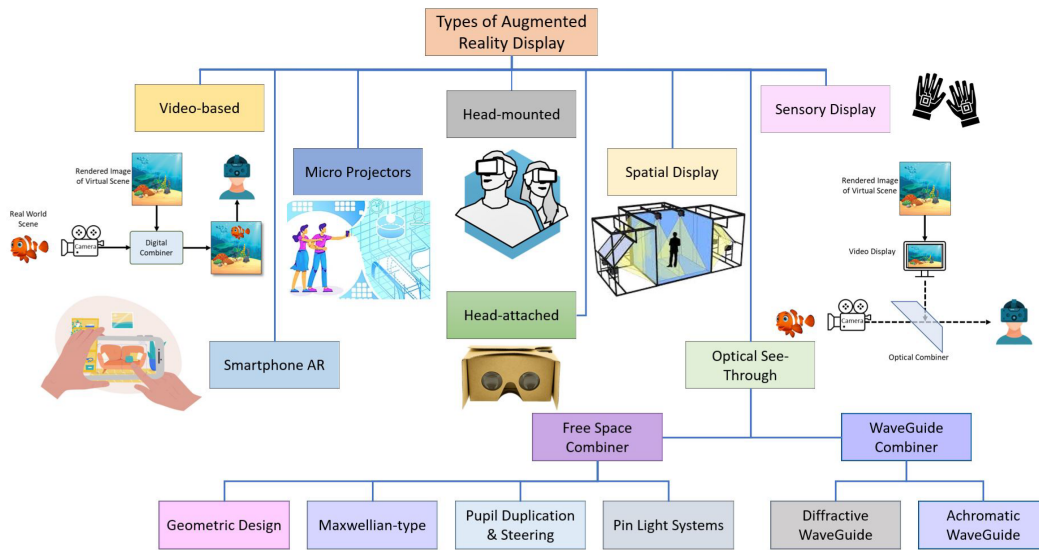


Figure 2.1: Types of Augmented Reality Display Technologies - In [33]

Predominant sub types of these forms of AR include:

Projection-based AR which, like implied by the name, projects the digital content onto a physical surface of the environment [37]. It can also provide a interactive experience, by detecting necessary user information, using the aforementioned SLAM³ algorithm [27];

Location-based AR which, by gathering information of the real world position and orientation of both the digital content and the AR device, is able to correctly trigger the AR experience when these two come within a defined range. One of the most popular examples of location-based AR is the Pokemon GO⁴ mobile game;

Superimposition AR, also referred to as overlay AR, which is able to modify the view of an existent physical object. Both the environment and non augmented object detection is essential to create this experience. It is most commonly used to allow the review of different

²Global Positioning System

³Simultaneous Localization and Mapping

⁴Pokemon GO website

colours of a certain product, as well as, to visualize their placement in the environment. It can also be used to digitally visualize a [AR](#) completion of a ruined monument, for example;

Contour-based AR, also referred to as outlining AR, which identifies and highlights the silhouette of objects and scenery that wouldn't usually be recognized by the human eye. It is most regularly used to assist in safe driving in low-visibility situations, consequently needing to resort to the use of special cameras.

More advanced designs of these forms of [AR](#) are also capable of capturing information about the surrounding environment, allowing them to better align the digital content within it. This includes the detection of vertical and horizontal surfaces, as well as the position and orientation of physical objects and their relative depth to the [AR](#) device. This information enables the possible placement of content with higher accuracy on the three dimensional axes, even partially behind physical objects as can be seen in (fig. 2.2). It is also possible to capture the light levels around the environment and generate realistic shadowing on the digital overlaid content [14].

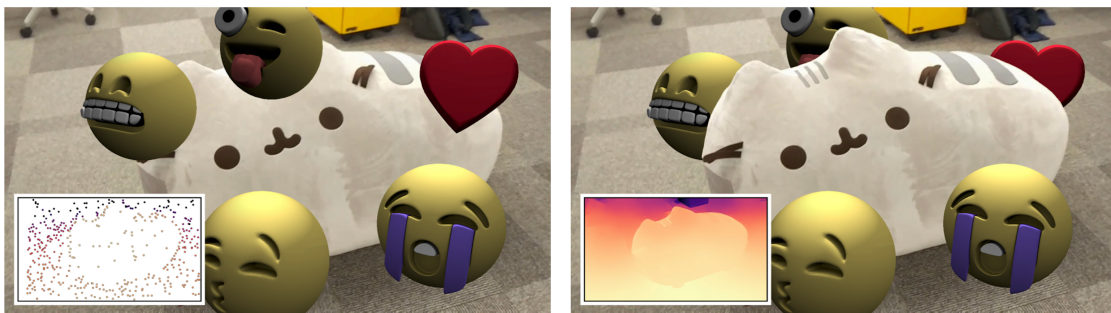


Figure 2.2: AR using SLAM on the left and Occlusion-Aware AR on the right - In [20]

J. Cheng et al. [10] developed an [AR](#) system with the intention of comparing marker-based AR with markerless AR, from both a technological and system developer perspectives. They concluded that:

At the time, [AR SDKs](#) had higher support for marker-based AR, thus being less complex to produce this type of [AR](#) experiences, over markerless AR;

The positional **accuracy** of content placement was relatively higher on marker-based AR. The most influential factor for the accuracy of marker-based AR was the lighting conditions of the environment, whereas the accuracy of markerless AR could be improved by incorporating more precise location technologies, such as RFID⁵, Wi-Fi⁶, or UWB⁷;

The **stability** of already-placed content was relatively higher on markerless AR. The most influential factor was, surprisingly, not the complexity of the [AR](#) content. On marker-based AR, it was the overall quality of the marker and [SDK](#) used, while on markerless AR,

⁵Radio Frequency Identification

⁶Wireless Fidelity

⁷Ultra Wideband

it was the location technology and gyroscopes of the device;

Marker-based AR is generally supported on desktops and mobile devices, while markerless AR is only supported on mobile devices. This is due to the lack of the necessary sensors on a desktop computer.

2.2.2 Mobile AR

As mentioned in the previous subsection, mobile devices have some advantages over static devices when it comes to perceiving an AR experience. They offer the possibility to freely move around the environment, and nowadays, when compared to desktop computers, they are also more likely to have the necessary sensors needed to execute markerless AR experiences.

Early attempts at developing mobile AR systems include examples such as, the location-based mobile AR application by G. Reitmay et al. in [43] and the AR mobile computational system by V. Vlahakis et al. in [57]. Although the development of these systems was considered successful, the hardware limitations were most commonly noted as the major target for possible future improvements.

The majority of these mobile AR systems were bulky, as they were composed of a large backpack with the PC, sensors, displays, batteries, and other essential equipment within it, as seen in (fig. 2.3). Likewise, due to their poor brightness, contrast, resolution, and field of view, optical and video see-through displays were typically inappropriate for outdoor use [26]. However, in order to solve this second problem, companies have started using and testing laser-powered displays, providing a new dimension in head-mounted and hand-held AR devices⁸.

Furthermore, recent advances in mobile technology, such as those found in smartphones, are closing the gap in facilitating mobile AR development. Most smartphones come equipped with cameras, accelerometers, gyroscopes, GPS and other crucial sensors for a more advanced AR experience. There are also a wider variety of more capable smartphone devices, mostly because of the rise of the mobile gaming industry [58], that provide impressive computational power and connectivity to wireless networks, which is also beneficial for mobile AR progression.

Even though smartphones seem like the perfect subject to implement mobile AR experiences on, they also introduce some drawbacks. The majority of these devices offer a relatively small display and lack the battery capacity to provide a proper immersive AR experience. It is also a complex problem in and of itself to develop the AR user interfaces for these cramped display devices. There are other technologies, such as Google Glass⁹, seen in (fig. 2.3), which can be used separately or in conjunction with these devices to increase immersiveness and get around their disadvantages.

⁸MicroVision's augmented reality laser powered microdisplay technology

⁹More about Google Glass



Figure 2.3: Mobile computational system from [43] on the left and Google Glass on the right

2.2.3 Cultural Heritage AR

There are numerous systems that implement different AR types and experiences across a wide range of possible applications. Their technology can be used in a variety of situations, such as education, gaming, e-commerce, tourism, and industrial settings. In e-commerce, it can be used to preview product placement or color before buying it, while in an industrial setting, it can be used to help assembly workers know precisely what pieces go where. This work focuses specifically on mobile AR systems associated with cultural heritage preservation, including education, gaming, and tourism.

There is an ample amount of research on the use of mobile AR systems in a cultural heritage setting. AR systems are key to increasing interest and capturing people's attention to interact with an application that is trying to provide an immersive experience for them to gain knowledge of other cultures' architecture, art, and customs.

V. Vlahakis et al. [57] give an overview of the ArchoGuide project, which involved rebuilding an Olympian Greece cultural heritage site. This project was a very early but successful attempt at using mobile computation systems in AR experiences related to the preservation of cultural heritage. It allowed visitors to explore and learn more about both the old structures and customs of the subject in question. They wore a see-through head-mounted display which overlaid the digital content while also carrying a computer to handle the necessary calculations. Although, at the time, the system was highly appreciated among the visitors, for offering a distinctive way to view the sites, by today's standards, this type of system is very hefty and cumbersome.

Z. Noh et al. [36] also conducted a relatively early study on the usage of AR in this scenario, developing a virtual heritage system. However, their main complication stemmed from poor registration and tracking technology, which required them to rely on

a great amount of markers in order to achieve acceptable alignment.

G. Lee et al. [28] developed a outdoor AR application called CityViewAR. The innovative function of the application, when compared to the aforementioned, was the ability to view 3D models of the buildings in AR, which were presented on a map together with their descriptions and relative images.

Keeping within the context of superimposing wrecked buildings using AR, C. Panou et al. [39, 40], present a location-based mobile AR system most similar to the one this work will focus on expanding. The system aims to enhance its users' experiences of interaction with cultural heritage sites, having as a case study, the historical part of Chania in Greece. The system works by having experts create points of interest on a map and placing AR scenes, additional information, and tasks on each one. Users can then access these geo-located points of interest in the real world and interact with the AR content. Gamification techniques were also used, with users earning points and climbing leaderboards for completing specific tasks, such as determining the historical period of monuments based on architectural features and other obtained clues.

They conclude by implying that interest in cultural heritage would increase if the platform were expanded to incorporate more interaction between the users, such as comments, likes, shares, and other forms of extra communication. They also add that increasing the gamification level would make an experience even more immersive and boost visitor involvement and engagement. This could be done by adding features like treasure hunts, scavenger hunts, location-aware storytelling, etc.

2.3 3D Modelling

The various existent techniques used to create 3D models [49], can be grouped into three main methods:

The **first method** consists of using 3D modelling software such as various Autodesk programs¹⁰, Blender¹¹ or SketchUp¹². This method is generally used to create models from scratch and requires a certain amount of knowledge and practise in modelling techniques and concepts. Even though some offer simplified aspects to make it easier for a common self taught user and others offer full courses on the entire functionality of the software, this is still considered a very time-consuming and professionalized method.

The **second method** consists of capturing real-world objects or scenes and extracting their shape and details to create their respective digital 3D model. This can be done with techniques such as light scanning, laser scanning, contact scanning or photogrammetry. These techniques have recently started becoming more easily available to the public and

¹⁰Autodesk website

¹¹Blender website

¹²SketchUp website

are a very useful tool. Depending on the models in question it can be way faster to produce content and does not need anywhere near the level of knowledge and experience required on the first method. The only drawback is the potentially reduced quality and amount of detail of the produced model which also depends on the technique used and its level of development.

The **third method** consists of using procedural techniques by applying algorithms and mathematical equations to create certain models, rather than doing them manually. This allows for the efficient generation of both simple and complex models, such as randomized structures or natural landscapes. Due to the increasingly growing demand for 3D content, and it not being satisfied by current methods, studies and implementations on how this can be achieved using AI technologies have emerged [4].

2.3.1 Scanning Techniques

Focusing on the second aforementioned method, 3D scanning techniques essentially gather the necessary information relative to a object or scenery, possibly including its surroundings, and are able to generate their digital copy into a 3D model file which can then be visualized, further modelled upon and 3D printed. For all 3D scanning techniques there is a common element. Hundreds of captures of the test subject must be taken from different views. These captures are then integrated in a reference alignment system, in order to obtain numerous meaningful projections, which are subsequently banded together to create the final digital model.

There are several criteria according to which 3D scanners can be categorised. By construction type, they can be divided into mobile or stationary, and by scanning method, they can be divided into laser, optical, ultrasound, x-ray, mechanical or destructive. Mobile scanners are commonly non-contact, meaning they do not physically touch the test subject, and though they are usually propped-up to stay anchored, they can also be hand-held. Their main advantage is that they can be easily brought to immovable objects. Stationary scanners, on the other hand, can resort to contact and are commonly larger and even though some can be dismantled and moved, the assembly can take a considerable amount of time [55]. Various scanners, usually mobile, can be grouped in order to create a large structure able to capture objects faster and with less hassle, such as 3D full-body scanners, which can, as an example, be used to create 3D avatars or figurines of a person¹³.

Laser based 3D scanners can be divided into two categories, those that use trigonometric triangulation and those that use time-of-flight (length of time the laser takes to reach an object and reflect back), the latter can be seen in (fig. 2.4). Triangulation scanners are quite common, therefore they are usually easy to use and have a comparatively low cost. They work by projecting multiple laser lines onto an object and capturing their

¹³Example of a 3D full-body scanner by: 3DFascination

reflection angles. Being generally better in irregular light scenarios and dealing with both dark and shiny surfaces. Time-of-flight scanners are commonly used for longer ranged scans being able to scan millions of points at a time. They work as described above, by measuring the time it takes for a certain laser line to reflect back to the scanner. There are pulse-based and phase-shift based time-of-flight scanner technologies. The first one being more suitable for very long range scenarios of more than one kilometer and the second one being more suitable for comparatively medium range scenarios of around three hundred meters.

Optical based 3D scanners have two prevalent techniques, projection of light and photogrammetry, the latter can be seen in (fig. 2.4). Projected light is a technique that works very similarly to trigonometric triangulation scanners, as the scanner projects light patterns into the object and then captures the deformities on said patterns. Scanners that use this technique are most commonly portable, fast and accurate and can be used to safely scan a living being with eyes, unlike laser based scanners. Photogrammetry is a technique that involves taking various overlapping photographs or a video sequence of an object, from multiple different angles, and creating the respective digital model by stitching up all imagery together based on coinciding pixels that represent the same physical position. Although in this technique the collection of data is faster, it takes a considerable time to process it and the end result's quality is heavily based on the capture resolution.

Contact based 3D scanners are composed of an articulated arm that runs a probe through the surface of an object, sampling enough points to accurately recreate the model in digital form. Since this technique involves physical contact with the object, which needs to be held in place, it isn't the most appropriate for fragile subjects. In return it is able to precisely scan reflective and transparent objects, in contrast to the other scanner types mentioned above.

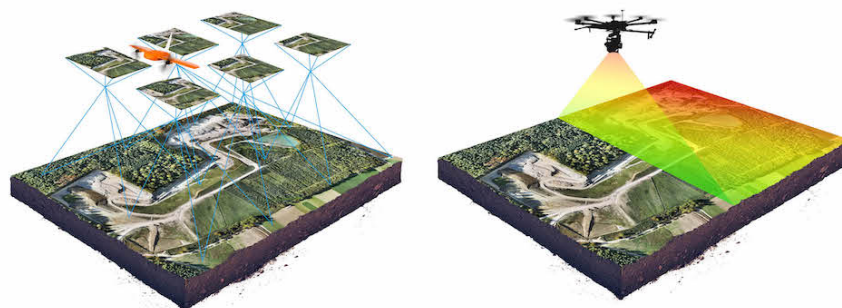


Figure 2.4: Drone survey - Photogrammetry on the left and LIDAR Laser Scanner on the right - [Image Source](#)

2.3.2 Heritage Recording

3D scanning techniques currently have a wide variety of applications, with ones prevailing over others in certain fields. These techniques can be used in medicine, to reconstruct or visualize parts of a human body, or in engineering, to run digital simulations with the scanned objects or to reverse engineer them. This work focuses on their usage for cultural heritage preservation.

There are numerous studies on the usage and respective appraisal of different 3D scanning techniques for land surveying and digitalization of cultural and historic monuments, sites and art pieces, even their reconstruction and possible creation of replicas.

M. Levoy et al. [29] introduced a system based on laser triangulation scanners for digitizing the shape and colour of large statues, having some of *Michelangelo's* work as case study. They explain the approaches to handling the extensive data sets of such detailed scans from the large statues and also on how to align and deal with the laser light scattering in the marble. They were able to capture all the minute details of the individual carvings and went as far as being able to speculate which specific chisel tools were used. The main defined problems, such as the overall cost, time spent scanning and protection of the statues, all derived from the scanning having to be done on field, as the statues were in public exhibition, fragile and very valuable.

Continuing with *Michelangelo's* statues as case study, A. Hattab et al. [19] show the purpose of using their novel method of differential 3D scanning by completing an unfinished statue masterpiece. Basically, this method simplifies reverse engineering problems and also allows to automatically update and segment a digital model based on real-time physical changes. This makes it so one can fluently alternate work subject between the virtual and physical models.

S. Singh et al. [52] used close-range photogrammetry, which is one of the most prevalent variants, to create a large scale city 3D model. A video sequence of all buildings was taken and subsequently, the physical position of the camera was added as information on some frames. They concluded that this technique proved to be fairly accurate, produced photorealistic content and was rather simple and cost-effective. The only pointed downside was the equipment requirement, as the final result highly depended on the resolution of the camera and the accuracy of measurements for its position, also the computer where the data was processed required some computational capability.

There are also studies on the combined use of techniques to achieve the best possible and complete results.

M. Rahrig and A. Luib [42] present a study on how this combination was achieved during the documentation of the *Temple of the Sacred Tooth* in Sri Lanka. A terrestrial 3D scanner, controlled by a computer, was used to capture the majority of the temple while the color information was being captured by a camera on top of the scanner. Additionally,

areas and objects of increased detail were captured by a hand-held structured light scanner. All gathered scans were then registered and merged using more than one specialized softwares. Other studies on different combinations of techniques [44, 50] show that this is a viable concept that reaps positive results, as it is possible to merge information gathered by distinct scanner types. One example can be seen in (fig. 2.5).

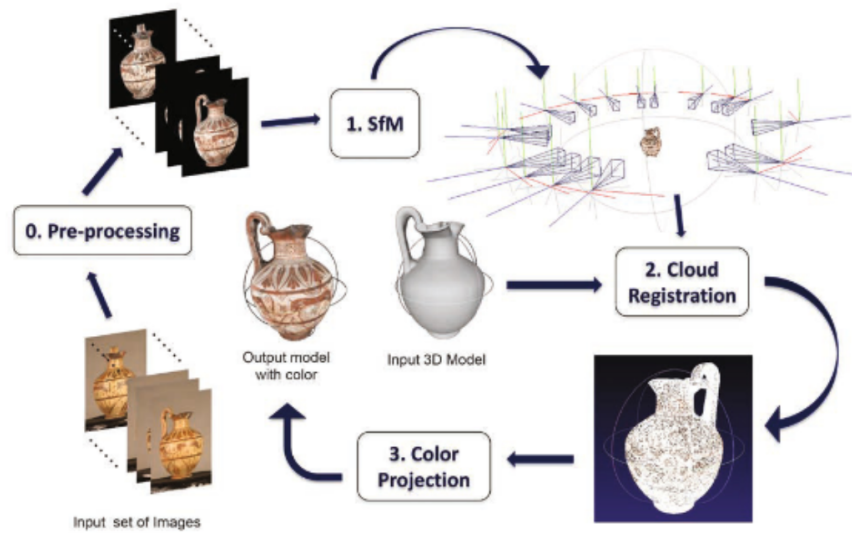


Figure 2.5: General overview of a data fusion pipeline using Structure from Motion (SfM) technique - In [50]

2.4 Gamification & Storytelling

Gamification is essentially the integration of game mechanics and aesthetics into applications that are not games by their nature. As video games gain significant market traction and expand from simple forms of entertainment to tools for education and training, this technique has gained increasingly more attention from academics [6]. The main goal of this technique is to offer users of gamified applications a fun and interactive way to accomplish the desired goals while also maintaining their engagement, motivation and satisfaction.

According to Deterding et al. [13], an application which uses gamification techniques is not to be confused with a serious game. While gamification simply refers to the integration of game-like aspects into systems that are not intended to be games, serious games are true games whose entertainment is not the primary goal.

The most common game design elements that are usually targeted to be added to an application are point systems with leaderboards, awards such as prizes, badges and achievements, and challenges that must be completed in order to gain these points and awards. Koivisto and Hamari [25] conducted a study on the individual effectiveness of these aspects of gamification on both psychological and behavioral levels. While obtaining

largely positive results, they also confirm that the aforementioned game elements are in fact the most common when gamifying an application. It is also specified that these results should not be considered universally, since they do not cover all possible contexts where gamification can be implemented. However, they continue to offer crucial guidance for choosing game design options. Examples of gamification integration in education [5] and tourism [34] show that, in these contexts, this technique is highly effective, having a positive outcome on increasing users engagement and overall results.

Storytelling is the act of creating and sharing a narrative or story with an audience, whether it be through oral or written traditions, or other forms of communication such as film, theater, or video games. The National Storytelling Network¹⁴ is an organization that spreads the word about the importance of storytelling and the connection it brings to communities. They summarize five descriptive aspects of storytelling in one phrase: *"Storytelling is the interactive art of using words and actions to reveal the elements and images of a story while encouraging the listener's imagination"*.

What defines a compelling story is quite arbitrary, thus existing several different models and sets of elements claiming to be the basis for storytelling. For example, N. Kashyap [23] describes the 3C's model, suggesting that character, conflict, and conclusion are the three main characteristics of an enticing story. Good characters are able to create emotional bindings with the reader and are a good central point for a story. They can either be protagonists, meaning everything revolves around them, antagonists, meaning they are between the protagonists and its goals, or extras, meaning they support the protagonist in some way. Conflict is what keeps the reader engaged and can also contain sudden surprises that radically shift the story. Finally, the conclusion is where the reader gets to discover if the protagonist was able to overcome the conflict or not.

Visual storytelling is an early-stage concept derived from the increase in interest in training artificial intelligence to interpret the semantics of visual scenes and describe them. J. Hu et al. [21] propose an innovative approach in this regard, obtaining stories that somewhat depict objects and context, based on a series of images. Although their experience gets impressive results and outperforms other previous attempts, it is still a work in progress. There also exist other artificial intelligence studies on story generation based on tables and text prompts.

Digital storytelling is the art of delivering stories using a variety of digital multimedia, such as text, photos, audio, and video. B. Robin [46] explains in more detail the characteristics of digital stories and what distinguishes them from ordinary multimedia content. He also presents the results of studies that implemented this technique in various contexts, from assorted levels of education to tourism and health science. Focusing on

¹⁴[National Storytelling Network website](#)

museums [59] and other non-profit community organizations, the integration of digital stories proved to be effective when engaging with the audience and capturing their attention.

Interactive storytelling, usually derived from its digital form, gives the reader the ability to intervene in the story and make meaningful decisions that impact its route and final outcome. Although the reader's decisions have great influence on the story, it will always be limited by the number of alternative paths it can take, due to the fact that they have to be usually written by an author [45]. The introduction of interactable characters to drive a story is one method of unrolling the narrative to the reader. Since they serve as the interface between the application and the reader, they often, besides providing story details, are a great way to give the reader the ability to interact with the story and make decisions on taking certain paths [30]. A. Katifori et al. [24] give an overview of the difficulties and challenges of applying interactive stories in a cultural heritage setting. Their main goal was to facilitate synergy between authors and mobile applications, which had the intent of implementing this technique. In doing so, they provide guidelines to overcome both design and authoring challenges.

Combining gamification with storytelling seems appropriate. As their integration in a application share the same objective, they will most likely end up augmenting one another. However, achieving this combination successfully isn't all that easy [18]. V. Cesário [9] provides research on guidelines for combining these techniques, having as case study museum trips experienced by teenagers.

C. Ibarra-Herrera et al. [22] give an example of how gamification and storytelling can be paired with educational intentions. They developed an application, to aid biology classes, that had several levels of challenges for students to complete, while applying storytelling when giving information about a subject. Following motivation tests, it was determined that students became more engaged in the topic, increasing their participation and overall learning. It is also mentioned that future improvements of the application would include AR elements, as they are very attractive and thus encourage students to interact more with the application. Other studies [38, 51] help support the successful results of using both these techniques together in an educational setting.

2.4.1 Challenge Design

As mentioned above, in gamification, point systems with leaderboards and awards like prizes, badges, and achievements are the most common game design elements that are incorporated into an application. However, in order to gain these points and awards, there first needs to exist some sort of challenges for the user to complete. The current platform allows for certified users to create narratives and itineraries for other users to follow, meaning that new locations and possible interactions are always emerging. This

gives rise to the necessity of implementing a challenge generation system, which is able to model itself around the narratives the user decides to follow and not being predetermined based on the existing ones.

J. Dormans [15] presents a study where generative graph grammars are used to create mission sequences. The simplest mission would be composed of a start, a group of challenges, and an end. A directed graph is a simple way to express this structure. Each node of the graph could potentially carry a different challenge. This could include enemies, treasures, tasks, or puzzles. Different missions, with parallel paths and rewards, can be generated by implementing various transformation rules within the grammar. This structure offers a highly controllable and expandable mission generation system, as the number of interactions is modified by the set of grammar rules and other graphs can be joined in a mission's non-terminal node in order to unravel it.

Lock and Key¹⁵ is a well known puzzle design pattern which can be easily incorporated in the generated directed mission graph. In its simplest structure, two nodes would be selected, one to carry a lock, such as in a chest or door, and the other would carry the respective key. The Lock node would provide some sort of barrier to progression in the mission graph, and the player, in order to breakthrough, would have to first find the Key node. However, in this case, it would be good game design to include a short, less rewarding path for players that would not want to interact with these types of challenges and would prefer to go straight to the end of the mission. As this type of puzzle can be efficiently introduced in about any space, it allows for a very flexible usage [2].

As there are players who would prefer to bypass some challenges, there are also players who would appreciate higher difficulty in order to stay engaged with the application. Applying the flow state ideology in this scenario, there should be balance between a player's skill and the mission's difficulty in order to increase player interest and retention.

J. Dormans and S. Bakkes [16] also explain how the same aforementioned generative grammars can be used to adapt a mission layout and scale its difficulty based on player models. As this mission graph structure is expandable, it is possible to grow parallel graphs, on non terminal nodes, based on a player achievements or performance in tasks. The difficulty scaling in each challenge can be achieved by dismantling the complete mission graph into several groups of smaller mission graphs. These can be reordered and combined or substituted by other smaller graphs depending on the desired difficulty and the used grammar adaptation rules. The choice of which adaptation rule to use for each player model is entirely up to the game designer, ensuring the structure remains controllable.

¹⁵[More about Lock and Key game design pattern](#)

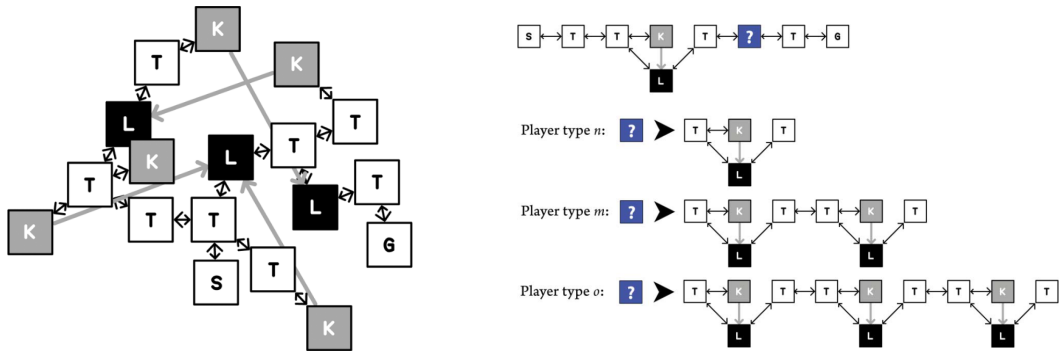


Figure 2.6: Organic mission graph layout on the left and Mission adaptation rules on the right - In [16]

SOLUTION DESIGN

This chapter begins by describing the case study, followed by the solution requirements, which clarifies the main features that should be included and some concerns to have when implementing them. Following that, a detailed breakdown of each feature's design is presented, along with some decisions taken in the process. The chapter ends with an overview of the technologies used and the system's architecture.

3.1 Case Study

This work is linked with an initiative led by T-Factor¹, a European project that aims to inspire innovative approaches to urban regeneration while encouraging co-creation and avoiding potential gentrification. One of their programs intends to grow the Trafaria neighborhood into a cultural hub. This includes the conversion of the Fort Nossa Senhora da Saúde da Trafaria into the IAT².

In collaboration with NOVA University³, as a way to progress towards improving Trafaria as a place, a augmented reality storytelling platform was created to help preserve Trafaria's cultural heritage. This platform ought to be improved and expanded during this work while also serving as its base. Likewise, all the contents included will take into consideration Trafaria's cultural heritage.

The base platform is divided into two components: web [41] and mobile [1]. Their relationship can be better visualized in (fig. 3.1).

The web component manages the creation of **point of interest (POI)**s, events, and itineraries while dealing with their respective upload to the web server. When creating a **POI** the user defines its location, general information, and the content that will be provided to enhance the experience in that specific point. This content can be either displayed when selecting the **POI** or only after arriving at its location. Itineraries can be created by grouping several **POIs** and defining the corresponding path between them. It is also possible to choose the files and amount of points awarded on completion.

¹T-Factor website

²More about the Institute of Art and Technology

³NOVA University website

The mobile component manages the following of the itineraries and the engagement with the provided content at each **POI** while dealing with their respective download from the web server.

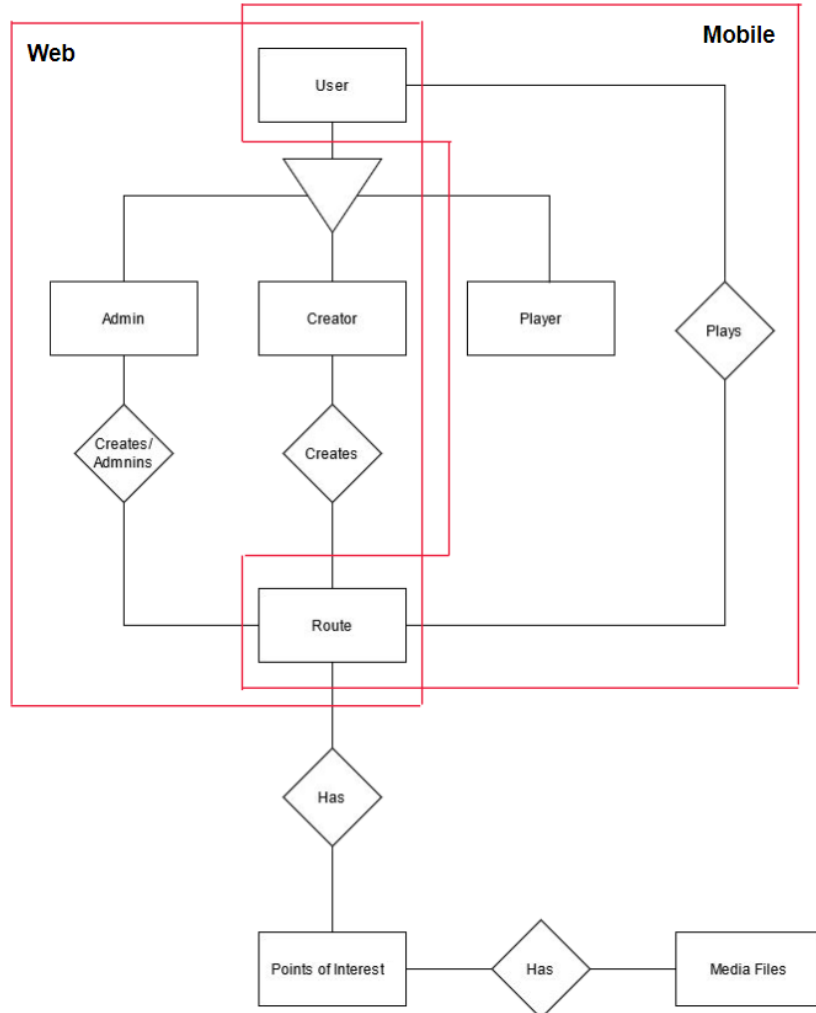


Figure 3.1: Base platform's Entity Relationship Diagram

3.2 Requirements

The main objective of this work is to ensure the preservation and exposure of communities' cultural heritage and stories, as well as to serve as an excellent educational resource by inspiring its users to seek out and discover more information, both inside and outside the platform, in order to gain knowledge about the cultures of other communities.

With this in mind, mechanisms to increase users' interest and the platform's compelling nature need to be implemented.

As studied in Section 2.4, gamification proved to be a very effective method when trying to increase and maintain users' engagement, motivation, and satisfaction. Taking this into

account, new gamified patterns of interaction must be introduced to the platform. These patterns should be customizable and relatively simple for users to incorporate into their narratives while still providing an interesting experience for the user interacting with them. It would also be beneficial if they had some sort of game-like feeling, resembling a fun task. Quizzes and puzzles are a good example of patterns that should be added. The quizzes can be created by only prompting text in the narrative creation interface, and then visualized using [AR](#). They also provide whoever interacts with them with relevant learning information. The puzzles can be created by only prompting images in the narrative creation interface, and their behavior fully controlled with [AR](#) capabilities to visualize them and move pieces around.

In order to introduce these new patterns, it should be taken into account not only their development but also their respective initialization through the platform's mobile app. Furthermore, despite not being the primary focus of this work, certain elements should also need to be included in the platform's web interface and server. These changes are necessary in order to support some of the additions.

Besides the interactions on user-created narratives, there should also be a global missions system with the intention of unifying interactions within all narratives as a cohesive storyline. Unlocking of content should be utilized as a technique to encourage users to go through the narratives. Lock and Key⁴ is a well known puzzle design pattern reviewed in Subsection 2.4.1, which should be incorporated in the generated mission system to achieve this.

Another desirable characteristic to have in consideration when developing both the new interactions and the mission system is that they should adapt to users' profiles, meaning there should be some sort of calculation of a user's difficulty level that affects these, scaling the difficulty to better suit all users, preferably making them harder for users of higher skill levels and easier for users with lower skill levels or that are not that interested in interacting with these often and do not want to be challenged in this app's context. Recapitulating, the main requirements of the solution are as follows:

1. Development of new game-like [AR](#) features;
2. Integration of the new [AR](#) features with the base platform's mobile and web components;
3. Development of a missions system capable of comprising the new [AR](#) features and utilizing a content unlocking pattern;
4. Incorporation of difficulty scaling into features that have the possibility of adapting to users' ability levels.

⁴[More about Lock and Key game design pattern](#)

3.3 Feature Design

After analyzing the solution requirements and also during the development, several ideas were considered and idealized throughout feature design phases. Each feature is briefly presented here, along with some explanation of the choices made before and while implementing them. Their implementation is later described in detail over the course of Chapter 4.

3.3.1 AR Interactions

For comprehension purposes, it is understood as an *interaction* all the distinct singular activities that can be launched in each POI during the following of an itinerary. In other words, the grouping of steps to be performed during each singular activity is defined in its entirety as an *interaction*.

As mentioned in the requirements (Section 3.2), puzzles and quizzes were the first basic and more obvious patterns of interaction that made sense to introduce to the platform, having its context in consideration. These are described as patterns because these should be customizable within the narrative creation interface, meaning when the users choose these as an interaction for some POI in their itinerary, they should be able to also prompt the necessary content as they like. For example, some desired image for the puzzles and a question and various answer options for the quizzes.

One small hurdle is that these interactions are, in essence, considered to be problem-solving exercises. So, in order to overcome this point at issue, they needed to be designed in a way that makes them resemble a fun task. This can be done by weaving in some game characteristics that could spark players' interest and maintain their motivation to use the platform.

3.3.1.1 Puzzles

Beyond the previously mentioned reasons, another justification for using puzzles as an interaction is that they do not require the player to become well-versed in a large number of complex rules in order to comprehend how to solve them. They are universally known, and therefore their rules are intuitively recognized by a vast majority of players.

Jigsaw Puzzle (fig. 3.2) - In order to create this interaction there should be a starting image, which would be the one prompted by a user in the narrative creation interface. This image should then be cropped in such a way that it creates the multiple pieces that make up the puzzle. The player should then have access to a list containing all the puzzle pieces and be able to select and place them somewhere in order to construct the puzzle, completing the interaction. With this in mind, there should also be a board, visualized using AR, where the player can place said pieces. This board could have a transparently dimmed preview of the original image and respective piece division as a means to help

the player discern the components of the puzzle and guide it towards the correct piece placement. After placing all the pieces correctly on the board and after a delay that allows the player to have time to admire the fully constructed puzzle image, there should be some sort of animation, for example, an explosion, that disperses all the pieces, revealing a final reward, also visualized using [AR](#), that can be clicked to be collected. This puzzle was initially designed to be fully interlocked and have blanks and interjambbs⁵. However, the implemented solution using Unity's shaders had some undesirable limitations and was not perfect, mainly due to the base platform using the Unity [Universal Render Pipeline \(URP\)](#). So this was decided against, and the puzzle pieces were instead made to be straight squares. This is not necessarily detrimental since it ultimately results in a cleaner puzzle that is more practical for players to understand and complete in an [AR](#) setting.

Sliding Puzzle (fig. 3.3) - Equally to the previous interaction, an initial image, chosen by a user in the narrative creation interface, is required to create the puzzle. The image should also be subsequently cropped such that it forms the various puzzle pieces. For this puzzle, instead of having a preview board and a list of pieces, the pieces should already be placed on the board which is also visualized using [AR](#). In order to compose the challenge, one of the pieces should be removed or faded out to create an empty space, and the rest of the pieces should be scrambled out of their correct position. The scrambling implementation must be mindful of the fact that not every assortment of pieces results in a solvable puzzle. There are various ways to confront this problem, such as having a group of predetermined assortments, or randomly assigning the pieces positions and then calculating if the puzzle is solvable or not, or the solution that was ultimately used, to work backwards from a solved puzzle, mixing it up. To solve the puzzle, the player should be able to move pieces adjacent to the empty space into its position by clicking on them until all the pieces are in their correct positions. After solving the puzzle, by moving all the pieces to their correct position, the missing piece that was initially removed or faded out should fully reappear, showing the player the entirety of the puzzle image. After a delay that allows the player to have time to admire the fully constructed puzzle image, there should be some sort of animation, for example, an explosion, that disperses all the pieces, revealing a final reward, also visualized using [AR](#), that can be clicked to be collected.

Difficulty Scalling - As a means to add a certain kind of game challenge characteristic, the puzzles should be timed, and the timer should be displayed to the player. The players' ability levels could be calculated using the times they took to finish each puzzle. From here, there are various ways to incorporate difficulty scalling into these interactions: either the player could be given a target time to beat, and this time would be adapted to the player's skill level, or the puzzle itself could be divided into a different number of pieces also based on the player's skill level. Ultimately, it was decided that the average of

⁵Widely recognized terminology for a jigsaw puzzle piece's indentation and corresponding rounded tab, respectively

times from the last ten puzzles of the same type would be used to determine each player's skill level, and the puzzle dimensions would be altered to suit it. With this in mind, a puzzle image should be cropped into more pieces if the player was, on average, very fast to complete the previous puzzles and cropped into fewer pieces if the player was, on average, very slow to complete the previous puzzles. Additionally, in the jigsaw puzzle, unlike the sliding puzzle, there is the possibility of a player making a conspicuously improper move by trying to place a piece incorrectly on the board, not in its correct position. This move could be penalized, further adding an additional game characteristic. It was determined that, in this instance, the timer would have to be incremented by ten seconds and a notorious animation should be played to inform the player of its mistake.

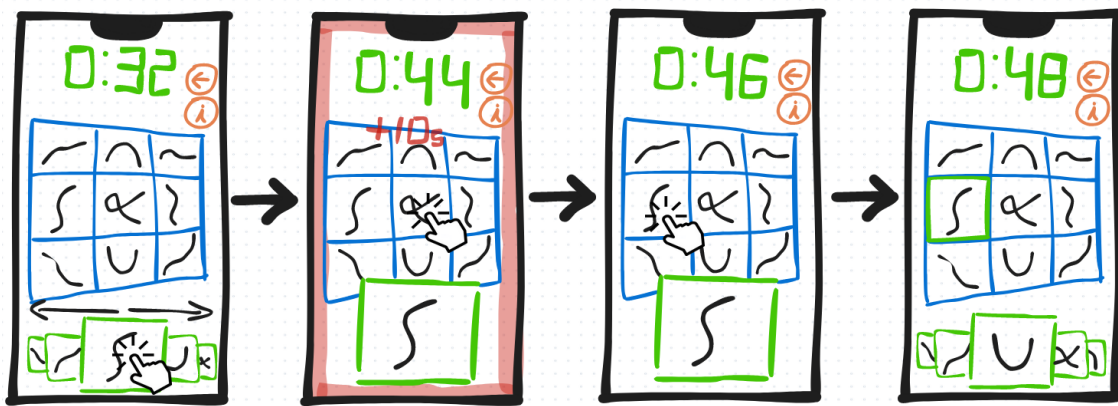


Figure 3.2: AR Jigsaw Puzzle Design Drawing

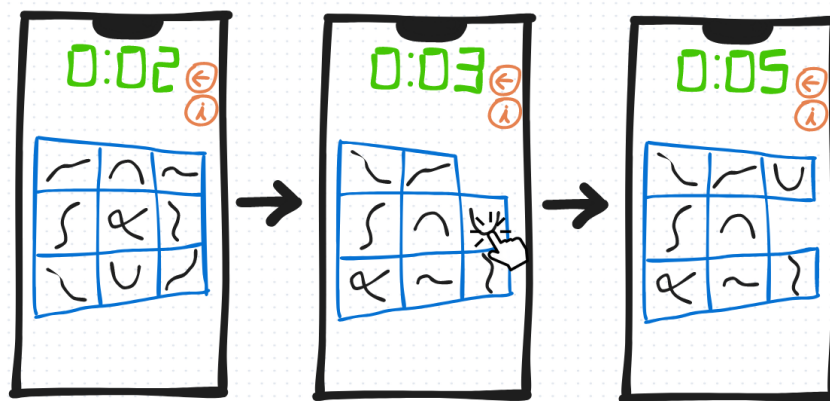


Figure 3.3: AR Sliding Puzzle Design Drawing

3.3.1.2 Quiz

As previously mentioned, a justification for using quizzes as an interaction, is that they can serve as an effective learning experience by engaging players in active recall and reinforcing interesting information. Quizzes can also provide immediate feedback on selected answers, allowing players to assess their knowledge and identify areas that can be further

studied. Overall, integrating quizzes as a learning interaction promotes active participation, enhances retention, and supports the continuous improvement of players' knowledge.

In order to create this interaction, there should be a starting question and various answer options, including the correct one. These would be prompted by a user in the narrative creation interface. The player should be able to visualize both the question and answer options using AR. This should be achieved by transforming each character into a three-dimensional object of itself and correctly placing it in the scene to form the correspondent words and phrases that make up the prompted text. There should also be a set limit for the number of characters in a single line, making the three-dimensional text continue to be written on the next line below. In order to engage with the quiz, the player should be able to select the various answer options by clicking on them. An animation should then be played to inform the player whether the chosen answer option was the correct one or not, enforcing the aforementioned immediate feedback on selected answers. In this quiz, the player would have the chance to choose two answer options before failing the quiz, meaning he could choose an incorrect answer option once and then be given one more chance to choose another answer option. In case the player fails the quiz by choosing two incorrect answer options, another animation should be played as a means to inform the player which of the answer options was the correct one, maintaining the relevancy of the quiz as an effective learning interaction. In case the player succeeds in choosing the correct answer option, there should be some sort of animation, for example, an explosion, that disperses all the three-dimensional text characters, revealing a final reward, also visualized using AR, that can be clicked to be collected. In the hope of increasing the quiz's appeal to the player, it was decided that the three-dimensional text that composes the question would be displayed right in front of the player, and the three-dimensional text that composes the various answer options would be orbiting around the player, each with a different designated height and with randomly assigned speeds and starting positions on their orbit circumference. This can be visualized in (fig. 3.4).

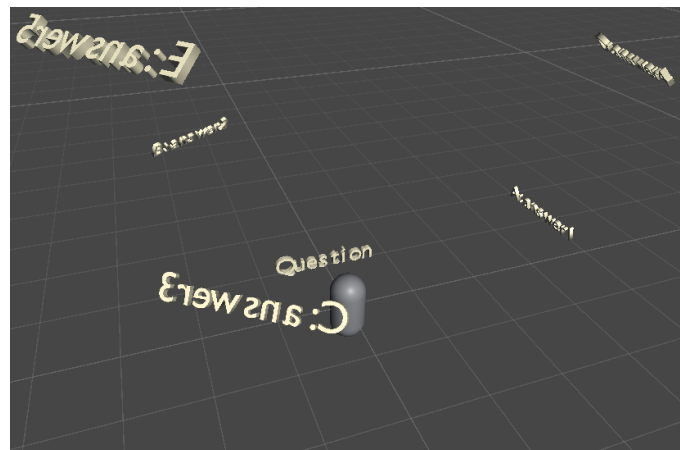


Figure 3.4: AR Quiz Design Preview

3.3.1.3 Balloon Pop

This feature was designed with the intention of having a straightforward mini-game as an interaction option for users to choose and add to a [POI](#) while creating their itineraries. Such a mini-game would intentionally stray from the context of the platform, meaning that, in its base form, it would not provide the player with any beneficial elements other than fun and the respective rewards. However, it would be easily built upon to incorporate these missing elements if needed, such as relevant and customizable information, similarly to the previously described patterns of interaction.

After some consideration of various mini-game types, it was determined that the balloon-popping concept was of appropriate suitability to the aforementioned underlying requirements. It can be played in an [AR](#) setting. It does not require the player to understand a great deal of intricate rules in order to play, much like the puzzles. And lastly, it provides a solid foundation of a fun mini-game that, if necessary, offers the possibility to develop and incorporate new characteristics that would enable it to deliver players a learning experience, similarly to the quizzes. Its sketches can be seen in (fig. 3.5).

To put this balloon-popping concept into practice, there should be a way for balloons to appear somewhere in the player's surroundings, visualized using [AR](#), and a way for the player to throw something at them in order to pop them. There are several approaches to constructing this mini-game. It was ultimately decided that the balloons would be constantly spawning and ascending from a stationary portal, and the player would be able to tap the screen in order to throw a dart and try to pop the balloons. As a means to help the player identify the defined area where the balloons are able to spawn, the portal should have some sort of notorious graphic or effect applied to it. Since the dart will be thrown with a tap of the screen instead of a force-defining and direction-defining swipe, it should be given additional attention to its configuration so as to feel satisfying from any distance, and a crosshair should also be included so the player knows in which direction it is throwing. This should be done in order for the mini-game to be easier and maintain its simplicity relative to how it is interacted with and played. To avoid spamming dart throws and to better control the later described difficulty scaling of the mini-game, there should be a limit to the amount of darts a player can throw per second, which could be described as their reload time. There is the possibility of having a pool of different balloons to spawn from, each with its own shape and properties, like ascending velocity and size. This would add more dimension and depth to the mini-game without introducing any difficult rule complexity. When the player finishes the mini-game, there should be some sort of animation, for example, all balloons currently unpoped in the player's surroundings would sequentially pop one after the other, revealing a final reward, also visualized using [AR](#), that can be clicked to be collected.

Difficulty Scalling - Taking inspiration from the previously designed puzzle interactions' difficulty scalling, it became apparent that this mini-game should also be timed, and the timer should be equally displayed to the player. With this in mind, the players' ability levels would be calculated analogously, using the average of the times from the last ten balloon-popping mini-games performed. There should now be a particular final goal for the player to pursue, which enables it to finish the timed interaction. This goal should be adaptable relative to each player's skill level. Two ideas were designed: there could either be a point system with a desired final point quantity and each distinct balloon would contribute different amounts to that goal when popped, or there could just be a counter for a target number of balloons to pop. It was also thought that certain balloons could offer perks or trigger abilities when popped, for example, reducing the current timer, slowing balloons' ascent, having explosions, firering darts faster, and the list goes on. Following a final assessment, the second alternative was selected for implementation in order to keep anything that would complicate understanding the mini-game rules as straightforward as possible, which was, as aforementioned, the main intention when designing it in the first place. Given that chosen option, the player should, as expected, be able to visualize both the target number of balloons to pop and the current amount of balloons popped during the mini-game. Considering this, the target number of balloons to pop should be higher if the player was, on average, very fast to beat the previous mini-games and vice versa.

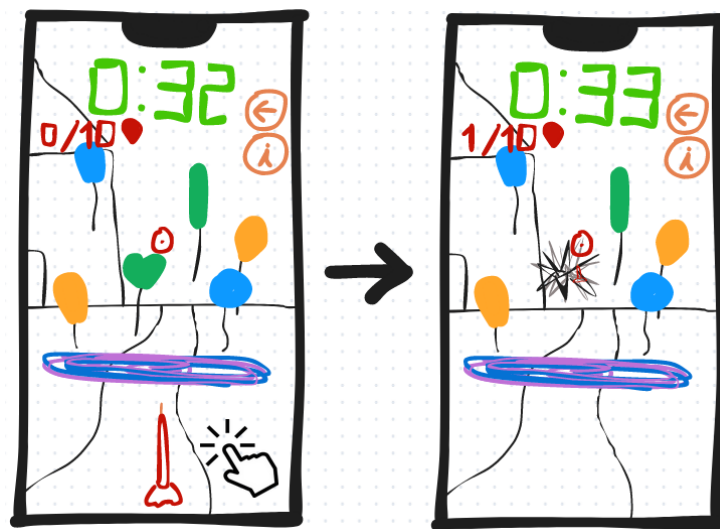


Figure 3.5: AR Balloon Pop Design Drawing

3.3.1.4 Geospatial Flyers

The idea for this feature emerged from the need for an interaction more tightly suited to the context of the application. The complete opposite of the previous interaction. Although puzzles and quizzes are assumed to be fantastic options, there needed to be something that presented a [POI](#) with even more pertinent information in an engaging manner. In

light of this, the thought of combining the interesting images of puzzles with the learning potential of quizzes arose. So it was put together the concept of having floating flyers that would hover above some POIs. These flyers would be visualized using AR and would contain relevant images of their corresponding POI along with pertinent information about it and an associated quiz regarding it as well. Furthermore, it was decided that this interaction would not be an option for users to choose and add to a POI while creating their itineraries and would instead be an entire independent interaction in and of itself. With this in mind, the process of creating a new flyer within Unity should be made to be as simple of a task as possible. This could be achieved by having a default blank flyer that can be minimally changed, preferably in the inspector, to create a new one for a specific POI. The first iteration of the flyers' cover and interior design can be seen in (fig. 3.6).

Each flyer should start in a closed or folded position, only displaying the name of its assigned POI and the name of its location. The player should then be able to click on it in order to open or unfold it and access its interior, where the respective additional information and quiz are located. The player should then be able to click on the quiz's answer options in order to choose them. Equally to the quiz AR interaction previously described, the player should have immediate feedback on the selected answer option, displaying whether it was correct or not, and in case it was not, the correct one should be highlighted, maintaining the relevancy of the quiz inside a flyer as an effective learning experience. To help the player know which way to walk in order to encounter each flyer, a compass should be added.



Figure 3.6: First Iteration of Flyers' Cover and Interior Design

3.3.1.5 Miscellaneous

Some other minor features, which are common to more than one interaction, were also designed, and they are described below.

Interaction Positioning - It was decided that there should be two main ways to position an interaction within the players' surroundings: either the interaction is previously

positioned in the scene relative to the starting position of the player, or the player is given the responsibility of positioning it itself within the start of the interaction.

Information and Instructions Foldout - This feature was designed as a means to help players who could ever feel lost at any point during an interaction. To accomplish this objective, there should be a button in the interface with a standard and therefore intuitive symbol for this purpose. After clicking on it for the first time, the player should be presented with a foldout with short, simple, and direct instructions on how to proceed during the interaction. This foldout could also be taken advantage of to show additional interesting information. For example, the puzzles could have a preview image displayed, the quiz could have a preview of the interaction itself, so the player understands it better, and the balloon pop could have a description of each balloon's properties. The player should be able to close the foldout by either clicking on its button again or anywhere outside of its perimeter. Certain interactions might need to take into account whether the foldout is open or not since they might be waiting for any type of player touch with the device's screen. In other words, if an interaction is in a state that is waiting for any type of player touch on the device's screen, then a touch meant to either open or close the foldout should not be considered.

Task Completion Popup - During an interaction, the mission system, which is later described in Subsection 3.3.2, is responsible for receiving updates from the interaction itself and managing the progress on all active tasks. Every time a task is completed during an interaction, the player should be notified of its completion. It was considered that an animated interface element that would slide in and out of the screen would suffice.

3.3.2 Mission System

As mentioned in the requirements (Section 3.2), there needed to be something that was able to unify all the interactions within the narratives to make them resemble a cohesive storyline. This could be achieved through a mission system, where there would be multiple questlines, possibly with stories behind them, that would each be composed of several missions, and each mission would subsequently be composed of several tasks. These tasks would be directly associated with the [AR](#) interactions mentioned in Subsection 3.3.1. The player would progress in the storyline by engaging with the interactions and completing their correspondent tasks, completing the loop, and thus ensuring that the intended cohesive storyline effect is accomplished.

After searching for already developed conventional Unity mission systems to integrate into this work, it was observed that all the discovered ones had at least one of three undesired negative attributes. The systems found would either be unnecessarily complex, not free, and sometimes even expensive, or would follow the same laborious design where the user would have to write all the logic for each independent quest.

Consequently, a new mission system had to be designed. The highly popular location-based mixed reality mobile game *Pokemon GO*⁶ served as the main inspiration and model for this mission system's external design, and the study of using graph structures to generate adaptable questlines from Subection 2.4.1 was taken into consideration for the internal design, as was the requirement of the Lock and Key⁴ pattern integration.

With this in mind, it was decided that this mission system would come about by first developing a built-in tool, using Unity's available [Graphical User Interface \(GUI\)](#) customization, for the purpose of having a general mission graph creation tool that can be used by any other Unity game and that at the same time obliged to the mentioned requirements. Secondly, the created mission graphs would need to have an interpreter that would be able to manage them by constructing the actual missions and dealing with their progress and data persistence. Thirdly, there would also be an interface capable of displaying some desired information and allowing players to interact with it.

3.3.2.1 Mission Graph Creation Tool

This tool was designed in such a way that it allows the user to not write any code in order to create a mission graph. It revolves around the idea that a graph is, naturally, composed of a set of connected missions, with each mission composed of a set of its respective tasks to complete and rewards awarded for completing all its tasks. From here, three main component types can be highlighted: the mission graph itself, with each node representing one mission; the rewards of each mission; and the tasks of each mission. All three of these component types should have their own independent custom creation and management window interfaces. The second mentioned custom [GUI](#), for management, would be very similar in all three component types. These would all have a list of every already created objects of their respective component type, along with a button to select each one and open a correspondent editable preview which would also have a delete option. There should also be a button to open the custom creation [GUI](#) of the respective component type.

Tasks - Taking into consideration the aforementioned inspiration for the styles of tasks that would be created, it can be determined that each task should have a display text and a list of variables, and that these variables would be of two types: accumulative and requisite. Accumulative variables would serve as the target goal that is incremented when progressing in a task and are also responsible for the displayed completion percentage calculations in the interface. Requisite variables would serve as mandatory values that should be met before incrementing accumulative variables, these could be of maximum, minimum, or equal types. The user should also be able to prompt something within the display text that would reference task variables, so these could be displayed as well.

⁶[Pokemon GO website](#)

For example, in *"Do action {varName} times"*, {varName} would be substituted in the interface for whatever task variable's value it refers to, in this case an accumulative variable named *varName*. A quality of life button should be included to automatically add task variables present in the display text to the variable list. Furthermore, the implementation of task variable types should consider the possibility of later adding new types of variables.

Rewards - Also taking into consideration the aforementioned inspiration, the rewards should be displayed in the interface with a representative icon. The rewards should be able to be collected from the interface after a player completes all tasks in a mission. The implementation of their collection method should make their creation as effortless as possible, again making sure the user does not have to write any additional code or logic.

Mission Graph - The mission graph creation window should offer the user the possibility of adding and removing nodes representing each mission and also of drawing edges between the nodes. Each mission node should also allow the user to add and remove its respective previously created tasks and rewards. In order to directly support the Lock and Key⁴ pattern, each mission should have a way to let the user determine if they are normal, a lock, and/or a key. If there are missions with lock and/or key properties, then the user should be prompted to draw additional edges between all the lock nodes and their respective key nodes.

3.3.2.2 Mission Graph Interpreter

To have the ability to process the created mission graphs, an interpreter needs to be developed. The interpreter is responsible for handling how the mission graphs are used to create the actual missions in the game, managing their progress, and also providing the interface with the required information. Different interpreters can be designed to achieve different final results. This interpreter should be able to let the user choose which mission graphs it wants to be processed and available in the game. Furthermore, it was decided to have, in addition to the **questlines**, a **daily quest** that would reset every day at the same hour, and special **achievements** that would only use tasks themselves. Relatively to the processing of the mission graphs, it was determined that in this case, progress would be tracked in a mission that has a path to the starting node with no unfinished missions in between. It was also determined that progress would be tracked even if the node was a lock, but it could only really be finished when unlocked by finishing all its respective key missions. Progress in the mission system, which is tracked by the interpreter, should be implemented in such a way that the user only has to call a singular function in its game code that sends the interpreter the information needed.

Difficulty Scaling - In this system, there are several approaches to scaling the difficulty of missions. It was ultimately decided that the accumulative task variables would

have minimum and maximum values, which would be prompted by the user when creating them. Therefore, when a mission is activated, each of its accumulative task variables will have its value calculated between the respective minimum and maximum, taking the player's ability level into account.

3.3.2.3 Interface

As mentioned before, the interface was designed having Pokemon GO⁶ as inspiration. Their missions interface was studied and divided into relevant components, and it was decided that all those components should be included in the developed interface as well. The division into detected relevant components, that should be included, can be better visualized in (fig. 3.7) and each component design is also briefly described bellow:

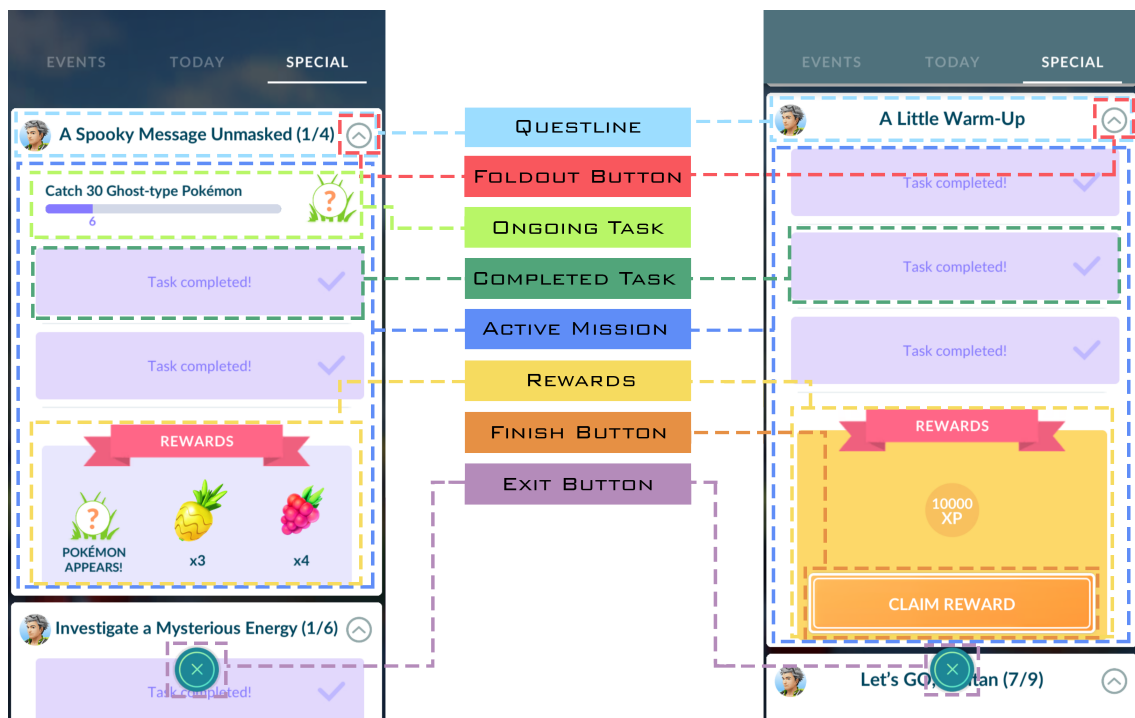


Figure 3.7: Pokemon GO Missions Interface Components

- **Questline** - Each questline, which is composed of all missions in a mission graph, should have an entry in the interface with a header containing its name and a content space to display active missions.
- **Foldout Button** - The player should have the possibility of opening and closing questlines' content as a foldout. To accomplish this, a button should be included in every questline's headers.
- **Ongoing Task** - Tasks that haven't been completed should have a text displaying the objective of the task itself and also some sort of indicator of its progress.

- **Completed Task** - Tasks that have been completed should be easily distinguishable from other tasks. For example, having a different background, a checkmark, and no longer displaying their objective text and progress.
- **Active Mission** - All active missions within a questline should be displayed in their respective questline's content space. In this work, there is the possibility of a mission being a lock and/or a key. It was decided that key missions would be displayed like normal missions. Meanwhile, as it was previously determined in Subsection 3.3.2.2 that a lock mission would have its progress tracked when active, it should also be displayed, but with some additional indication of its locked state.
- **Rewards** - The icons for each mission's rewards should be displayed in a designated space below their respective mission's tasks.
- **Finish Button** - When all tasks of a determined mission have been completed, its representation should be easily distinguishable from an unfinished mission. There should also be a button that allows the player to collect said mission's rewards and progress in its respective questline. In the case of a lock mission, it was decided that it would suffer the same changes in appearance while still maintaining the additional indication of its locked state. A lock mission's finish button, if not yet unlocked, should not be clickable.
- **Exit Button** - There should be a button that allows the player to exit the interface at any moment, so it feels in control.

As it can also be seen in (fig. 3.7), the interface includes three tabs at the top that allow users to scroll through different pages. This should also be implemented in the developed interface in order to scroll between the daily quest page, normal quests page, and achievements page. The player should be able to switch between pages by clicking on each tab's name or by swiping left and right on the screen.

3.4 Technology and Architecture

The development of the additional functionalities was mostly done around the same technologies as the base platform's mobile component. There were also some other supplementary technologies used to implement certain features or to help create some desired content. All these technologies are described below:

- **Unity Game Engine**⁷ is a leading cross-platform game engine renowned for being easy to use and versatile. It provides developers with a multitude of tools for mainly creating games, interactive experiences, and simulations, across a range of platforms.

⁷Unity website - last accessed: 03/2024

Its user-friendly interface and strong community support make it a popular choice for game development.

Since it was also used to implement some of the base platform's mobile features, it continued to be the main development tool used for this work.

- **ARFoundation**⁸ is a framework that allows developers to create [AR](#) experiences within Unity. It abstracts the complexities of working with different [AR](#) platforms, such as ARKit for iOS and ARCore for Android, providing a unified [application programming interface \(API\)](#) for common [AR](#) tasks, making it easier for developers to create immersive [AR](#) experiences with Unity.
- **ARCore Extensions**⁹ is an external add-on for the ARCore platform, which extends its capabilities within Unity. It allows developers to integrate more advanced [AR](#) features, such as cloud anchors, into their Unity projects. It also simplifies the development process by providing a user-friendly [API](#) to implement said features on Android devices.
- **Geospatial Creator**^{10,11} is one of the ARCore Extensions features that helps developers build and launch robust world-anchored experiences within Unity. Furthermore, the uncomplicated [API](#) and Unity component interfaces allow developers to anchor objects in real-world locations while visualizing a three-dimensional world preview. This technology was mainly used for the geographic coordinate-based positioning of the Geospatial Flyers, previously described in Subection [3.3.1.4](#).
- **Cesium for Unity**¹² is a platform for creating virtual globes, allowing developers to integrate accurate real-world geographic information, such as satellite imagery, terrain data, and three-dimensional buildings, into Unity projects. It is able to receive information from Geospatial Creator and optimize it as three-dimensional tiles for visualization and interaction within the Unity ecosystem.
- **Unity UI Toolkit**¹³ is a framework for creating custom user interfaces within Unity projects. It provides developers with a flexible and efficient way to design and implement interfaces using built-in elements such as buttons, sliders, text fields, and more, while also allowing for dynamic layout and styling using style sheets.

This framework was used to implement the custom [GUI](#) that allows to manage and create the Tasks, Rewards, and Mission Graphs described in Subsection [3.3.2.1](#). The *Graph View* UI toolkit feature, which is still in its experimental stages, was also used and tested for the Mission Graph creation and editing window interface.

⁸ [ARFoundation website](#), [ARFoundation manual](#) - last accessed: 03/2024

⁹ [ARCore Extensions Documentation](#) - last accessed: 03/2024

¹⁰ [ARCore Geospatial Documentation](#) - last accessed: 03/2024

¹¹ Geospatial Creator was launched by Google on May 2023, after this thesis had already started.

¹² [Cesium for Unity website](#) - last accessed: 03/2024

¹³ [Unity UI Toolkit Documentation](#) - last accessed: 03/2024

- **Blender**¹⁴ is an open-source software that offers a wide range of features, including three-dimensional modeling tools, sculpting, rigging, texturing, animation, simulation, and more.

This software was used to create some of the three-dimensional models present in this work, such as the star reward that appears after completing some interactions and the default Geospatial Flyer model. The latter also had its respective opening animation and uv coordinate texturing done using Blender as well.

- **Adobe Photoshop**¹⁵ is an image editing software mainly focused on raster graphics that is widely used by professionals and hobbyists alike to create and manipulate digital images. This software was heavily used throughout this thesis.

In (fig. 3.8), it can be seen how the aforementioned Cesium for Unity technology is able to provide a three-dimensional real-world view inside of Unity's editor.

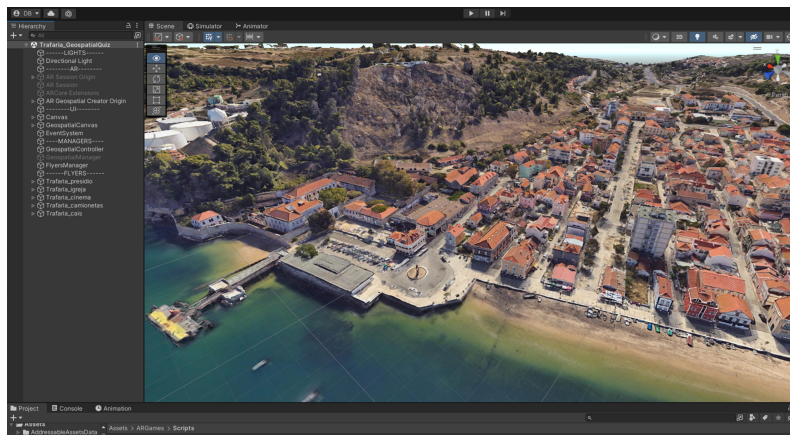


Figure 3.8: 3D View of Trafaria inside Unity's Editor

A high-level system architecture is presented in the diagram of (fig.3.9), and it shows the main components of the system along with their connections. The player engages with the interactions by using a mobile device with [AR](#) capabilities and a minimum Android [API](#) level of 24. Geospatial Creator sets this higher [API](#) requirement¹⁰. The device runs the Unity Application, which contains a different manager for each interaction that is responsible for dealing with the player's inputs and controlling the interactions' flows. All the interactions are connected to the Mission System since it contains tasks corresponding to all of them, whose progresses need to be tracked. Data persistence is achieved with the Data Controller, which saves and loads files at a standard location in the device's file system. This is used to keep some necessary data, like game results for difficulty scaling, flyer completion, and the Mission System state and progress. Some interactions require content that is provided by a user in the narrative creation interface. This is simulated

¹⁴[Blender website](#) - last accessed: 03/2024

¹⁵[Adobe Photoshop website](#) - last accessed: 03/2024

by performing a *UnityWebRequest*, which uses the HTTP protocol in order to download this content from external files. Finally, ARCore Extensions, which is responsible for incorporating Geospatial Creator, accesses a project on Google Cloud through an [API](#) key. This project enables the use of geographic anchor positioning, world visualization in Unity, and the visual positioning system.

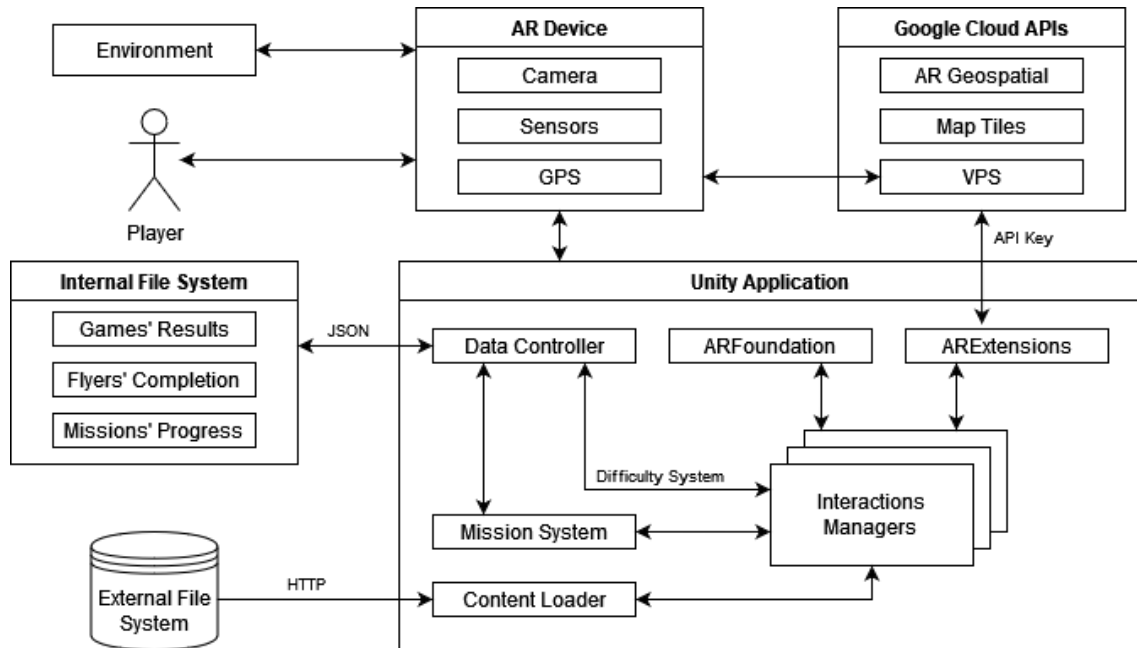


Figure 3.9: High-level Representation of the System's Architecture

IMPLEMENTATION DETAILS

This chapter provides a comprehensive explanation of the implementation process for each designed feature, considering the previously outlined requirements and decisions discussed throughout Chapter 3.

4.1 AR Interactions

For the development of each AR interaction, there was a need to setup a different scene that included all the necessary components responsible for its behavior, AR integration, and communication with some other features and outside components.

The process of integrating AR capability and functionalities in a determined scene within Unity is made simple with the use of ARFoundation, having to instantiate two mandatory *GameObjects* identified as *AR Session* and *AR Session Origin*. The first represents the core of the experience and is responsible for managing and tracking the device's position and orientation while also handling the processing and integration of the real-world data that is obtained through various of the device's sensors, serving as the connection between the surrounding environment and the virtual elements. The latter represents the starting position of the device and acts as the primary reference point in the scene to which all other elements correlate, making it responsible for the manipulation of the virtual content placement and scaling. Since the *AR Session Origin* scale was set to the default value of one, all the hereinafter dimensional values mentioned have a 1:1 scale ratio between the real-world and the virtual environment.

4.1.1 Puzzles

Both the designed puzzles are composed of a set of pieces that, when positioned correctly, construct a full image. While the two puzzles present different challenge designs in relation to the way the pieces are initially laid out and then moved around, they use the same approach when loading their respective image and creating the pieces themselves.

As the images used in these puzzles would be prompted by a user in the narrative creation interface of the base platform and subsequently stored in the database, it basically

means that they need to be downloaded from an external file system. With this in mind, a *UnityWebRequestTexture* is performed to obtain said images from a URL¹ already in a usable texture format within Unity. In the event that something goes wrong and the image does not get correctly loaded, for example, if the request returns a *ConnectionError* or *ProtocolError* response, then a UI element appears, notifying the player that the asset failed to be loaded and giving it the option to exit the interaction. On the contrary, if everything goes right and the image is correctly loaded as a texture, then it is applied to two separate materials, one dedicated to the three-dimensional objects of the pieces (materialA) and another dedicated to the UI previews of the image and, in the case of the jigsaw puzzle, also the pieces present in the list (materialB).

In the normal course of these interactions, after the image is successfully loaded, the player then has to choose a position for the puzzle board within any desired detected space, this is later detailed in Subsection 4.1.5. After that, the main behavior controller for these interactions starts generating the puzzle pieces and positioning them accordingly. For clarification, the placed board is actually just an *empty GameObject*, which is only used to position the pieces in relation to it.

Generating the Puzzles

Each puzzle piece is essentially a *Quad*, which is a very simple flat geometry that only contains one face made up of two triangles. The pieces that are positioned on the puzzle board additionally have a flattened cube in order to create some dimension and depth while also making the puzzle visible from the back, as the *Quad* by itself does not have a back face. A single puzzle piece *Prefab* was made for both cases in order for them to be instantiated at runtime through the behavior controller. To be able to generate the puzzle, some additional parameters are defined, such as the width of the puzzle, the number of pieces it is divided into, the gap thickness between each piece, and, in the case of the jigsaw puzzle, also the transparency alpha of the board and the time increment in seconds.

The puzzles were designed to be square, having the same width and height. The width value of the puzzles can be changed without affecting their correct generation since all the necessary calculations take the puzzle dimensions into consideration. It was ultimately decided to make the puzzles two meters long, as during experiments, this seemed like a suitable proportion. To achieve this, because of the aforementioned *AR Session Origin* scale, the *puzzleWidth* was set to be two Unity units long.

The number of pieces a puzzle is divided into is the designed dynamic factor that enables the puzzle to be **difficulty scaled**. Therefore, it is calculated based on the player's ability level by using the average of the last ten times that the player took to complete the puzzles of each type. These values are loaded from an internal file, along with the level of the previous puzzle performed. These last ten values of time are managed using a circular array. The level of a puzzle corresponds to the number of pieces along one side and is

¹Uniform Resource Locator

identified as its *size*. So if a puzzle has nine pieces, its level and *size* have a value of three. It was decided that the starting level for both puzzles is three, and whenever a player has an average of times superior to 90 seconds, then the level of the puzzle decreases by one, and on the contrary, if a player has an average of times inferior to 30 seconds, then the level increases by one. The minimum level is two and there is not a maximum level.

Following that, the aforementioned single piece *Prefab* is instantiated once for every puzzle piece as a child of the board. To simplify certain aspects, it was decided that the center of the puzzle board would be the same as the combination of all the pieces, so each piece gets scaled and is positioned going from -1 to +1 via the subsequent calculations based on their row and column, and also the *gapThickness* defined:

$$width = \frac{puzzleWidth}{size} \quad localScale_{(x,y,z)} = (width - gapThickness) * (1, 1, 1)$$

$$LocalPosition_x = (width * col) - \frac{puzzleWidth + width}{2}$$

$$LocalPosition_y = \frac{puzzleWidth - width}{2} - (width * row)$$

Each piece is also given a name matching its position on the board ($row * size + col$) and gets assigned the aforementioned materialA. The previously loaded texture that was applied to this material corresponds to the entirety of the image, so in order for each piece to only display its respective section of the image, its *Mesh* retrieved from the *MeshFilter* component needs to have the UV coordinates² altered. Because of how the image is loaded, the pieces are instantiated, and the *size* of the puzzle can vary, these coordinates have to be calculated dynamically at runtime. The position of each piece's vertices within the texture, depending on the *size* and *gapThickness* of the puzzle can be seen in Table 4.1. This coordinate system is normalized from 0 to 1 and the origin exists in the bottom left corner. The order in which they are set is top-left, top-right, bottom-left, bottom-right.

Table 4.1: Puzzle pieces' UV coordinates positioning

UV	Horizontal Position	Vertical Position
(0, 1)	$(nWidth * col) + gap$	$1 - (nWidth * (row + 1)) + gap$
(1, 1)	$(nWidth * (col + 1)) - gap$	$1 - (nWidth * (row + 1)) + gap$
(0, 0)	$(nWidth * col) + gap$	$1 - (nWidth * row) - gap$
(1, 0)	$(nWidth * (col + 1)) - gap$	$1 - (nWidth * row) - gap$

$$normalizedWidth = nWidth = \frac{1}{size} \quad gap = \frac{gapthickness}{2}$$

After the puzzle is successfully generated, a timer appears at the top of the screen and starts counting forward from zero. This feature is detailed in Subsection 4.1.5.

²UVs are two-dimensional texture coordinates that are responsible for how an image texture gets applied onto a geometry.

Jigsaw Puzzle

Specifically in the jigsaw puzzle, when each board piece *Prefab* is instantiated, it has its color's alpha value set in order for the board to be transparent, forming the desired preview effect. Additionally, while instantiating each piece, a copy that uses the same UV coordinates is also instantiated as a child of a *ScrollView's container*³, this time using the aforementioned materialB which has the same texture but is better suited for UI elements. All the pieces in the list are then shuffled by assigning each of them a random sibling index between zero and the number of pieces.

This *ScrollView*, containing all the pieces, has them organized horizontally, equidistant from each other, and is positioned at the bottom of the screen, establishing the designed list of pieces. Since by default it only allows for unrestrained horizontal scrolling by swiping on the screen, as a means to make the scrolling experience more satisfying, a script meant to control its behavior is applied to it. This script uses both the number of pieces present in the container and the current scroll position value in order to calculate which piece is closest to the middle. When a user stops swiping, the scroll value is then changed in order to snap this piece into the middle while also moving the other pieces in the list accordingly. The piece's snapping is done smoothly by using linear interpolation between its current position and the scroll's middle position, and it takes into account a determined speed, which can be changed, as well as the time between rendered frames, so as to make the consequent animation frame rate independent.

The scale of the pieces is also changed during and after the swiping action, making the middle piece increase in size and all the others decrease in size. A smooth transition between the different scales is also achieved using linear interpolation while taking into consideration the same aforementioned characteristics. Furthermore, it was made possible to scroll directly to a certain piece by clicking on top of it when it is not in the middle. Conversely, clicking in the middle piece selects it, making it increase even more in size for better visualization and vanishing the other pieces. From here, the player is able to either click on it again to deselect it and cause the list to go back to normal or to click on a piece present on the board to try and place it there. The act of clicking the pieces is accomplished by performing a *Raycast*⁴ from the touched screen position straight to the far clipping plane, in order to detect if a piece was clicked on and, if so, which piece it was. Two different types of *Raycast* are made, one specifically for UI elements and another for the objects. In order to distinguish the start of a swipe touch from a click touch, some calculations are performed based on the time the user spent touching the screen as well as the distance covered by that touch during that time.

When trying to place a piece on the board, there are two possible outcomes. If the selected list piece's name matches the one clicked on the board, then it is removed from the

³ScrollView is a Unity UI element composed of a ScrollRect, a container, and a scroll bar. ScrollRect allows the designation of an area and the horizontal and vertical scrolling of elements present in the container within that area.

⁴Raycast is essentially a ray that is fired from a point in space along a particular direction, returning references to objects hit.

list, and the board piece gets its color's alpha value set to normal. On the other hand, if the selected list piece's name does not match the one clicked on the board, then three distinct but synched animations are played to inform the player of its mistake. The first blinks a red border around the screen by changing an image color's alpha value to one and back to zero. The second swivels the selected list piece to the right, then to the left, and then back to its initial rotation. The third makes a text, which displays the time increment, appear in the middle of the screen, that then slides upwards towards the timer and disappears.

Finally, the puzzle is completed when $size^2$ pieces are placed on the board.

A unique sound effect is played, using the AudioManager detailed in Subsection 4.1.5, whenever the following events happen: The list's middle piece changes; The player selects a piece; The player deselects a piece; The player places a piece on the board; The player fails to place a piece on the board; The player completes the puzzle.

This interaction can be visualized in (fig. 4.1), where on the left and middle there can be seen level three and four jigsaw puzzle, and on the right there can be seen the animation played when trying to incorrectly position a piece on the board.



Figure 4.1: AR Jigsaw Puzzle Interaction

Sliding Puzzle

Specifically in the sliding puzzle, after all the piece *Prefabs* are instantiated, the last one, corresponding to the bottom-right corner, gets its color's alpha reduced to a low value and is subsequently deactivated, meaning that it disappears, in order to create the desired empty space. After that, all the pieces on the board get shuffled using an ad hoc brute

force algorithm that performs *size*³ moves using a function that swaps the empty space with an adjacent piece while also never swapping the same piece twice in a row.

For the purpose of allowing the player to move pieces and try to finish the puzzle, whenever the screen is touched, a Raycast⁴ is sent from the touched screen position directly to the far clipping plane. This way, it is recognized if the player clicked on top of a piece, and if so, which piece was clicked. In the event that the player in fact clicks on a piece, the same function used in the shuffling algorithm is called to try and swap it with the empty space. A piece can only be swapped with the empty space if it is adjacent to it.

Finally, the puzzle is completed when all the pieces are moved to their correct positions. Following that, the bottom-right piece, which was initially faded out and deactivated, gets reactivated and fades back to normal. The fade in animation is done using linear interpolation of the color's alpha value, again with the same characteristics as before.

A unique sound effect is played, using the AudioManager detailed in Subsection 4.1.5, whenever the player swaps a piece with the empty space and also when the player completes the puzzle.

This interaction can be visualized in (fig. 4.2), where on the left and middle there can be seen a shuffled puzzle and a completed puzzle, respectively. On the right of this figure, there can also be seen the final reward that is revealed at the end of all the interactions.



Figure 4.2: AR Sliding Puzzle Interaction and Final Reward

After Completion

In both puzzles, when they are completed, the timer gets stopped, and there is an explosion animation that disperses all the puzzle pieces on the board, revealing a final star reward

in its position. This explosion happens only after a set delay that allows the player to first have time to admire the constructed puzzle image. To achieve this explosion animation, first the reference to each puzzle piece on the board is stored in a list when it is instantiated during the puzzle generation. Secondly, at the time of the explosion, all pieces get assigned a *Rigidbody* component that provides them with physics-based behavior and allows for the simulation of forces and other properties. Subsequently, a large force is applied to all pieces from the direction of the center of the puzzle, making the pieces fly out in a radius, and an explosion sound is played from the explosion position using the *AudioManager*.

After the explosion, the reward *Prefab*, which is composed of a star model made in Blender and a particle system that forms a continuous shining visual effect, is instantiated in the same position where the middle of the board was located. Furthermore, it gets assigned a perpetual movement animation that makes it hover up and down, due to a *sine* function, and rotate around itself. This reward can be clicked on to be collected using the same method previously described that uses a *Raycast* sent from the touched screen position. Collecting this reward instantiates yet another particle system that pulses a star visual effect once.

Moreover, after each puzzle is completed, the interaction's information, which in this case corresponds to the time it took the player to complete the puzzle and the level/size of the puzzle, is both saved internally using the data persistence system and sent to the mission system to be processed. These systems are detailed in Subsection 4.1.5 and Section 4.2 respectively.

4.1.2 Quiz

The designed quiz interaction is composed of a question and its respective answer options, which include the correct answer, and allows players to visualize all its corresponding text using *AR*, choose a desired answer, and get feedback on it.

As all the text that establishes the quiz would be prompted by a user in the narrative creation interface and subsequently stored in the database, it essentially indicates that it needs to be downloaded from an external file system. Taking into consideration how the mobile component of the base platform is capable of launching the desired Unity scenes and the information it passes through to them using JSON, the text does not need to be requested from a URL, similarly to the previous interaction, since it would already be included in that said information. However, if this was not the case, a *UnityWebRequest* equivalent to the aforementioned could easily be performed to obtain this information, as the response has the option to directly access and convert the text to the string format. For simplicity, the required text is extracted from a single string containing the question and all answer options, separated using a special delimiter, and additionally having a special character in order to signalize the correct answer option.

In the normal course of the previous interaction, after the content is successfully loaded, the player is given the chance to position the board itself in a desired space. Consequently,

in this interaction, it was decided that the content would already be pre-positioned in the scene relative to the player's starting position. This way, there would be two different interactions with different positioning methods to be later tested and allow for player preference analysis. The player could either prefer to have the responsibility of positioning the interaction where it desires, or prefer to not have that burden and for the content to already be pre-positioned.

Generating the 3D text

After the required string is loaded, the main behavior controller for this interaction starts generating the three-dimensional text and positioning it according to how it was designed. Before instantiating anything, first, the string is separated into various other strings, which contain the distinct text clauses that form the quiz, by the defined delimiter, and after that, while iterating through the answer options, the correct answer is identified by a special defined character as well. Additionally, just for added effect, an uppercase letter from A to Z is introduced before each answer option, followed by a colon. Furthermore, a limit on the number of characters that would be written in a single line was established. Writing text on the next lines is achieved by iteratively dividing it into substrings of substrings if, after twenty characters, they still have a whitespace and more text to be written, while also shifting each substring's writing position downward.

Subsequently, these separated texts need to be instantiated within the scene. It was ultimately decided to present them using three-dimensional characters instead of plain two-dimensional ones for visualization using [AR](#). This is believed to help enhance immersion and realism while also allowing for more dynamic movement that includes, for example, rotations. Therefore, this makes three-dimensional text the preferred choice in this situation when trying to meet the requirement for a compelling interaction. A small hurdle to this decision is that Unity does not offer any built-in functionality to create three-dimensional text within a scene, thus some outside methods were explored. Although there were a few appealing and distinctive methods found, which would be usable, the one that was eventually determined to be implemented was the least complicated and most straightforward of the bunch and was imported as a Unity package⁵. This method requires a *Prefab* that includes all the intended characters' three-dimensional models as its different children, which all start deactivated. These models can be easily made using Blender by extruding text to the desired depth. This is a simple but somewhat laborious process that also bears the favorable characteristic of allowing for the later addition of special characters and also different fonts. To generate the three-dimensional text within the scene, this *Prefab* is instantiated while also serving as a text root and writing cursor. Then, for each character in the text, a switch statement evaluates it and instantiates its corresponding three-dimensional model at the cursor's position, which is then moved to the right according to a determined amount based on what character was written. Finally,

⁵[3DText Unity package in GitHub](#)

some adjustments are made to change the rotation and position of the entire text's pivot in order to simplify the subsequent steps.

Quiz

After the required text of both the question and the answer options gets generated in the scene as three-dimensional text, since it was decided that this interaction would have its content pre-positioned, they both need to be scaled and placed as designed in relation to the *AR Session Origin*, which represents the player's starting position in the [AR](#) scene. Firstly, the question gets positioned right in front of the player, about one and a half meters away. Secondly, each answer option gets positioned iteratively higher than the previous and gets assigned a random angle between 0 and 2π that is used in order to also position it around the origin in a radius of three and a half meters. The height of multi-lined texts is also taken into consideration. Their center's position, in relation to the origin, on the three axes is defined by the following expression:

$$Pos_{(x,y,z)} = Origin_{(x,y,z)} + (\sin(angle) * radius, height * (i + 1), \cos(angle) * radius)$$

As also determined during the design phase, although the question is stationary, the answer options are meant to orbit in a circumference around the origin. To achieve this, since their positioning at the start already took this aspect into consideration, only their movement needs to be handled. During their positioning, each answer option also gets assigned a random rotation speed, so as to make their animation more interesting. After that, their rotation around the origin is achieved by increasing the angle that was assigned during their positioning while taking into consideration the mentioned rotation speed and the time between frames, so the consequent animation is frame rate independent. Furthermore, their rotation around themselves is also changed in order for them to always be facing the origin.

For the purpose of allowing the player to select an answer option, whenever the screen is touched, a *Raycast*⁴ is sent from the touched screen position directly to the far clipping plane. This way, it is recognized if the player clicked on top of any answer option, and if so, which one it was. Differently from the previous interactions, both the normal and the [AR](#) specific *Raycasts* were noticed to not be accurate enough, where sometimes a registered touch, which was clearly on top of an answer option, would not have its *Raycast* return any hits. To solve this, a *Spherecast* is performed instead. This way, even with a very small radius, it sends out a wider ray that always hits the desired clicked text.

After selecting any answer option, an animation that reinforces the immediate feedback of the quiz is carried out in order to inform the player whether it was the correct one or not. This animation begins, regardless of whether the answer is right or wrong, by moving the selected text on top of the question and facing it the same way as well. This is accomplished by using linear interpolation between the current position of the text and a calculated position above the question, based on the height and number of lines in the answer. During this movement of the answer, a spherical linear interpolation is also being

performed for its rotation, so that the consequent animation ends with the answer facing the right way. After this first animation is finished and after a short delay that allows the player to reread the text while also adding in some suspense, there are two possible outcomes. In case the selected answer is the correct one, then all the texts still present in the interaction explode in a sequence, and the final reward is revealed at the question's position, finishing the quiz. On the other hand, in case the selected answer is not the correct one, only it explodes and a red border blinks around the screen by changing an image color's alpha value to one and back to zero, and, after that, the quiz carries on. Since it was determined that the player would have two opportunities to select the right answer, the quiz also ends after the player selects two wrong answers. In this case, after the last wrong answer selected explodes, the right answer does the aforementioned animation that moves it atop the question, maintaining the relevancy of the quiz as an effective learning interaction. The reward instantiated at the end and the method used to perform the explosion animation both have the same characteristics described previously in the puzzle interaction.

A unique sound effect is played, using the AudioManager detailed in Subsection 4.1.5, whenever the player either selects a right or a wrong answer. Moreover, after the quiz is finished with a correct answer, the interaction's information, which in this case corresponds to the number of tries the player took, is sent to the mission system, described in Section 4.2, to be processed.

This interaction can be visualized in (fig. 4.3), where it is shown a landscape view of the question and answer options from far away so as to better capture them.



Figure 4.3: AR Quiz Interaction - distant view

4.1.3 Balloon Pop

The balloon pop mini-game is composed of a portal, which continuously makes ascending balloons appear from it, and a dart that can be thrown at them in order to pop them, all visualized using AR. Differently from the two previously explained interactions, this one does not require any content to be loaded from an external file system. Therefore, the first

event in the normal course of this interaction is regarding its positioning. At this stage it was considered that letting the player position the content at its will, where it desired within any detected space, was the better solution. Consequently, the player then has to place the portal in its surroundings using the feature that is later detailed in Subsection 4.1.5. After that, the main behavior controller for this interaction makes a timer appear on top of the screen that starts counting forward from zero, while at the same instance, the balloons also begin appearing from the portal and going upwards. This timer is also the same used in the aforementioned puzzles, for consistency, and is later detailed in Subsection 4.1.5. Additionally, after the portal is positioned by the player, a horizontally flat particle system, which forms a blue and purple vortex with rays coming out of it, is also positioned at its location in order to both allow the player to better identify the defined area where the balloons will be appearing and increase the mini-game attractiveness.

Spawning the Balloons⁶

It was determined during the design phase that there would be different balloons appearing from the portal with distinct properties, since it adds more dimension to the game without making it less intuitive. To achieve this, a set of six unique balloon models was taken and used from the Sketchfab website. Each balloon was subsequently processed by creating its respective *Prefab* that included all the components necessary. These components consist of the *MeshRenderer* for applying the desired materials, a set of *Colliders* that could cover the balloon entirely, a *Rigidbody* to deal with its ascent, and a properly positioned pivot to facilitate some steps afterwards. The usage of the *Colliders* and *Rigidbody*s also allows for the handling of collision events, making the balloons' behavior as close as possible to reality when they, at any determined point, go against each other, either vertically or horizontally. The spawning of balloons is dealt with using a specific type of function, denominated *Coroutine*. Unlike normal functions that execute their code within one frame and move to the next instruction, *Coroutines* allow for a piece of code to be executed across multiple frames while also allowing the use of yield statements, which return an enumerator object value that makes it possible to, for example, wait for a number of seconds or for the next frame before continuing to execute code within an iteration. These functions are not to be confused with asynchronous ones, as their execution is still synchronous. With this in mind, the balloon spawning function has a loop that keeps making balloons appear until the mini-game ends. In each iteration, first, a random interval is calculated between half a second and two seconds that will make the function wait before proceeding to spawn a balloon. Secondly, one of the six balloon *Prefabs* is randomly chosen to be instantiated and is so randomly within the circle of the portal. This is done by using a random angle's sine and cosine, along with a random radius between zero and the radius of the portal, to determine its position. During this step, the rotation of the instantiated balloon is also changed for the purpose of having it face in the player's

⁶Spawning is, in a gaming context, a slang term that refers to the process by which characters or entities are generated or appear within the game world.

direction. Finally, in order to make the spawned balloon ascend, its *Rigidbody* gets the gravity calculations turned off and its rotation due to physics frozen on the X and Z axes. The latter makes the balloons not rotate in any abnormal way whenever they collide with each other. Subsequently, a positive force on the Y axis is applied to the *Rigidbody* based on the desired ascending velocity. There is a flag present in the code that allows to choose whether the desired velocity for each balloon comes from a predetermined value, which can be unique for each of them, or a random value within a range. Lastly, a reference to the instantiated balloon *Prefab* is added to a list, and a function is invoked after fifteen seconds to automatically pop that balloon so it does not go out flying forever.

Popping the Balloons

The number of balloons a player is required to pop in order to complete the mini-game is the designed dynamic factor that enables it to be **difficulty scaled**. Therefore, much like the puzzles, the file from the internal file system containing the game's results is loaded, and the average of the last ten times that the player took to complete this mini-game is used to calculate whether the also loaded previous level stays the same, increases, or decreases. The level of the mini-game corresponds to the target number of pops divided by five. This was mainly due to it being decided that, to be consistent with the puzzles, the starting level would also be three and that the first time a player would be going through the interaction it would have to pop fifteen balloons. Consequently, whenever a player has an average of times superior to 90 seconds, then the level of the mini-game decreases by one, having to pop five less balloons, and on the contrary, if a player has an average of times inferior to 30 seconds, then the level increases by one, having to pop five more balloons. Analogously to the puzzles, the minimum level is two and there is not a maximum level. In the interest of displaying this challenge goal to the player, a UI element is included at the top left of the interface, near the timer. It exhibits both the current amount of balloons popped during the mini-game, which starts at zero, and the target number of balloons to pop, respectively, separated by a forward slash. The left portion of this UI element is, as expected, incremented every time the player pops a balloon.

In order for the player to pop the balloons, it is given the ability to throw darts at them. This is achieved by having a dart positioned in the bottom center of the screen close to the camera, and when the player touches the screen, the dart is thrown forward in a trajectory that passes through the middle of the screen. Several factors were taken into consideration when implementing this, and also because, by design, the player does not define the direction nor the force of the throw through a swipe motion, its configuration needs additional attention so as to feel satisfying from any distance. To start of, it was determined that the player would be limited to throwing only one dart per second, meaning the darts have a reload time of one second. This was done in order to better control the flow of the interaction and to scale the difficulty of both the interaction and its correspondent missions. When reloading a dart, its respective *Prefab*, which includes the dart model and also all the components aforementioned in the balloons' case, is instantiated at its

determined location as a child of the camera, which represents the camera of the device. This makes the dart anchored to that position, moving and rotating it in coherence with the device. An arbitrary position that is relatively far from the device's position and is always aligned with the center of the screen was adjusted throughout many experiments to use as the darts' throwing target. The loaded dart, while anchored to the camera, stays pointed in that target's direction. Whenever the player touches the screen, if a dart is loaded in, then this dart stops being anchored to the camera and is thrown in the direction of the target. The throw is performed by applying a three-dimensional force to the dart's *Rigidbody* according to these expressions:

$$\vec{u} = Target_{(x,y,z)} - Dart_{(x,y,z)} \quad \hat{u} = \frac{\vec{u}}{\|\vec{u}\|} \quad \vec{F}_{applied} = \hat{u} * F_{throw}$$

Additionally, to make the dart's falling motion as realistic as possible, two approaches were tested. The first uses the physics calculations of the *Rigidbody* and includes a scrip that shifts the center of mass of the dart to be closer to the front, where the metal section is located. The second redirects the dart's orientation in each frame to always be pointing towards where it is moving. Both approaches apply a gravitational force to the dart, causing it to descend into the ground. The second approach was observed to produce a more believable result and is consequently the one used. The dart is then, in every frame, oriented towards $Dart_{(x,y,z)} + \vec{v}$, where $\vec{v}$ represents its velocity vector. Moreover, much like the balloons, the thrown darts are also destroyed after fifteen seconds to avoid cluttering and to not negatively affect performance. During the trajectory of the dart, any balloons it passes through get popped, removing them from the scene, incrementing the aforementioned current number of pops, briefly instantiating and removing a particle system that forms a pop visual effect, and playing a pop sound effect using the *AudioManager*. The dart is considered to have passed through a balloon whenever both the balloon and dart *Colliders* intersect.

As a means to help the player estimate in which direction the dart is going to be thrown, an animated crosshair is included as an UI element in the center of the screen. The animation is performed by constantly rotating the crosshair around itself and also continuously increasing and decreasing its size using linear interpolation based on a *sine* function.

After the player finishes the mini-game by popping the target number of balloons, the balloon spawner stops making new balloons appear, the portal's visual effect is slowly removed, the timer also stops, and all the balloons still present in the interaction are popped in a sequence using the same visual and sound effects previously mentioned. Furthermore, the final reward *Prefab*, which has the same characteristics described previously in both the puzzles and the quiz interactions, gets instantiated at the device's height atop the portal's position. Finally, the interaction's information, which in this case corresponds to the time it took the player to finish the mini-game and the level/number of balloons to pop of the mini-game, is both saved internally using the data persistence system and sent

to the mission system to be processed. These systems are detailed in Subsection 4.1.5 and Section 4.2 respectively.

This interaction can be visualized in (fig. 4.4).



Figure 4.4: AR Balloon Pop Interaction

4.1.4 Geospatial Flyers

This interaction allows players to engage with floating flyers that are positioned atop various POIs. Each of these flyers contains a set of images, interesting information, and a quiz relative to its corresponding POI. Their positioning is dealt with using Geospatial Creator for Unity, which makes it possible to directly anchor these flyers using coordinate-based positioning within the real world. Similarly to the previously explained interaction, this one does not require any content to be loaded from an external file system due to its design decisions explained in Chapter 3. This interaction was designed with the intention of being independent from the itineraries or to possibly be complementary to a AR navigation feature in the future. Consequently, the content used in each flyer is not prompted by a user in the narrative creation interface. Instead, the necessary content gets added onto a flyer within Unity and this flyer gets anchored in its respective coordinate-based location manually. However, it is simple to modify this implementation so each flyer's content does come from an external source, including its coordinates, so it gets positioned in the scene.

Creating the Flyers

As a means to offer a straightforward procedure for creating new flyers within Unity, a default flyer *Prefab* that can be minimally changed to generate these was created. To achieve this, first, there needed to be a three-dimensional model for said flyer. Since there were not any models found that suited the originally thought idea of having different pages and the ability to be opened and closed, a new model was created using Blender. Furthermore, this model had its opening animation and UV mapping also done within Blender. Each page is a two-faced plane, and the animation is performed by rotating each page sequentially while having the center of the flyer as a pivot point. Even though the original animation includes a *Deform* property on each plane as a means to give it a more realistic look, this is not supported by Unity, and therefore it cannot be imported into it by default. A specific Blender external plugin was tested to deal with this problem, which basically exported the animation information for each mesh point and reimported it as a distinct animation, but an acceptable result failed to be achieved. Thus, the pages' *Deforms* were not carried to Unity, resulting in them being flat. A texture template, which is composed of bordered images, texts, and titles relative to a certain [POI](#), was made using Photoshop for the purpose of facilitating the creation of further flyers' textures. The model's pages' UVs were subsequently mapped into this texture's layout. The created model and its characteristics within Blender can be better visualized in (fig. 4.5). The final flyers do not have blank middle pages, like in this figure, as their textures have an image on the far right as well.

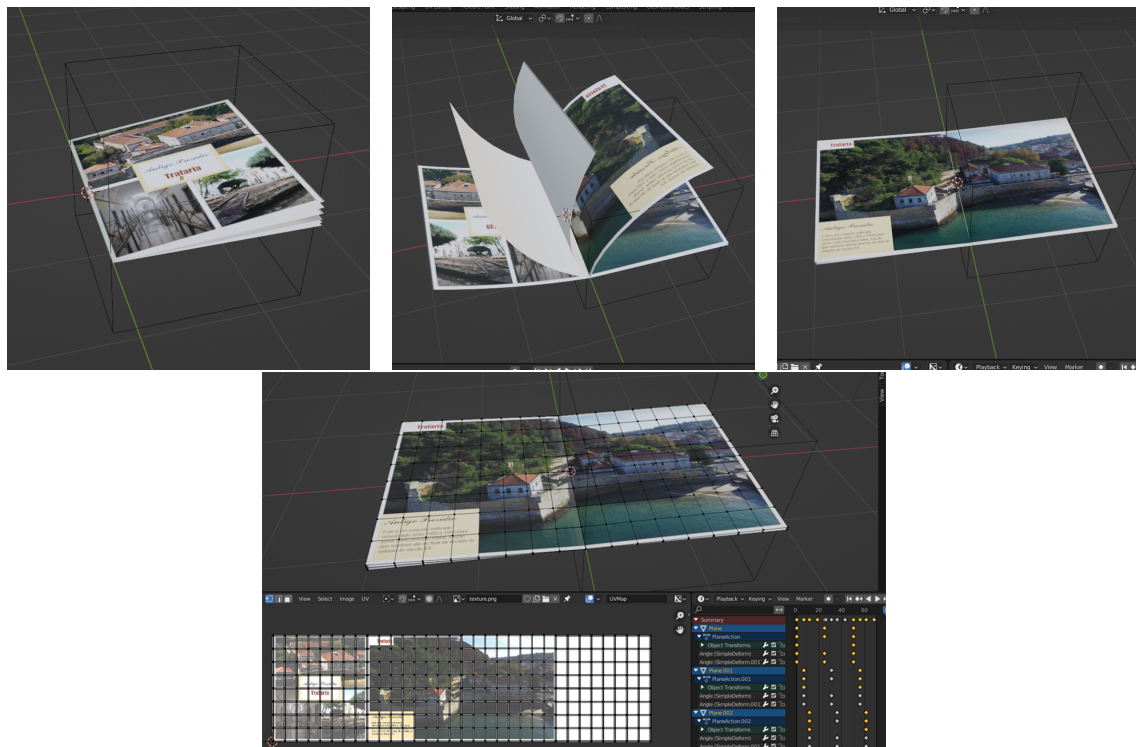


Figure 4.5: 3D flyer model created, animated, and UV mapped using Blender

Secondly, the mentioned default flyer *Prefab*, in addition to the three-dimensional model, also includes all the elements that form the quiz inside the flyer and a script that controls its behavior. This script gives the possibility of writing a question and three answer options within the Unity inspector⁷ while also updating the quiz in editor mode for better visualization. Moreover, the texture can be assigned to the entire model by adding it in the inspector as well. Finally, a checkmark was also added in the inspector after each answer option in order to identify which one is correct. These checkmarks were implemented to be mutually exclusive to help avoid mistakes.

In summary, in order to create a new flyer, a copy of the default flyer needs to be made. Then, the question and answer options need to be written in their corresponding text boxes within the copy's inspector. Finally, the correct answer option needs to be checkmarked, and the texture, which can be created using the template, needs to be assigned, both in the copy's inspector as well.

Additional Setup

In order to start using Geospatial Creator in Unity, two external packages need to be included in the project: ARCoreExtensions for Unity and Cesium for Unity. These packages were both previously described in Subsection 3.4. After that, several project settings also need to be adjusted⁸. From here, a new Google Cloud project needs to be setup by enabling the necessary APIs. These are the ARCoreAPI and the MapTilesAPI. Subsequently, a API key for that Google Cloud project needs to be generated and added to both the ARCoreExtensions in the project settings of Unity and the Cesium Georeference component. This component is present in a mandatory *GameObject* that needs to be instantiated into the scene and is identified as *AR Geospatial Creator Origin*. This *GameObject* serves as the reference for other elements in the scene and is responsible for converting the real-world position of anchors into game space coordinates while also loading the three-dimensional map tiles for visualization in the editor. Furthermore, other mandatory components also need to be included in the scene, such as the *ARCore Extensions*, *AR Anchor Manager*, *AR Earth Manager*, *AR Streetscape Geometry*, and finally a script that deals with the request for the device's GPS permissions. From this point on, each flyer can be manually positioned by first adding a *AR Geospatial Creator Anchor* component to it and prompting its real-world coordinates. Moving the flyer within the Unity scene automatically updates the anchor's real-world coordinates as well. At the same time, the three-dimensional representation of the map in the editor helps to better position each flyer without having to worry about its exact coordinates.

Interaction

With the flyers already anchored in their POIs' locations, a flyers manager component was

⁷Unity's Inspector Window is a view within the editor that allows a user to edit the properties of almost any Unity component, including the ones associated with GameObjects/Prefabs.

⁸[Geospatial Creator Quickstart for Unity](#)

created. This component is responsible for hiding or showing flyers based on their distance to the player and also based on whether their respective quiz was already answered or not. The first is achieved by periodically invoking a function that checks for nearby flyers by using a stationary *Spherecast* with the desired radius. The second is achieved by using the data persistence system to save and load a file containing the IDs of answered flyers. In order for the player to know in which direction to walk to get to each available flyer, a compass is included as a UI element at the top of the screen. The compass displays all four cardinal points along with a symbol for each available flyer and its respective distance. The position of each flyer's symbol within the compass and its distance to the player is calculated using the following expressions:

$$\begin{aligned}\vec{d} &= \text{Flyer}_{(x,y,z)} - \text{Camera}_{(x,y,z)} \quad \text{distance} = \|\vec{d}\| \\ \theta &= \text{atan2}(\vec{d}_{(x,z)} \times \text{cameraForward}_{(x,z)}, \vec{d}_{(x,z)} \cdot \text{cameraForward}_{(x,z)}), \quad -\pi < \theta \leq \pi \\ \text{clampedPos} &= \min(1, \max(\frac{2 * \theta}{\theta_{fov}}, -1)) \quad \text{Pos}_x = \frac{\text{compassWidth} * \text{clampedPos}}{2}\end{aligned}$$

Whenever a player gets near an unanswered flyer, this flyer gets revealed, and the player is then able to click on it to open it, triggering the opening animation. The click action is once again evaluated by performing a *Raycast* from the touched screen position. While a flyer is visible and closed, it is animated by constantly hovering up and down, using linear interpolation based on a *sine* function, and rotating over itself. While a flyer is visible and opened, it gets assigned a billboard effect so as to always be facing the player and allow for a closer view depending on the direction the player is looking.

Upon opening a flyer, the player is then able to read and try to answer its respective quiz. The flyers' quizzes are fully composed of UI elements, and their answer options are buttons that the player can click on. Naturally, the buttons can only be clicked when the flyer is in the open state. After choosing an answer option and clicking on its respective button, the buttons change color based on the correct answer. If the player clicks on the correct answer, then its button turns green. On the other hand, if a player clicks on the wrong answer, then its button turns red, and after a short delay, the correct answer's button turns green. This way, immediate feedback on answers is enforced, and the quiz maintains its relevancy as a learning interaction. After answering a quiz, its ID gets saved in the answered flyers file, and therefore its respective flyer disappears after a delay.

This interaction can be visualized in (fig. 4.6), where there can be seen on the left a closed flyer, in the middle the opened flyer information, and on the right the opened flyer quiz after choosing a wrong answer. Additionally, all the created flyers can be seen in Appendix C.

4.1.5 Miscellaneous

Some additional minor features, which are common to more than one interaction, were also implemented, and they are explained below:

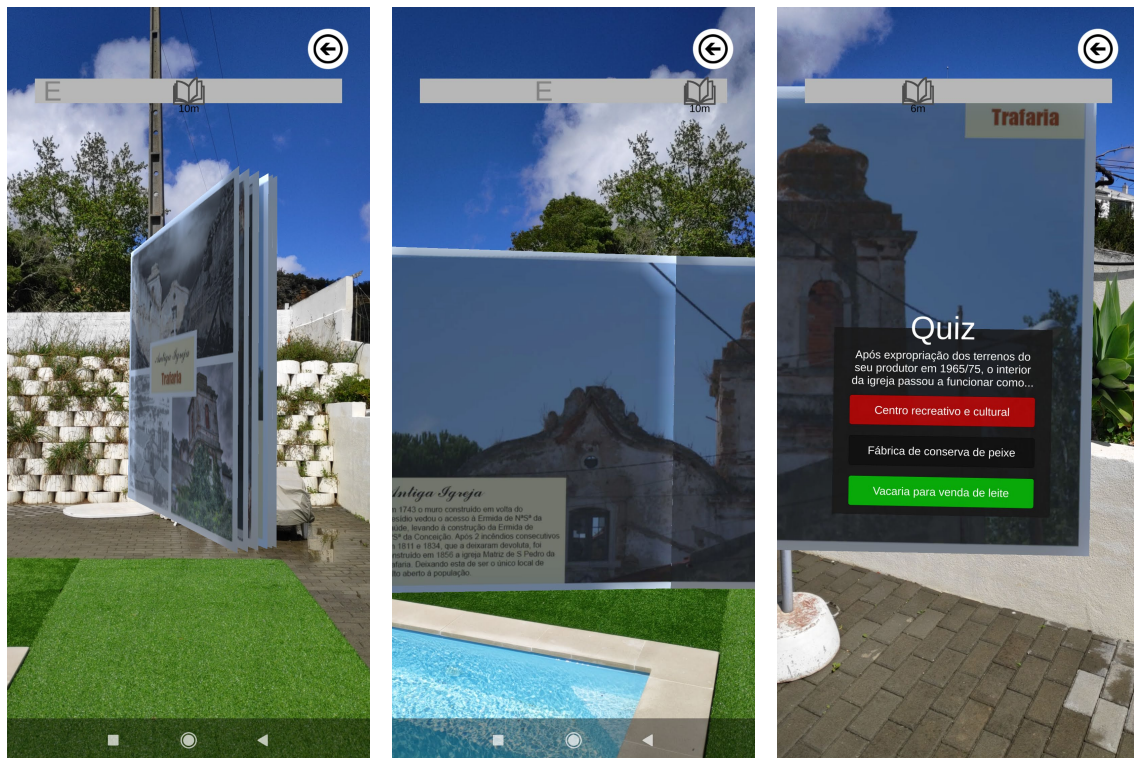


Figure 4.6: AR Geospatial Flyers Interaction

Audio Manager

This feature contains a script with a *Singleton* class. *Singleton* is a design pattern that makes the class have a singular static instance present at any moment, which is therefore globally accessible from any other script. This class is used as a container for all the sound effects that can be played across the whole project. Each sound effect has a name, a correspondent audio clip, a defined volume, a loop toggle, and a repetition timer. Thus, this feature offers other scripts the possibility to only call one function in order to play a sound effect by name. Furthermore, the sound effect can be either played normally on top of the player or at a determined world position.

Interaction Positioning

This feature is achieved by using a standard *AR Tap to Place* procedure that requires the *AR Session Origin* to contain both a plane and raycast managers. It essentially starts by detecting planes in the environment captured by the device's camera and allows the player to visualize a placement indicator on top of these planes where the player might be pointing the device. After tapping the screen, the interaction is placed in the indicator's position and this feature gets disabled.

Debug Logger

This feature was implemented in order to write relevant information on the screen during

testing on a mobile device. Each log is written in a new line within a *ScrollView*³, allowing navigation through all the logs. Furthermore, the texts of each log fade in and out depending on whether new logs are being written or if the user is scrolling through them.

Information and Instructions Foldout

This feature is composed of a UI button positioned in the top right corner of the screen that serves as a toggle for a large *ScrollView*³ centered in the screen, which forms the desired foldout effect. This button, when the foldout is closed, has a standard *Info* symbol, and when it is open, has an *X* symbol. The player can also close the foldout by clicking anywhere outside its perimeter. Some other features stop reacting to player's screen touches whenever the foldout is open. This is done to avoid any undesired behavior and in order for the player to feel in control. Each interaction has its own content in its respective foldout *ScrollView*, which includes their specific step-by-step instructions. Additionally, puzzles display a preview of their image, quizzes present a preview of the interaction itself for easier comprehension, and the balloon pop has a list with a preview of all the balloons that can appear from the portal. The three mentioned foldouts can be better visualized in (fig. 4.7).



Figure 4.7: Information and Instructions Foldouts - Puzzle / Quiz / Balloon Pop

Task Completion Popup

After an interaction sends its specific information to the mission system to be processed, there is the possibility of certain tasks being completed at that instance. Therefore, whenever a task gets completed during an interaction, an animated UI element slides in from

the right, displaying "Task Completed!" as a message to inform the player of this event. After a brief pause, this UI element slides back out of the screen.

Timer

This feature is composed of a text UI element that uses a digital-style font. It has a behavior controller script that allows for its standard utilization as a clock, chronometer, or countdown timer. Both the puzzles and the balloon pop interaction use it as a chronometer.

Back Button

The standard back button is in the top right corner with a back-arrow symbol, which allows the player to exit the interaction at any given moment, so it feels in control.

Data Persistence System

This system is responsible for handling the saving and loading of files to and from the internal file system of the device. The directory path where the files are saved into is retrieved from a function that returns the device's default application persistent data path. Additionally, the specific file name and the directory path are combined using a function that also uses the device's default delimiter. This is the conventional approach, and it makes the saving and loading of files independent of the device the system is running on.

Each file has an associated class, which contains all the desired variables to be persisted, and inherits from the abstract class *Data*. The system then has a static class that offers globally accessible functions that save and load these files. These functions are of the generic type *T* that is required to be from *Data* inheritance. The data classes are serialized and deserialized to and from JSON. Two used variable types that are not serializable by default had a custom version implemented by using the *ISerializationCallbackReceiver* interface, which allows to write a function that is called before serializing the class and another after deserializing it. The first type is the Dictionary that is converted into two separate lists, one for the keys and other for the values. The second type is the Sprite that is converted into a texture that is then encoded to png bytes.

4.2 Mission System

The mission system developed can be divided into three components. The first component is the mission graph creation tool, which offers built-in interfaces to easily create and manage, without writing code, all of the elements necessary to produce the missions. The second component is the mission graph interpreter, which draws on the information from the elements created using the tool to generate the actual missions, manage them, and deal with their progress' tracking. Finally, the third component is the missions interface, which is responsible for displaying all the desired information to the player while also allowing it to advance within the missions by collecting the finished ones.

4.2.1 Mission Graph Creation Tool

During the design phase described in Subsection 3.3.2, it was determined that, in order to generate the desired missions, three main elements sufficed. A **graph** in which each node represents one mission, the **tasks** that are required to be performed in each mission, and the **rewards** that the player receives after completing each mission.

Consequently, two separate interfaces were made for each of these elements. The first is for their management, and the second is for their creation. All six interfaces were implemented using Unity's UI Toolkit⁹, which basically allows to override how certain serializable objects get displayed in the inspector⁷ while also enabling the creation of independent windows and the integration of all available UI elements into them.

The management interface windows created are very similar to each other. They are the main interfaces that allow to visualize, edit, delete, and open the creation window for each of the three aforementioned elements. Because of this, they are also introduced into Unity's own Windows menu in order for them to be opened from there. Each management window is separated into two sections. The first section, visualized on the left, presents a list with all the existent created elements of its respective type. The second section, visualized on the right, starts by being empty and prompting the user to select an item from the first section's list for it to be displayed. After selecting an element of a determined type on the list, it is then displayed in the right section. This allows for the editing of this element and offers the option to delete it as well. Additionally, in the left section, there is also a button to create a new element of that determined type, which, when clicked, opens the respective creation interface window. One of the manager interface windows can be visualized in (fig. 4.8), where it shows the view after selecting an element from the list. The other two manager interfaces are very similar to this one, since they have the same structure and components but only display a list and a view of different elements.

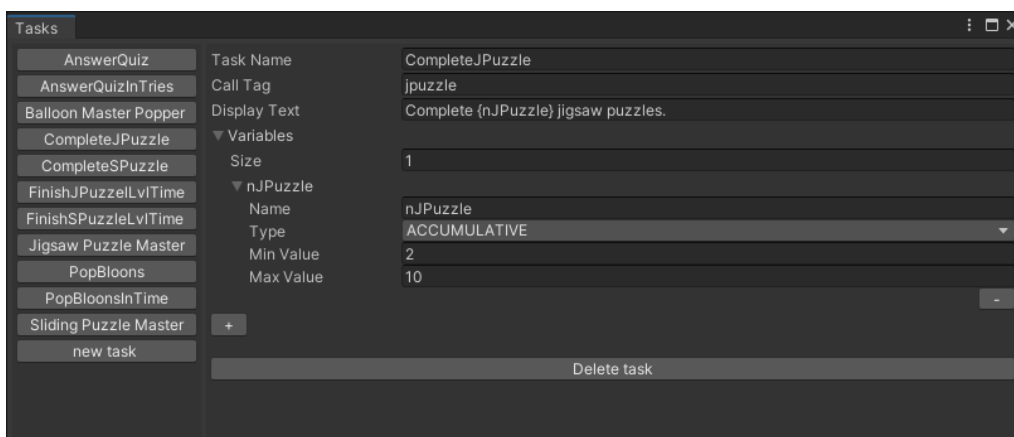


Figure 4.8: Tasks Manager Interface

Finally, another similar aspect of the three elements is that after they are created, they get their information stored using *ScriptableObject*s. These are assets saved in the project's

⁹Unity UI Toolkit Documentation - last accessed: 03/2024

database that serve as great data containers that are independent from class instances.

Tasks

The task creation interface allows to write the name, calltag, and display text of a task in their respective text boxes. The name of the task does not need to be unique because when a task is saved, if there is already a task in the asset database with the same name, then it gets assigned a new standard name using the number of tasks created (Task#). The calltag of the task is the value used when sending information to update tasks, and it serves to allow for task type grouping so the mission system does not try to update all active tasks, thus making it more efficient. The display text of the task is the text used in the interface to inform the player of the task's objective. It allows to also reference variables in it, so their value will show up in the missions interface as well.

Moreover, the task creation interface also has a section to specify the necessary variables in a list for the task's functionality. As described in Subsection 3.3.2, there are two types of variables: Accumulative and Requisite. In order to allow a user to insert these variables, choose their type, and have their prompts change depending on their type, a *Custom Property Drawer* had to be implemented. There is a button that adds new variables to the visualized list, and each variable includes a button to remove it from the list. Each variable has a text box for its name and a drop-down that allows to choose its type. Accumulative variables also have a prompt for their minimum and maximum values, and requisite variables have a prompt for their value and check boxes to choose which type of limit they represent. These variables are implemented as subclasses of an abstract class *TaskVariable* and is relatively simple to later add new variable types if necessary. Furthermore, a quality of life button is also included to automatically add all variables between brackets in the display text to the variable list.

The task creation interface, with an example task inserted into it, can be better visualized in (fig. 4.9), where it can also be seen that it includes some informatory texts in each section.

Rewards

The reward creation interface allows to directly insert a desired display image to represent the reward in the missions interface. This was done with a Sprite property field input. Another *Custom Property Drawer* was implemented in order for the Sprite chosen by the user to display a preview on the creation interface as well. Furthermore, for the reward's collection, an event pattern was used. It was thought that a well-implemented game would usually already have functions to deal with the collection of items or the increment of in-game values like points, gold, etc. Therefore, the event pattern in Unity offers input for various event listeners, each allowing to choose a script and having a section to select a function from that script and prompting the necessary parameters for it. Whenever the reward gets collected, this event is triggered, and all the listeners are called.

The reward creation interface, with an example reward inserted into it, can be better visualized in (fig. 4.10).

New Task

Task Name
It would be best if the task name were unique. Although not mandatory.
FinishSPuzzleLvTime

Call Tag
Remember this value. It's used to reference different tasks when trying to update them.
spuzzle

Display Text
If you want to show variables in the display text add them in their place between ().
Finish a level {level} or higher sliding puzzle in under {time} seconds.
You need to at least have the display text variables in the list (with the same name).
You are free to add other variables.
Remember their name to reference them when trying to update the task.
Add display text variables to list

Variables

Size
3

level
Name
level
Type
REQUISITE
Requisite
3
Less Than
☐
Equal
☒
More Than
☒

time
Name
time
Type
REQUISITE
Requisite
15
Less Than
☒
Equal
☐
More Than
☐

nSPuzzle
Name
nSPuzzle
Type
ACCUMULATIVE
Min Value
1
Max Value
1

Save task

Figure 4.9: Task Creation Interface - filled out with an example task

New Reward

Script
Reward
Reward Name
MockReward
displayIcon
pointsrewardicon

On Collect ()
Runtime Only
MissionSystem.MockRewardCollect
MissionSystem (MockReward) 50

Save reward

Figure 4.10: Reward Creation Interface - filled out with an example reward

Mission Graph

The mission graph creation interface uses Unity's Experimental *GraphView* in order to allow the user to zoom in and out, move the view, add or remove nodes, drag nodes, and create or delete edges between nodes. The interface includes a name text box and a button both in a toolbar. The latter allows to add new nodes to the view by clicking on it. Each node, which represents a single mission, then has a prompt for its name, the necessary ports for linking sequential nodes, a list of tasks, a list of rewards, and checkboxes to determine if they are normal, a lock, and/or a key. In case the user determines that a certain mission is a lock and/or a key, new ports are displayed in order for the user to link the lock nodes with their respective key nodes. The graphs are saved as a list of edges, which includes both the sequential links and lock and key links, and a list of nodes, which each include all their respective mission's information. The nodes additionally save information about their position within the view in order for the graph to be reconstructed correctly when the user goes to edit it later.

The mission graph creation interface, with an example questline inserted into it, can be better visualized in (fig. 4.11).

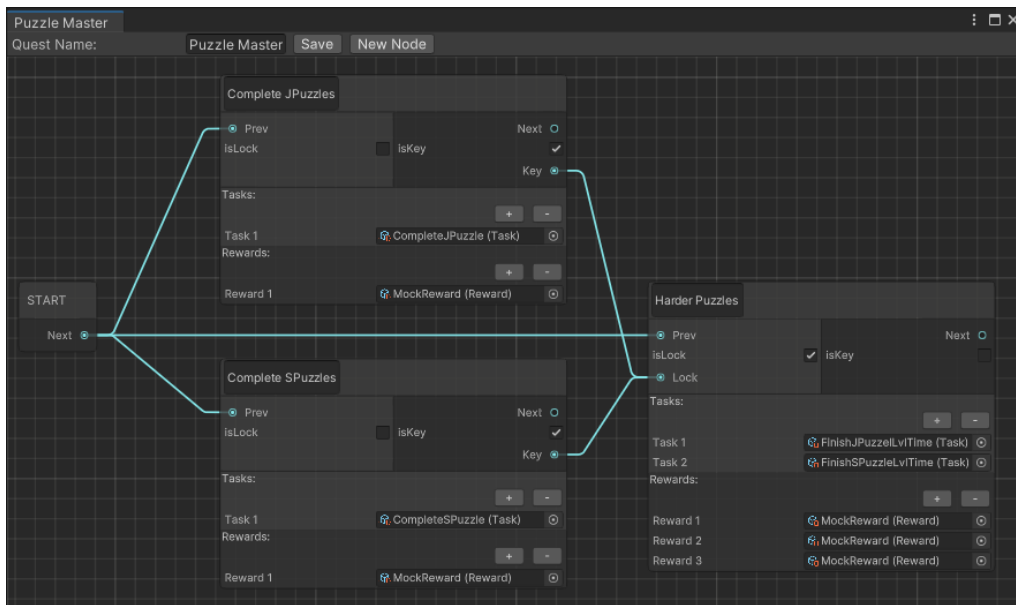


Figure 4.11: Mission Graph Creation Interface - filled out with an example questline

4.2.2 Mission Graph Interpreter

The mission graph interpreter offers the possibility for users to choose which of the created mission graphs are used for the active questlines and the daily quest, while also allowing them to choose which tasks are used for the achievements. Since, as previously described, these elements are stored in *ScriptableObjects*, this is achieved by having lists for their insertion in the inspector. The daily quest also prompts the hour at which it will be

reset every day, and the achievements also prompt a Sprite for their icon displayed in the missions interface.

The interpreter is then responsible for converting the information in these *ScriptableObjects* into the actual quests and achievements, which get stored in the internal file system using the data persistence system described in Subection 4.1.5. Subsequently, the interpreter also tracks all of the active tasks' progress. This is accomplished by using the *Singleton* pattern that allows to have a singular static instance of the interpreter present at any moment, which is therefore globally accessible from any other script. Then, it offers a progress public function that receives a calltag and a list with a variable number of parameters that contains all the necessary information to update the desired tasks. The tasks get updated internally, meaning that the interpreter sends the mentioned list to the tasks corresponding to the correct calltag. Inside the task, firstly, all the requisite variables are checked. A task does not progress if there is any requisite variable missing or if a requisite variable is not within the specified limits. Secondly, if all requisite variables were met, then the accumulative variables present in the list get incremented into their respectives in the task.

Whenever a task gets completed, it is checked if all tasks within the same mission are also completed, and if so, the mission gets completed as well. This changes the mission's view in the interface and allows the user to click on a button to fully finish the mission and collect its rewards. Whenever a mission gets finished, all its rewards get their respective collection events triggered. After that, the interpreter tries to activate any existing sequential missions that were linked to the finished one, taking into consideration not to activate missions that are already activated. Furthermore, if the finished mission was a key, then the interpreter also tries to unlock its corresponding lock missions. A mission only gets unlocked if all its respective key missions are finished. Furthermore, every time a task gets activated, its accumulative variables' final values are calculated between their minimum and maximum values based on the player's skill level, which uses the averages between all the difficulty scaled interactions.

4.2.3 Interface

The developed missions interface allows players to visualize all active missions, their respective tasks, and also the player's progress in those tasks. Additionally, players are allowed to finish completed missions and collect their rewards by clicking a button. The interface presents three pages. The first is for the daily quest, the second is for the main quests, and the third is for the achievements. The first two pages, which present the quests, also allow players to open and close quest's containers as a folldout, displaying only their headers when closed, as an option for better organization. The player is able to switch between the three pages by either swiping left and right on the screen or by clicking on each tab located at the top of the screen. The structural design of the interface was, as described in Subsection 3.3.2, inspired by another game. However, its visual appearance

took into consideration the base platform's color scheme, textures, and icons in an effort to effectively blend in.

The three pages are positioned inside a horizontal *ScrollView*³ and each of the pages has its own vertical *ScrollView*. Since the horizontal one is meant to switch between pages and Unity only supports unrestrained scrolling by default, a script was included to adjust the scrolling experience. This script snaps the more visible page to the middle position when the player stops swiping. The snapping is done smoothly by using linear interpolation.

As Unity does not support stacking *ScrollViews* on top of each other by default, since the one in front blocks the engagement with the one on the back, a nested scroll solution was implemented. This solution essentially transmits the scroll's touch information to all parent *ScrollViews* whenever the player tries to scroll in an orientation that is not the same as the touched *ScrollView*. This way, the player is able to scroll horizontally through the pages even though it is touching their vertical scrolls.

All the main components present in the interface have their own independent *Prefab* that includes both a positioning layout and a content size filter. The first adjusts the positioning of each item in its respective container so it is aligned and correctly spaced in relation to other items. The second automatically changes the container's dimensions depending on the size of the items currently in it. This means that when these components are instantiated at runtime, inside their respective containers, all the content gets dynamically aligned and spaced correctly.

The interface can be better visualized in (fig. 4.12) and the mentioned main components that compose it are briefly described below:

- **Questlines** are instantiated in the page's container. They have a header with the questlines's name and an arrow. When the header is clicked, it opens or closes the questline's container, also turning the arrow. When a questline is finished, it only displays the header and a check mark instead of the arrow.
- **Missions** are instantiated in the questline's container. They only present their name. When locked, they additionally present a lock above them.
- **Tasks** are instantiated in the mission's container. When in progress, they are blue and present the objective text and a completion percentage. When completed, they turn green, and the percentage turns into a checkmark.
- **Rewards** are instantiated in a banner's container, which is instantiated in the mission's container after all the tasks. When in progress, the banner is blue. When completed, the banner turns green and displays a button that, when clicked, finishes the mission.
- **Achievements** are instantiated in the page's container. When in progress, they are blue and present the title, objective text, completion percentage, and an icon. When completed, they turn green, and the percentage turns into a checkmark.



Figure 4.12: Missions Interface Pages

EVALUATION AND RESULTS

This chapter starts by presenting the evaluation methods used and some studies that helped formulate them. Following that, the usability tests performed for both the main solution and the Unity tool have their results presented, analyzed, and discussed.

5.1 Evaluation Methods

Since the primary goal of this work involves increasing users' engagement, motivation, and satisfaction regarding the platform, it is especially important to perform usability tests when evaluating the different solutions.

This section describes the protocol followed during the performed usability tests, along with a brief analysis of some research studies that helped design these tests and their associated questionnaires.

5.1.1 Protocol

Some studies were analyzed in order to help define the final structure of the usability tests that needed to be performed. All while taking into consideration this work's scenario of being an [AR](#) application in the context of cultural heritage.

A. Dunser and M. Billinghamurst [17] discuss the possible problems that can be encountered when evaluating [AR](#) applications. They also describe different approaches to evaluating guidelines and conclude that no set of guidelines is appropriate for finding low-level design problems in every case. Therefore, the first step should be to decide which meaningful heuristics and guidelines will be considered. It is also explained that qualitative evaluation approaches are most suited to test user experience.

With this in mind, to help define these heuristics and guidelines, another study that matches this work's scenario was looked into. S. Sylaiou et al. [53] show the results from evaluating *ARCO*, a cultural heritage [AR](#) game. They provide both a qualitative and quantitative analysis of the results and describe which metrics were used for the evaluation.

With the structure of the usability tests already roughly designed, a target number of test subjects should be determined for a credible final result analysis of the developed solution. K. Caine [8] presents research on user studies sample sizes in a multitude of Computer-Human Interaction (CHI) manuscripts, along with offering transparency on local standards within the CHI community. He concludes that the most common reported sample size across all regarded studies was 12, by a considerable margin. This way, providing insight on a feasible, minimum sample size goal for the usability tests performed in this work.

With everything taken into account, a minimum of 12 usability tests should be performed. These tests would rely on the following of undescriptive tasks, heuristic evaluation, and questionnaire answering, inspired by the aforementioned studies. There would also be the possibility of performing behaviour recording and using the "think aloud" method to further document and assess each users' performance and amusement.

5.1.1.1 Questionnaires

After performing each set of undescriptive tasks, the participants are usually required to answer questionnaires in order to evaluate various aspects of what they have just tested.

System Usability Scale (SUS)

J. Brooke [7] proposes the System Usability Scale (SUS) as a tool to evaluate the usability of products or services. This tool consists of a series of 10 affirmations where the participants are prompted to answer whether they agree with them or not, having to choose an option within a 5-Likert-scale measurement, ranging from (1) *Strongly Disagree* to (5) *Strongly Agree*. In order to evaluate the results of a SUS questionnaire, the researchers need to follow a simple process. Firstly, subtract 1 from the odd-numbered affirmations answer value, and then subtract the even-numbered affirmations answer value from 5. Secondly, add the values obtained of all the participant's answers and multiply them by 2.5. The final value will range between 0 and 100 but it does not reflect a percentage of the system's quality. The average value of a SUS questionnaire is 68 points out of 100, so any value below 68 is below average and any value above this is above average.

The 10 original affirmations that compose this questionnaire and were subsequently used are replicated in Annex I.

User Experience Questionnaire (UEQ)

M. Schrepp et al. [48] propose the User Experience Questionnaire (UEQ) as a tool to measure both classical usability aspects, like efficiency, perspicuity, and dependability, and user experience aspects, like attractiveness, stimulation, and novelty. This tool consists of a series of 26 pairs of opposite adjectives, 6 to measure attractiveness, and 4 to measure each of the other mentioned aspects. Participants are prompted to answer within a 7-scale

ranging from one adjective to its correspondent opposite. For example, from *Annoying* to *Enjoyable*. In order to evaluate the results of a UEQ questionnaire, the development team also provides a data analysis tool that automatically divides each participant's 26 entries into their respective aspects and performs some standard statistical calculations. They also offer a benchmark, which is updated every year, in order to compare the obtained results with other product evaluations.

The 26 original pairs of opposite adjectives that compose this questionnaire and were subsequently used are replicated in Annex II.

Custom Questionnaires

In order to further evaluate specific qualitative aspects of the solution, certain questions can be asked relative to each particular tested feature, while taking into consideration the aforementioned significant heuristics. Reiterating, it is important to assess these qualitative aspects that correlate to the test subjects' opinions since the main goal of the solution is to increase their drive to use the application. Much like the SUS questionnaire, when possible, these questions should be shifted to an affirmative statement format, where the participant uses the 5-Likert-scale measurement to answer. This format is believed to make participants feel more confident in their decisions during the testing process while also minimizing response biases and errors. Additionally, this time, contrary to the SUS questionnaire, only positive items will be included. Since including both positive and negative worded items can lead to mistakes and there is no significant difference in response bias between positive and negative items, this makes using only positive affirmations more reliable [47].

5.1.1.2 Procedure

Two distinct usability tests were ultimately designed, one for the main solution, which incorporates all the AR interactions and the missions interface, and another for the developed Unity mission system tool. The two tests followed an identical structure, which is described below:

To carry out each test, the presence of two elements was necessary: the facilitator, who supervised the tests, and the volunteer, who performed the tests. Some test sessions took place in person, while others were done online, with the help of a videoconferencing platform to allow communication between the two parties and for the volunteers to be able to share their screen with the facilitator.

Each session starts by explaining or having the volunteer read an explanation of the test's context along with a brief description of what he was going to be put through and a consent form, which can be seen in Appendix A. After consenting to participate in the test, the volunteer is then presented with a demographic profile fill-out form, which collects information for statistical purposes only.

Subsequently, the volunteer then goes through a series of groups, which are composed of a set of unresponsive tasks followed by a set of quantitative and qualitative questions, as described before. A few of those groups also include some information that might help clarify any volunteers' doubts without compromising their valuable, unbiased experience and answers.

Finally, the volunteer is presented with the standard and custom questionnaires previously described in Subsection 5.1.1.1. Regarding the standard ones, the main solution test exhibits both the SUS and UEQ questionnaires while the mission tool test only exhibits the SUS questionnaire.

5.2 Main Usability Tests

These main usability tests aimed to evaluate the final implemented state of the system, which was described in detail over the course of Chapter 4. This includes all the [AR](#) interactions and the missions interface.

The form used to guide the flow of the tests is divided into various sections, each relative to a different interaction to be evaluated. Each section, except three specific ones, is subsequently constituted by its respective description, a set of unresponsive tasks to be performed by the volunteer, a questionnaire that allows for the gathering and storage of all the necessary data for a final analysis of the test results, and finally, two text boxes where the volunteer can record its opinion about positive aspects and aspects that need improvement regarding the tested section. These latter mentioned text boxes were also sometimes utilized by the facilitator in order to document relevant user behavior in some cases for later analysis. The aforementioned three sections that stray from the format described are the ones that contain the custom and standard questionnaires laid out in Subsection 5.1.1.1. The last two of these three sections correspond to the SUS questionnaire and the UEQ respectively.

For these tests, a demonstration version of the system was made that allowed users to launch any interaction from a main menu. This made it so the test subjects would not have to follow any itineraries and move between locations during the tests, since this was not relevant for this work's evaluation and was also already previously tested by one of the authors of the base platform. Furthermore, interactions that would normally use contents prompted by users in the narrative creation interface had them withdrawn from a predefined set of contents just for the purpose of the tests.

Lastly, in these tests, great success can be considered achieved if the volunteers, after finishing all the tasks and answering the questionnaires, unbiasedly ask if they can experience going through more interactions. The correctness of difficulty adaptability based on user profiling calculations is also evaluated using these tests, since an unbiased player is needed to weigh the difficulty level of both the [AR](#) interactions and missions' tasks.

5.2.1 Participants Demographic Profile

A total of 17 volunteers participated in this study, meeting the aforementioned minimum requirement of 12 participants. As seen in (fig. 5.1a), these volunteers ranged in age from 18 to 39 years old, with an average of 23,76 and a standard deviation of 4,09. Out of the 17 participants, 13 were male and 4 were female.

As seen in (fig. 5.1b), the majority of the volunteers did not have high experience with AR, with only one choosing an option higher than average. Additionally, in (figs. 5.1c and 5.1d), it can be seen that the majority of volunteers were avid gamers, playing games weekly or every day, and that the most preferred game modes were single-player and player-versus-player.

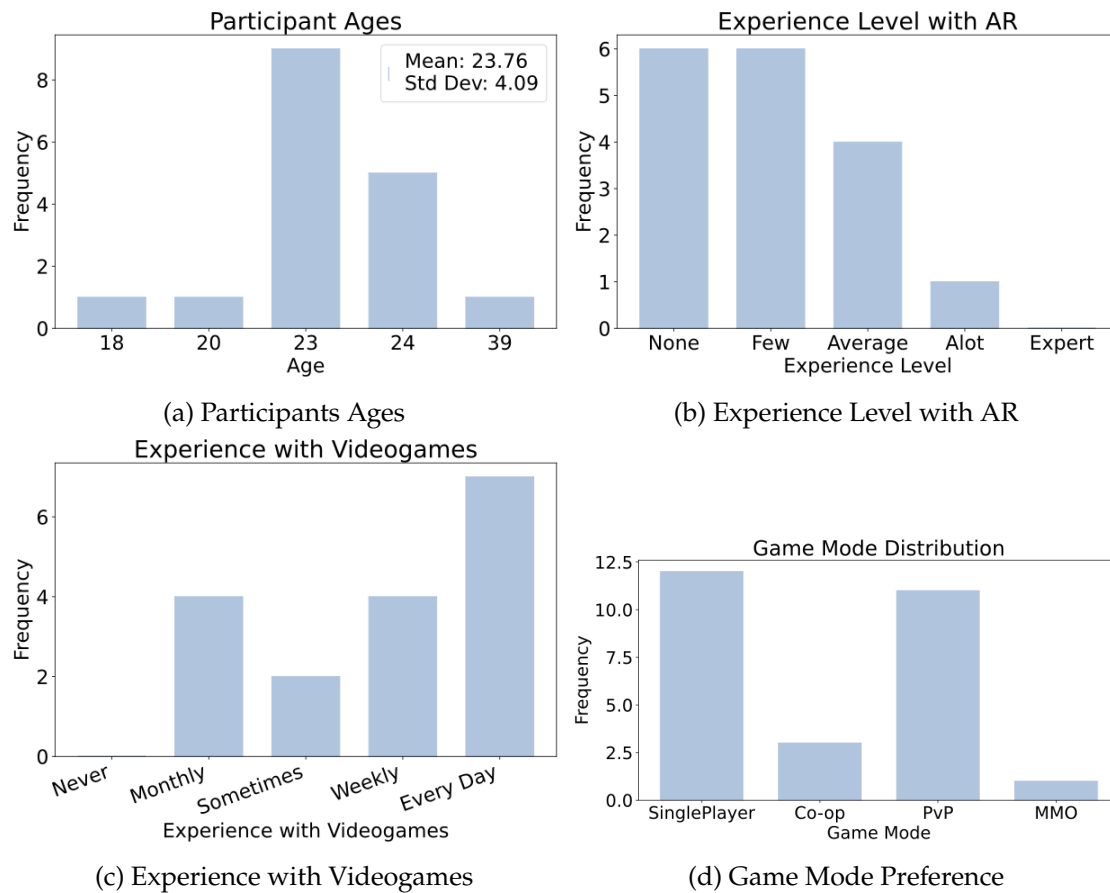


Figure 5.1: Participants' Demographic Profile Graphs - Main

5.2.2 Tests and Results Analysis

The following subsections will go over each section of the aforementioned form, following its structure. In each one, a description of the tasks the volunteers were inquired to perform and of the respective questionnaires they had to answer is given. Following that, the results are presented along with their analysis and observations made by both the volunteers and the facilitator. Finally, the results are also discussed.

The questionnaires presented in the first four sections have a very similar structure. The first affirmation is regarding whether the volunteer was able to complete the puzzle and has three answer options. Either the volunteer was able to successfully perform the tasks without help, or it was able to do them with help, or it was not able to complete them. The following affirmations are relative to how the volunteer perceived certain aspects of the interaction and are answered on a 5-point Likert scale, with the exception of one question, which required inserting how many times the volunteer performed the interaction as a means to better understand its answer in the affirmation that immediately follows it.

5.2.2.1 Section S1 - AR Jigsaw Puzzle

In this section, it was intended that volunteers would go through the AR Jigsaw Puzzle interaction. In order to guarantee that the volunteers would experience all the possible events during this interaction, they had as an objective to perform the following tasks: Consult the help information; Position the mini-game on the floor in front of you; Select a piece and try to position it incorrectly; Complete the puzzle in question by positioning all the pieces correctly; Collect the reward that appears after completing the puzzle.

After the volunteers performed all the tasks, they then were required to answer the following questionnaire:

- **S1Q1** - I was able to complete the puzzle.
- **S1Q2** - The navigation through the piece list was intuitive.
- **S1Q3** - It was clear how to select a piece and position it on the board.
- **S1Q4** - I was able to understand when I was trying to position a piece incorrectly.
- **S1Q5** - The increment in time when I tried to position a piece incorrectly was noticeable.
- **S1Q6** - How many times did you perform this interaction?
- **S1Q7.1** - After performing this interaction an indeterminate number of times, I found that its difficulty adapted well to my profile.
- **S1Q7.2** - I only performed the interaction once and found the initial difficulty appropriate.

- **S1Q8** - I found this interaction attractive.

According to the results seen in Table 5.1, all the volunteers were able to complete the puzzle (S1Q1) and, therefore, the interaction itself, with only one volunteer needing help from the facilitator. Overall, the basic engagement with this interaction (S1Q2, S1Q3) showed positive results. The volunteers were able to choose pieces from the list and place them in a desired position on the board. The first has a median of the highest value. However, the latter shows a lower result with an average below four. Additionally, the feedback the volunteer receives when trying to position a piece incorrectly on the board (S1Q4, S1Q5) also showed positive results, both with a median of the highest value. Once again, the latter shows a little more variance in its answers, with a lower average as well. Although the starting difficulty (S1Q7.2) was considered by volunteers to be appropriate, its scaling (S1Q7.1) presented the lowest results, having the lowest median and an average lower than three. Finally, the final affirmation regarding the overall satisfaction of the volunteer and the attractiveness of the interaction (S1Q8) showed a great result, with a median of the highest value and an average superior to four and a half.

Table 5.1: Section S1 Questionnaire Results

S1Q1	Participants(#)			S1Q6	Participants(#)	
Yes	16			Once	8	
Yes, with help	1			Twice	2	
No	0			Three times	3	
				Four times	4	
Quest.	Mean	Std.	Med.	1st Q.	3rd Q.	
S1Q2	4.82	0.38	5	-0	+0	
S1Q3	3.94	0.87	4	-1	+1	
S1Q4	4.94	0.23	5	-0	+0	
S1Q5	4.17	0.98	5	-2	+0	
S1Q7.1	2.67	1.05	3	-1	+0	
S1Q7.2	4.12	0.93	4	-0	+1	
S1Q8	4.59	0.49	5	-1	+0	

Observations

During this interaction, it was observed that some volunteers did not read the instructions provided in the information foldout, preferring to try and discover how to proceed by themselves. This led to some also trying to drag pieces from the list instead of clicking on them.

In relation to positive aspects, the volunteers wrote observations on how the images were of good quality and that the whole AR interaction felt like a game. On the other hand, in relation to aspects to improve, volunteers wrote how the opacity of the board would be better if increased, that there should additionally be a dragging functionality, and that the time increment was only somewhat noticeable and that it should include the time unit representation. Finally, there were volunteers who suggested that both the starting and subsequent difficulties could be higher.

Discussion

The mentioned lower results in question (S1Q3) are considered to come from the volunteers trying to drag the pieces instead of clicking on them. This behavior is due to them not reading the instructions. However, since for some of them, dragging was the first instinct, it does make sense that this should be an included feature.

Concerning the difficulty (S1Q7), volunteers made it clear that, for this type of puzzle, the difficulty could be higher, meaning that they preferred a puzzle divided into more pieces. The difficulty system is based on the time the player takes to complete the puzzles and reduces the difficulty if the player is very slow. In this case the difficulty was reduced, but volunteers felt like they did not mind to spend more time constructing a larger puzzle. This is both a negative and positive aspect, since it shows that the difficulty system is not well calibrated, but at the same time, that volunteers found the interaction enjoyable. This argument is reinforced in the last affirmation (S1Q8), which shows that the volunteers did in fact find the interaction attractive. In conclusion, the difficulty should be increased.

5.2.2.2 Section S2 - AR Sliding Puzzle

In this section, it was intended that volunteers would go through the AR Sliding Puzzle interaction. In order to guarantee that the volunteers would experience all the possible events during this interaction, they had as an objective to perform the following tasks: Consult the help information; Position the mini-game on the floor in front of you; Complete the puzzle in question by moving all the pieces to their correct position; Collect the reward that appears after completing the puzzle.

After the volunteers performed all the tasks, they then were required to answer the following questionnaire:

- **S2Q1** - I was able to complete the puzzle.
- **S2Q2** - How many times did you perform this interaction?
- **S2Q3.1** - After performing this interaction an indeterminate number of times, I found that its difficulty adapted well to my profile.
- **S2Q3.2** - I only performed the interaction once and found the initial difficulty appropriate.
- **S2Q4** - I found this interaction attractive.

According to the results seen in Table 5.2, all the volunteers were able to complete the puzzle (S2Q1) and, therefore, the interaction itself. This time, two volunteers needed assistance from the facilitator. Since this interaction's only engagement with the puzzle revolved around clicking on it, there was no specific question regarding basic engagement or feedback. Similarly to the previous puzzle, the volunteers gave the starting difficulty (S2Q3.2) a better score than its adaptability (S2Q3.1). The first has a median of four, and

the second has a median of three. The overall results for the difficulty of this puzzle, although still positive, were not great. Finally, the final affirmation regarding the overall satisfaction of the volunteer and the attractiveness of the interaction (S2Q4) showed a positive result. With a median of four, volunteers found this puzzle attractive, but not as much as the previous one. This can also be seen in question (S2Q2), where it shows that the volunteers went through the interaction fewer times overall.

Table 5.2: Section S2 Questionnaire Results

S2Q1	Participants(#)			S2Q2	Participants(#)	
Yes	15			Once	7	
Yes, with help	2			Twice	5	
No	0			Three times	5	
				Four times	0	
Quest.	Mean	Std.	Med.	1st Q.	3rd Q.	
S2Q3.1	2.90	0.70	3	-1	+0	
S2Q3.2	3.85	1.24	4	-0	+1	
S2Q4	4.11	0.83	4	-0	+1	

Observations

During this interaction, the same volunteers who did not consult the instructions continued to do so. It was also observed that one of the volunteers did not recognize this type of puzzle from the get-go, not understanding exactly how to solve it. However, the volunteer did figure it out quickly after reading the instructions. It was also observed that in this puzzle, although the pieces are moved by clicking on them, some volunteers were doing a sliding motion.

In relation to positive aspects, the volunteers wrote observations on how this type of puzzle requires more brain usage to solve, serving as a good mental exercise. However, other volunteers wrote this exact observation as an aspect to improve, not enjoying the inherent difficulty of the puzzle, and some even finding it annoying.

Discussion

Although this interaction showed a slightly lower result regarding its difficulty, this is believed to be related to the inherent difficulty of the puzzle type itself. Therefore, it might be better if this puzzle maintained its starting size for a wider range of times while still allowing interested players to be capable of achieving higher levels. Additionally, something should be included to increase this puzzle's appeal.

The second observation made, regarding the volunteer who did not recognize this puzzle type, shows the usefulness of the information and instructions foldout. Additionally, regarding the volunteers who were doing a sliding motion instead of just clicking, which was the intended action, this did not impose any problems. Due to this type of puzzle only allowing pieces adjacent to the empty space to be moved into its position, this means the pieces are only able to move in one direction at any given time. Therefore, there is no relevant or noticeable difference between sliding a piece or clicking a piece.

5.2.2.3 Section S3 - AR Balloon Pop

In this section, it was intended that volunteers would go through the AR Balloon Pop interaction. In order to guarantee that the volunteers would experience all the possible events during this interaction, they had as an objective to perform the following tasks: Consult the help information; Position the interaction on the floor in front of you; Complete the mini game by popping the target number of balloons; Collect the reward that appears after completing the mini game.

After the volunteers performed all the tasks, they then were required to answer the following questionnaire:

- **S3Q1** - I was able to complete the mini-game.
- **S3Q2** - I found the dart throwing action well configured.
- **S3Q3** - I found the different properties of each balloon to be evident.
- **S3Q4** - I was able to have a good perception of the depth/distance at which each balloon is located.
- **S3Q5** - I found the number of balloons to be popped and the current pop count explicit.
- **S3Q6** - How many times did you perform this interaction?
- **S3Q7.1** - After performing this interaction an indeterminate number of times, I found that its difficulty adapted well to my profile.
- **S3Q7.2** - I only performed the interaction once and found the initial difficulty appropriate.
- **S3Q8** - I found this interaction attractive.

According to the results seen in Table 5.3, all the volunteers were able to complete the mini-game (S3Q1) and, therefore, the interaction itself. This time, contrary to both the puzzles, none of the volunteers needed the facilitator's intervention. Overall, this interaction showed great results. The affirmations relative to the basic engagement with the interaction (S3Q2, S3Q3, S3Q5) all have a median of the highest value. The volunteers were able to throw darts and pop balloons in a satisfactory manner while understanding their properties and at what relative distance they were located. Additionally, volunteers found the mini-game's goal quite explicit (S3Q5), also having a median of the highest value. Once again, similarly to the two previous interactions the volunteers gave the starting difficulty (S3Q7.2) a better score than its adaptability (S3Q7.1). The first has a perfect score, with all volunteers agreeing that the starting difficulty was appropriate. The latter is the only question to not score the highest value for the median. Finally, the final affirmation regarding the overall satisfaction of the volunteer and the attractiveness

of the interaction (S3Q8) showed the best possible result, with all volunteers completely agreeing with the affirmation.

Table 5.3: Section S3 Questionnaire Results

S3Q1	Participants(#)	S3Q6	Participants(#)		
Yes	17	Once	5		
Yes, with help	0	Twice	5		
No	0	Three times	5		
		Four times	2		
Quest.	Mean	Std.	Med.	1st Q.	3rd Q.
S3Q2	4.94	0.23	5	-0	+0
S3Q3	4.29	0.89	5	-1	+0
S3Q4	4.35	1.02	5	-1	+0
S3Q5	4.47	1.03	5	-0.5	+0
S3Q7.1	4.42	0.49	4	-0	+1
S3Q7.2	5.00	0.00	5	-0	+0
S3Q8	5.00	0.00	5	-0	+0

Observations

No major observations were made by the facilitator during this interaction, besides the fact that, taking into consideration the volunteers' reactions, this was by far the most engaging interaction. And that the same volunteers who did not consult the instructions continued to do so.

In relation to positive aspects, the volunteers wrote observations on how they all around really enjoyed this interaction. Additionally, one volunteer wrote that the trajectory of the dart was well implemented and that it resembles reality. On the other hand, in relation to aspects to improve, volunteers wrote how they would prefer to have a higher pop goal on subsequent playthroughs. Moreover, one volunteer wrote that the pop count's position in the screen was not the best and there should be more clarity regarding the objective of the mini-game.

Discussion

Even though the question regarding the difficulty adaptability (S3Q7.1) had a great score, because of the amount of observations on how there should be a higher goal, it can be assumed that the high result was due to a bias since all the volunteers completely agreed on the interaction being attractive. Therefore, the difficulty of further levels should be increased, and the progression should be made easier by increasing the allowed time to go up a level.

Although this mini-game, in this form, does not contain many elements related to the work's context, it showed amazing results regarding the satisfaction and engagement of volunteers (S3Q8), solidifying the decision taken during the design phase.

5.2.2.4 Section S4 - AR Quiz

In this section, it was intended that volunteers would go through the AR Quiz interaction. In order to guarantee that the volunteers would experience all the possible events during this interaction, they had as an objective to perform the following tasks: 1.Consult the help information; 2.Observe the question and the different answer options; 3.Select an answer option; 4.Collect the reward if you answered correctly or Observe the correct answer if you answered incorrectly; 5.Reopen the interaction and repeat tasks 2 to 4. This last task is for the volunteer to be able to both see what happens when it wins or loses. Additionally the facilitator intervened in task 3 to let the volunteer know which answer option to choose.

After the volunteers performed all the tasks, they then were required to answer the following questionnaire:

- **S4Q1** - I managed to find all the answer options.
- **S4Q2** - I found the question and all the answer options readable.
- **S4Q3** - I was able to select the answer options announced by the facilitator.
- **S4Q4** - I was able to understand when I chose a incorrect answer option.
- **S4Q5** - I was able to understand when I chose the correct answer option.
- **S4Q6**- After answering the quiz incorrectly, I managed to figure out what the correct answer was.
- **S4Q7** - I found this interaction attractive.

According to the results seen in Table 5.4, all the volunteers were able to understand they had to look around in order to find all the answer options (S4Q1), with three of the volunteers needing a slight help from the facilitator. Additionally, all the volunteers were able to select the answer options announced by the facilitator during task number three (S4Q3), besides only one, which required help from the facilitator. Overall, the three-dimensional text readability (S4Q2) and the quiz's feedback on the volunteers' responses (S4Q4, S4Q5, S4Q6) showed positive results. The last mentioned question (S4Q6), although having positive results, with an average above three, stood out for having the worst answers from the volunteers. Finally, the final affirmation regarding the overall satisfaction of the volunteer and the attractiveness of the interaction (S4Q7) showed the worst results so far, with only a median of three.

Observations

During this interaction, it was observed that some volunteers failed to, at first, understand what was the correct answer option after failing the quiz.

In relation to positive aspects, the volunteers wrote observations on how the text is easily readable and the movement of the answer options around the question is fluid. On the other hand, in relation to aspects to improve, volunteers wrote that although

Table 5.4: Section S4 Questionnaire Results

S4Q1	Participants(#)			S4Q3	Participants(#)	
Yes	14			Yes	16	
Yes, with help	3			Yes, with help	1	
No	0			No	0	
Quest.	Mean	Std.	Med.	1st Q.	3rd Q.	
S4Q2	4.35	0.90	5	-1	+0	
S4Q4	4.58	0.69	5	-1	+0	
S4Q5	4.23	0.87	5	-2	+0	
S4Q6	3.47	1.14	3	-0.5	+2	
S4Q7	3.35	0.83	3	-0	+1	

they could understand when they chose a correct or incorrect answer option, they also considered the feedback animation after choosing a answer option to be confusing.

Discussion

Even though there were some volunteers who were able to easily go through the interaction without problems and understood all its gimmicks and animations, there were others who were not able to do so. This was noticed specifically after failing a quiz, as mentioned in the first observation. Some of the somewhat confused volunteers still managed to understand whether they chose the correct or incorrect answer option, mainly due to the sound effects and complementary animations besides the movement one. Therefore, taking inspiration from a suggestion made by one of the volunteers, it would be more enjoyable for some players to have static answer options instead of orbiting ones. This could be implemented either by including it as an option or as a difficulty scaling feature.

5.2.2.5 Section S5 - AR Geospatial Flyers

In this section, it was intended that volunteers would go through the AR Geospatial Flyers interaction. In order to guarantee that the volunteers would experience all the possible events during this interaction, they had as an objective to perform the following tasks: 1.Move to the location of a flyer; 2.Get closer and open the flyer; 3.Observe the contents of the flyer including the quiz presented; 4.Select an answer option; 5.Observe whether or not the selected answer option is correct; 6.Repeat tasks 1 to 5. This last task is for the volunteer to be able to see what happens when it gets the answer right or wrong. Additionally the facilitator intervened in task 4 to let the volunteer know which answer option to choose.

After the volunteers performed all the tasks, they then were required to answer the following questionnaire:

- **S5Q1** - I was able to understand which direction to walk to get to each flyer.
- **S5Q2** - I was able to understand how to interact with the flyers.
- **S5Q3** - I found the content on the cover and inside of the flyers to be readable.

- **S5Q4** - I was able to select the answer options announced by the facilitator.
- **S5Q5** - I was able to understand when I chose a incorrect answer option.
- **S5Q6** - I was able to understand when I chose the correct answer option.
- **S5Q7** - After answering the quiz incorrectly, I managed to figure out what the correct answer was.
- **S5Q8** - I found this interaction attractive.

According to the results seen in Table 5.5, all the volunteers were able to understand how to open a flyer (S5Q2), when they got to it, without any help from the facilitator. Additionally, all the volunteers were also able to select the answer options announced by the facilitator during task number four (S5Q4). Following the included compass in order to get to a flyer's location (S5Q1) was intuitive for the volunteers, with the results having a median of the highest value and an average of above four and a half. The readability of both the cover and inside contents of the flyers (S5Q3) had even greater results with the same characteristics. In contrast to the previous quiz interaction, this quiz's feedback on the volunteers' responses (S5Q5, S5Q6, S5Q7) had a perfect score, with all the volunteers perfectly understanding whether they chose a correct or wrong answer, and in the case of choosing a wrong answer, understanding which was the correct one. Finally, the final affirmation regarding the overall satisfaction of the volunteer and the attractiveness of the interaction (S5Q8) showed a great result, with a very similar score to the jigsaw puzzle.

Table 5.5: Section S5 Questionnaire Results

S5Q2	Participants(#)			S5Q4	Participants(#)		
Yes	17			Yes	17		
Yes, with help	0			Yes, with help	0		
No	0			No	0		
Quest.	Mean	Std.	Med.	1st Q.	3rd Q.		
S5Q1	4.64	0.96	5	-0	+0		
S5Q3	4.82	0.38	5	-0	+0		
S5Q5	5.00	0.00	5	-0	+0		
S5Q6	5.00	0.00	5	-0	+0		
S5Q7	5.00	0.00	5	-0	+0		
S5Q8	4.52	0.49	5	-1	+0		

Observations

During this interaction, neither the facilitator nor the volunteers made any major observations. One volunteer pointed out that the feedback on the chosen answer options was in fact much better achieved in this interaction when compared to the previous quiz.

Discussion

The fact that the volunteers could better understand the feedback given by this quiz when compared to the previous one is considered to be due to the animation being more simple

and straightforward, with no over-the-top movement animations that the players have to follow since it is static.

5.2.2.6 Section S6 - AR Interactions Questionnaire

This section presents questions that are common to the [AR](#) interactions tested in the previous sections. The volunteers were therefore required to answer the following questionnaire:

- **S6Q1** - I was able to consult the help and information foldout.
- **S6Q2** - I found the help and information displayed useful.
- **S6Q3** - I was able to position the augmented reality interactions where I wanted.
- **S6Q4** - The method used to position augmented reality interactions was intuitive.
- **S6Q5** - Interactions were fluid.
- **S6Q6** - Interactions were responsive to my actions.
- **S6Q7** - I found the interactions engaging.
- **S6Q8** - The generated elements were well integrated with the real space.
- **S6Q9** - I felt like I had control over the interactions.
- **S6Q10** - I was able to focus on the problem to be solved in the interactions rather than on how to interact with them.
- **S6Q11** - The sound effects used were appropriate for their respective events.
- **S6Q12** - Would you return to the application to experience any of the interactions again?

According to the results seen in Table [5.6](#), all the volunteers were able to find and consult the instructions and information foldout without help (S6Q1). Following that, they also found the information present within it to be useful (S6Q2). On the contrary, although all volunteers were able to position the interaction where they desired (S6Q3), eight of them required the help of the facilitator. However, after quickly understanding this procedure, the volunteers found it quite intuitive (S6Q4), with its results still having a median of four. Overall, all seven following affirmations, which are relative to the volunteers opinions on certain heuristics (S6Q5 to S6Q11), showed great results, all having a median of the highest value, except one. This mentioned affirmation (S6Q7) only has a median value of four. Another of these worth highlighting has the best score, with all volunteers completely agreeing on being in control during the interactions (S6Q9). Finally, the results for the last question (S6Q12) can be seen in (fig. [5.2](#)). All the volunteers would

like to reexperience the balloon pop mini-game, while roughly half also enjoyed the jigsaw and geospatial flyers interactions. The sliding puzzle and especially the quiz were not as preferred.

Table 5.6: Section S6 Questionnaire Results

S6Q1	Participants(#)						S6Q3	Participants(#)
Yes	17						Yes	9
Yes, with help	0						Yes, with help	8
No	0						No	0
Quest.	Mean	Std.	Med.	1st Q.	3rd Q.			
S6Q2	4.65	0.59	5	-1	+0			
S6Q4	4.06	0.80	4	-1	+1			
S6Q5	4.82	0.38	5	-0	+0			
S6Q6	4.88	0.32	5	-0	+0			
S6Q7	4.17	0.78	4	-0.5	+1			
S6Q8	4.65	0.48	5	-1	+0			
S6Q9	5.00	0.00	5	-0	+0			
S6Q10	4.65	0.48	5	-1	+0			
S6Q11	4.88	0.32	5	-0	+0			

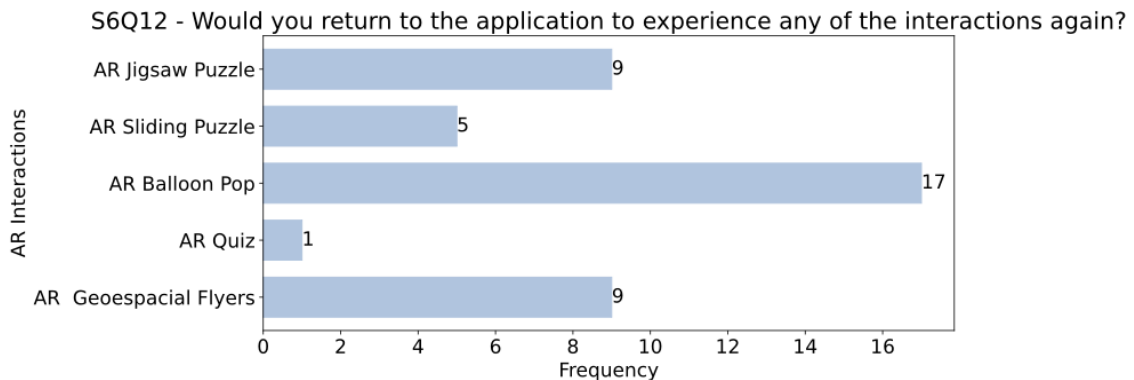


Figure 5.2: AR Interactions Preference

5.2.2.7 Section S7 - Missions Interface

In this section, it was intended that volunteers would go through the Missions System Interface. In order to guarantee that the volunteers would experience all the possible events during this interaction, a custom save file with some already finished tasks and missions was used. Furthermore, the volunteers had as an objective to perform the following tasks: Observe your progress in the daily mission, general missions and achievements; Collect rewards from completed missions.

After the volunteers performed all the tasks, they then were required to answer the following questionnaire:

- **S7Q1** - I found the popup that informs of the completion of a task during the interactions in the previous sections to be noticeable.
- **S7Q2** - Navigating through the different tabs was intuitive.
- **S7Q3** - I found the interface contents to be readable.
- **S7Q4** - I was able to distinguish between completed and uncompleted tasks and missions.
- **S7Q5** - I was able to identify my progress on unfinished tasks.
- **S7Q6** - I was able to collect rewards from completed missions and progress through the questlines.

According to the results seen in Table 5.7, during the interactions, the majority of volunteers did notice the popup that informs of a task's completion (S7Q1). However, this affirmation's results show a discernible discrepancy in answers. There were volunteers who did not notice this popup at all. All the volunteers completely agreed that the navigation through the different tabs was intuitive (S7Q2), with the results showing a perfect score. Moreover, volunteers found the interface to be very readable (S7Q3), with only one user not completely agreeing. The following affirmation shows that, yet again, all but one completely agree with the completed and uncompleted tasks and missions being easily distinguishable (S7Q4). Again, with the best score, all the volunteers found the progress on unfinished tasks to be very perceptible (S7Q5). Finally, the majority of volunteers were able to progress by collecting the completed missions' rewards (S7Q6), with a small discrepancy but still a median of the highest value.

Table 5.7: Section S7 Questionnaire Results

Quest.	Mean	Std.	Med.	1st Q.	3rd Q.
S7Q1	4.06	1.43	5	-2	+0
S7Q2	5.00	0.00	5	-0	+0
S7Q3	4.94	0.23	5	-0	+0
S7Q4	4.94	0.23	5	-0	+0
S7Q5	5.00	0.00	5	-0	+0
S7Q6	4.41	1.09	5	-1	+0

Observations

During this section, it was observed that two of the volunteers were trying to collect tasks instead of missions. Furthermore, some volunteers also tried collecting rewards by clicking on their icon instead of the designated button.

In relation to positive aspects, more than one volunteer wrote an observation on how they recognized this interface's layout from another game and that it was intuitive to use. On the other hand, in relation to aspects to improve, volunteers wrote that the navigation

through the tabs when swiping was cumbersome, having to swipe a long distance before the pages switched.

Discussion

Relatively to the two first observations made, all the correspondent volunteers had less than average experience with videogames. Additionally, they also somewhat quickly understood that they were doing something wrong and eventually figured out how to do it without any help from the facilitator. Relatively to the cumbersome swiping, it is possible through a script to increase this specific scroll's sensitivity in order for the player to not have to swipe such a long distance to switch pages.

5.2.2.8 System Usability Scale (SUS)

After finishing all the previous sections, the volunteers were required to answer a post-session SUS questionnaire. The volunteers were told to answer while taking into consideration everything they had tested until then, including all the [AR](#) interactions and the missions interface.

The questionnaire's results for each affirmation can be seen in Table 5.8. In odd affirmations, which are highlighted in blue, a higher value is better, and, on the contrary, in even affirmations, which are highlighted in red, a lower value is better. An affirmation worth emphasizing is the number seven, where, although the answers were still positive, it was the lowest scored. This indicates that the volunteers thought the system was not that easy for most people to learn quickly.

Additionally in (fig. 5.3), each volunteer's independent calculated scores can be better visualized. The average of all scores was **86.76**, which is higher than the SUS average of 68, indicating an overall positive result.

Table 5.8: Main SUS Questionnaire Answers - Shortened Affirmations

<i>SUS Affirmations</i>	<i>Mean</i>	<i>Std.</i>	<i>Med.</i>	<i>1st Q.</i>	<i>3rd Q.</i>
1. I would use the system frequently.	4.06	0.63	4	-0	+0.5
2. The System is complex to use.	1.41	0.84	1	-0	+0.5
3. The System is easy to use.	4.52	0.50	5	-1	+0
4. I would need technical support.	1.06	0.23	1	-0	+0
5. The system has well integrated functions.	4.65	0.48	5	-1	+0
6. The system is inconsistent.	1.18	0.32	1	-0	+0
7. The system is easy to learn.	3.88	0.67	4	-1	+0
8. The system was cumbersome to use.	1.23	0.42	1	-0	+0.5
9. I felt confident using the system.	4.23	0.64	4	-0	+1
10. I needed to learn a lot beforehand.	1.82	0.70	2	-1	+0

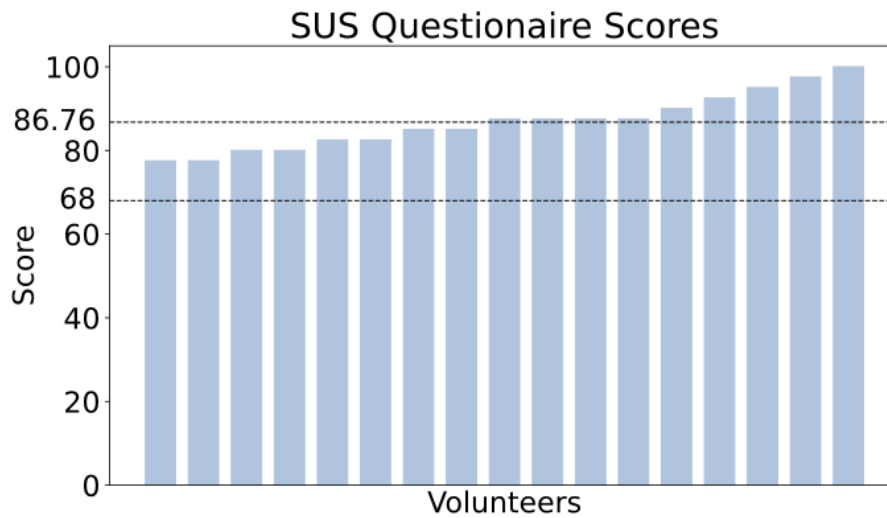


Figure 5.3: SUS Questionnaire Scores

5.2.2.9 User Experience Questionnaire (UEQ)

After filling out the SUS questionnaire, the volunteers were then required to answer the UEQ as the last post-session questionnaire of the usability test.

Although each point of the questionnaire gets answered on a 7-point scale from 1 to 7, the tool provided by the UEQ team presents the results in a range from -3 to 3. The results from each pair of adjectives can be better visualized in Appendix B. Additionally, the results from the combined 26 pairs of adjectives, which form the 3 classical usability aspects, like efficiency, perspicuity, and dependability, and the 3 user experience aspects, like attractiveness, stimulation, and novelty, can be seen in Table 5.9. It is considered by the UEQ team that values above +0.8 represent a positive evaluation. Therefore, all the results presented for each parameter are positive.

Additionally, the measured scale means from Table 5.9 are plotted relative to existing values from a benchmark dataset also provided by the UEQ team. This can be visualized in (fig. 5.4), where it is possible to conclude that the tests showed an overall highly positive result. With the attractiveness, efficiency, dependability, stimulation, and novelty being in the excellent range, which corresponds to the top 10% of the best results, and the perspicuity being in the good range, which corresponds to being above 75% of the results.

Table 5.9: User Experience Questionnaire Scales Results

<i>UEQ Scales</i>	<i>Mean</i>	<i>Var.</i>	<i>Std.</i>
Attractiveness	2.392	0.22	0,471
Perspicuity	1.971	0.47	0,684
Efficiency	2.309	0.18	0,429
Dependability	1.794	0.18	0,426
Stimulation	1.941	0.50	0,705
Novelty	1.721	0.52	0,723

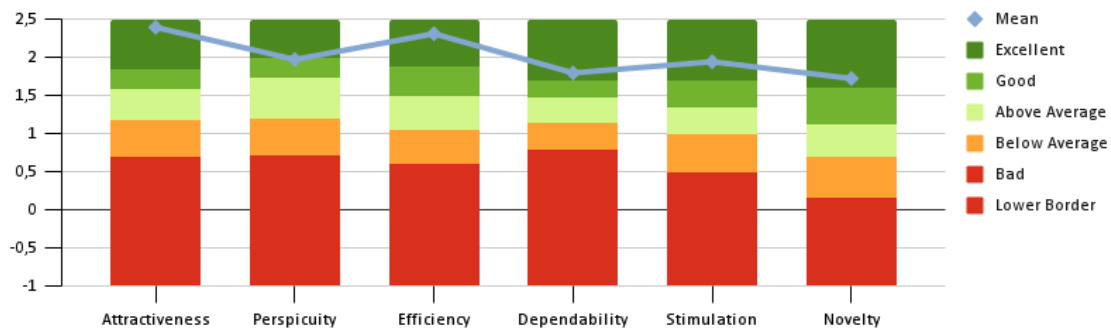


Figure 5.4: User Experience Questionnaire Benchmark

5.3 Tool Usability Tests

The tool usability tests aimed to evaluate the final developed version of the Mission Graph Creation Tool while also considering the implementation of its corresponding interpreter and interface.

The form used to guide the flow of the tests is divided into various sections, each relative to a different element that composes a final quesline. Each section, except the last one, is subsequently constituted by its respective description, a set of undescriptive tasks to be performed by the volunteer, a questionnaire relative to the time the volunteer took to perform the section and whether it was able to complete it, and finally, two text boxes where the volunteer can record its opinion about positive aspects and aspects that need improvement regarding the tested section. The aforementioned section that strays from the format described is relative to the SUS questionnaire.

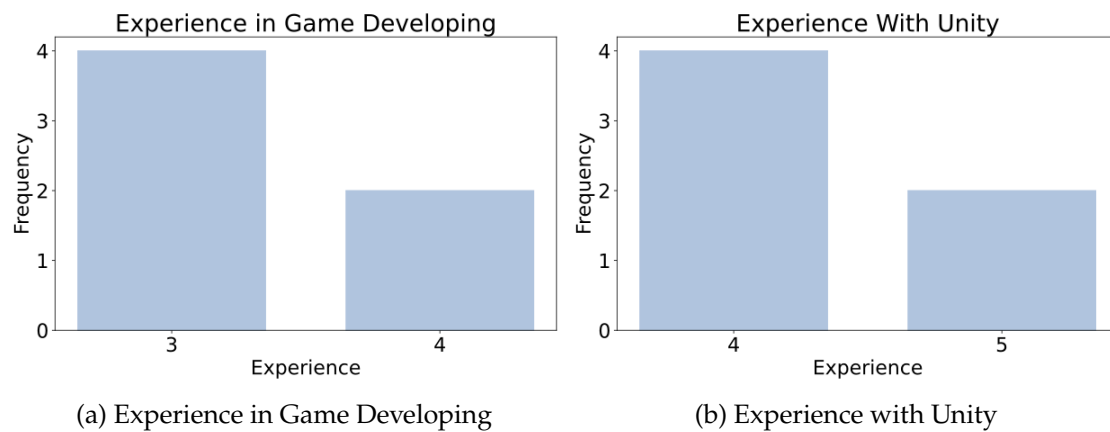
5.3.1 Participants Demographic Profile

A total of only 6 volunteers participated in these tests. Although this does not meet the aforementioned minimum requirement of 12 participants, the results will still be presented here. The results can still be considered usefull as J. Nielsen [35] shows a study on how only five participants normally suffice to find 85% of usability problems.

As seen in Table 5.10b, these volunteers were between the ages of 23 and 24 years old. All of the 6 participants were male, as shown in Table 5.10b. Additionally in (figs. 5.5a and 5.5b), it can be seen that the volunteers considered themselves above average regarding experience in game developing and experience with the Unity game engine, respectively.

Table 5.10: Participants' Demographic Profile Tables - Tool

(a) Participants gender		(b) Participants age	
Gender	Participants(#)	Age	Participants(#)
Male	6	23	4
Female	0	24	2



5.3.2 Tests and Results Analysis

The following subsections will go over each section of the aforementioned form, following its structure. The overall tests were aimed at users who have a minimum level of experience with the Unity game engine or game development in general, which was the case. Although the tool was developed with the intention of being easily used by everyone, the most pertinent questions regarding its usage, relevance, and versatility are believed to be only be adequately answered by users who possess the aforementioned experience. During the test the volunteer was expected to use the tool to create a mission graph with each mission having its respective tasks and rewards.

The volunteers were given for each element a description of the intended final result when creating them, this way eliminating any unwanted variables when comparing and analysing all the tests results.

Only the results are presented and they are not discussed mainly due to the small sample size not giving sufficient insight.

5.3.2.1 Section S1 - Creating a Simple Task

In this section, it was intended that volunteers created a task that did not involve much thinking regarding its parameters.

Task to be created: name: puzzle task; description: This task counts the puzzles completed by the player and has the text displayed in the interface "Complete X puzzles" with X being the number of puzzles to be completed to finish the task. The value of X is intended to be between 2 and 10.

The volunteers had as an objective to perform the following tasks: Open the task management window; Open the task creation window; Write the name and text to be presented; Configure the necessary variables; Save the created task.

All the volunteers were able to finish this section, 5 without the facilitator's help and

1 with help. The volunteers that did not require help took between 27 seconds and 1 minute and 54 seconds to complete the tasks while the volunteer that required help took 3 minutes and 24 seconds.

5.3.2.2 Section S2 - Creating a More Complex Task

In this section, it was intended that volunteers created a task whose parameters were not as obvious as the previous section's task.

Task to be created: name: timed puzzle task; description: This task is completed when the player completes a puzzle above a certain difficulty level within a set time. The text presented in the interface is "Finish a level X or higher puzzle in under Y seconds" with X being the minimum level of the puzzle and Y being the maximum time in seconds. The value of X is intended to be 3 and the value of Y is 15.

The volunteers had as an objective to perform the following tasks: Open the task management window; Open the task creation window; Write the name and text to be presented; Configure the necessary variables; Save the created task.

All the volunteers were able to finish this section, this time, in contrast with the previous section, 1 without the facilitator's help and 5 with help. The volunteer that did not require help completed the tasks in 54 seconds while the others completed them between 2 minutes and 9 seconds and 3 minutes and 18 seconds.

5.3.2.3 Section S3 - Creating a Reward

In this section, it was intended that volunteers created a reward.

Reward to be created: name: reward sim; description: It is intended that this reward, once collected by the player, calls the "MockRewardCollect" function of the prefab that contains the mission system script.

The volunteers had as an objective to perform the following tasks: Open the rewards management window; Open the reward creation window; Choose the name and image to be displayed; Configure the function to be called in the reward collection event; Save the created reward.

All the volunteers were able to finish this section without help. The volunteers took between 28 seconds and 1 minute to complete the tasks.

5.3.2.4 Section S3 - Creating a Mission Graph

In this section, it was intended that volunteers created a mission graph which used the lock and key ports.

Graph to be created: name: test quest; description: This graph has 3 missions. Each mission has 1 task and 1 reward. The 3 missions are successors to the starting node. One of the missions is a lock and another of the missions is the key to that same lock mission. It is intended that the tasks and rewards in each mission will be those created in the previous sections. If you were unable to create them, existing ones can be used.

The volunteers had as an objective to perform the following tasks: Open the mission graph management window; Open the mission graph creation window; Add new nodes according to the number of intended missions; Add the desired sequence links between missions; Choose which missions will be locks and which missions will be keys; Add links between key missions and respective lock missions; Add the respective tasks and rewards to each mission; Save the created mission graph.

All the volunteers were able to finish this section without help. The volunteers took between 2 minutes and 2 seconds and 3 minutes and 28 seconds to complete the tasks.

5.3.2.5 System Usability Scale (SUS)

After finishing all the previous sections, the volunteers were required to answer a post-session SUS questionnaire.

The questionnaire's results for each affirmation can be seen in Table 5.11. In odd affirmations, which are highlighted in blue, a higher value is better, and, on the contrary, in even affirmations, which are highlighted in red, a lower value is better. An affirmation worth emphasizing is the number seven, where, although the answers were still positive, it was the lowest scored. This indicates that the volunteers thought the system was not that easy for most people to learn quickly. Another affirmation also worth pointing out is the number ten, where volunteers thought they needed to learn some things before being able to start using the system.

Additionally in (fig. 5.6), each volunteer's independent calculated scores can be better visualized. The average of all scores was 91,25, which is higher than the SUS average of 68, indicating an overall positive result.

Table 5.11: Tool SUS Questionnaire Answers - Shortened Affirmations

<i>SUS Affirmations</i>	<i>Mean</i>	<i>Std.</i>	<i>Med.</i>	<i>1st Q.</i>	<i>3rd Q.</i>
1. I would use the system frequently.	4.66	0.47	5	-1	+0
2. The System is complex to use.	1.00	0.00	1	-0	+0
3. The System is easy to use.	4.83	0.37	5	-0	+0
4. I would need technical support.	1.00	0.00	1	-0	+0
5. The system has well integrated functions.	5.00	0.00	5	-0	+0
6. The system is inconsistent.	1.00	0.00	1	-0	+0
7. The system is easy to learn.	3.83	0.69	4	-1	+0
8. The system was cumbersome to use.	1.00	0.00	1	-0	+0
9. I felt confident using the system.	4.67	0.47	5	-1	+0
10. I needed to learn a lot beforehand.	2.66	0.47	3	-1	+0

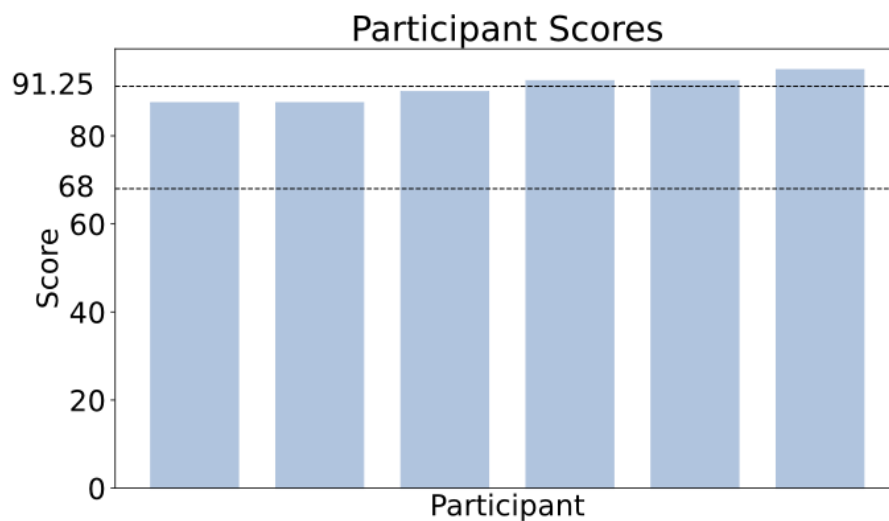


Figure 5.6: SUS Questionnaire Scores - Tool

CONCLUSIONS AND FUTURE WORK

This chapter contains an overview of the conclusions that were drawn from the work developed during this dissertation. After that, this chapter also outlines potential avenues for expanding and improving the solution in the future.

6.1 Conclusions

The work developed throughout this dissertation resulted in the creation of a set of various features to be integrated with an existing platform whose primary goal is to preserve and share communities' cultural heritage and stories while having Trafaria's neighborhood as a case study. The platform additionally seeks to serve as an educational tool that motivates users to explore and learn about diverse cultures both within and beyond it.

The implemented features take this goal into consideration and benefit from mechanisms such as augmented reality and gamification in order to draw the attention of users and maintain their engagement with the platform. By weaving in some game characteristics into augmented reality activities that incorporate cultural information, these features are able to offer users a fun and effective learning experience. The mentioned activities, also referred to as interactions, include puzzles, quizzes, and mini-games that are intended to be added as extra content for users to engage with when following the platform's planned routes, which go through diverse points of interest in a determined region. Additionally to these interactions, a missions system was also implemented as a means to unify them by offering users the possibility of following questlines and collecting rewards upon completing the various tasks that make up the missions. To try and achieve even higher user satisfaction, difficulty scaling was included in features that had the possibility of adapting to users' abilities.

In order to evaluate whether the implemented features truly fulfilled their intended objectives, various user tests were conducted. These tests aimed to assess various usability aspects of the features while also extracting valuable insights on experience aspects. Overall, the tests yielded positive results. Even users with little to no experience in augmented reality were able to intuitively engage in the various activities and successfully

complete the desired tasks, finding most of the interactions to be interesting, attractive, and enjoyable. This leads to the conclusion that the main objective of the solution was indeed successfully achieved. On a side note, the aspect of difficulty scaling in the various features, although also showing positive results, was found to be lacking and in need of improvement. The next section includes a topic discussing how this improvement can be achieved in the future.

Finally, the developed mission system also includes a built-in Unity tool that allows the creation of the necessary elements required to generate the missions. This tool has also undergone usability testing in order to assess its versatility and relevance in the scope of integrating missions into a game developed in Unity. Although user tests yielded surprisingly positive results, two limitations were identified during experiments in the content creation phases for the system's demonstration. These limitations relate to the inability to offer disjunctivity options in missions and task objectives, as well as the inability to offer tasks whose requisites must be met multiple times consecutively. These characteristics were not taken into consideration from the get-go since none of the initially designed elements had them. Subsequently, the next section includes a topic that explains how these functionalities can be easily incorporated in the future due to the way the system was implemented.

In summary, the objectives of this dissertation were achieved, and all contributions outlined in Subsection 1.4 were successfully realized. Nevertheless, the developed final result still offers significant potential for improvement, and, in this context, the platform continues to have room to incorporate endless ideas, with a vast sea of possibilities and features that can be explored.

6.2 Future Work

During the development of the various features and throughout the user testing sessions, several ideas for future research directions and overall improvements to the application emerged or were suggested by the volunteers. Below are some of these ideas:

Difficulty Scaling - As previously mentioned, effective difficulty scaling of certain features plays a crucial role in enhancing user satisfaction and maintaining motivation during interactions. Implementing a new and improved difficulty scaling system could involve utilizing more diverse performance metrics and evaluations. Moreover, artificial intelligence algorithms can be leveraged to automatically adjust features and suggest personalization aspects, thereby offering a more tailored and engaging experience for users.

Mission System - The missing functionality of disjunctivity in the mission system can be easily introduced due to the system's implementation. Concerning missions, a good and elegant solution would be, for example, to introduce new entries for a different type of edge between the nodes of the graphs. This new type of edge would, therefore, provide

the possibility of disjunction between missions. As for tasks, it would only be necessary to create a new subclass of the abstract *TaskVariable* class that would have the necessary components to support disjunction. Another less elegant but more practical solution would be to add an option to accumulative task variables that could make them disjunctive. Regarding tasks whose requisites must be met consecutively, for example, *Win X games in a row*, the same solution of creating a new subclass of *TaskVariable* applies.

Expanding the Platform - During the development of the [AR](#) interactions, their implementation took into consideration how the mobile component of the base platform uses the Android application to launch the Unity scenes that contain them. However, their integration with the mobile component was not fully realized due to unidentified issues concerning the handed-out Android section. Nonetheless, shortly before the end of this dissertation, the previous researcher discovered the cause of these problems and was in fact able to launch the developed interactions as they were. This leaves the possibility of, in the future, making the minor adjustments required to fully complete the integration.

Considering intriguing ideas for future platform features, the possibilities are abundant. For instance, leaderboards could be incorporated into each interaction, providing a competitive element. Additionally, different final rewards could be introduced for each interaction, which would perhaps be usable in a cosmetics shop where there could be, for example, items that would customize the appearance of interactions, such as puzzle borders or different 3D text fonts for quizzes. Furthermore, a dialogue system could be implemented, along with interactive characters that could accompany users on their learning adventures about different communities' cultures. Lastly, enabling users to communicate with each other and share their adventures and discoveries could be another valuable addition.

Apple's Vision Pro¹ - The emergence of Apple's new Vision Pro technology opens up new exciting and innovative research possibilities in the realm of augmented reality, which can be further expanded, much like in this dissertation's scenario, to contribute to cultural heritage preservation.

¹[Apple's Vision Pro website](#) - last accessed: 03/2024

BIBLIOGRAPHY

- [1] P. Agostinho. “Web Of Stories Mobile”. Unpublished Master’s Dissertation. NOVA School of Science and Technology, N.D (cit. on p. 20).
- [2] C. Ashmore and M. Nitsche. “Situated Play, Proceedings of DiGRA 2007 Conference The Quest in a Generated World”. In: *DiGRA conference* (2007) (cit. on p. 18).
- [3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. “Recent advances in augmented reality”. In: *IEEE Computer Graphics and Applications* 21 (6 2001-11), pp. 34–47. ISSN: 02721716. DOI: [10.1109/38.963459](https://doi.org/10.1109/38.963459) (cit. on p. 6).
- [4] B. B, K. K, K. N, D. A, S. K, S. S, and T. D. “3D Modelling by Means of Artificial Intelligence”. In: *Journal of Theoretical and Applied Information Technology* 31 (6 2021). ISSN: 1817-3195. URL: www.jatit.org (cit. on p. 12).
- [5] G. Barata, S. Gama, J. Jorge, and D. Gonçalves. “Improving Participation and Learning with Gamification”. In: *Proceedings of the First International Conference on Gameful Design, Research, and Applications* (2013). DOI: [10.1145/2583008.2583010](https://doi.org/10.1145/2583008.2583010). URL: <http://dx.doi.org/10.1145/2583008.2583010> (cit. on p. 16).
- [6] F. Bellotti, R. Berta, A. D. Gloria, A. D’Ursi, and V. Fiore. “A serious game model for cultural heritage”. In: *Journal on Computing and Cultural Heritage (JOCCH)* 5 (4 2013-01). ISSN: 15564673. DOI: [10.1145/2399180.2399185](https://doi.org/10.1145/2399180.2399185). URL: <https://dl.acm.org/doi/10.1145/2399180.2399185> (cit. on p. 15).
- [7] J. Brooke. “SUS: A quick and dirty usability scale”. In: *Usability Eval. Ind.* 189 (1995-11) (cit. on p. 66).
- [8] K. Caine. “Local Standards for Sample Size at CHI”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016). DOI: [10.1145/2858036.2858498](https://doi.org/10.1145/2858036.2858498). URL: <http://dx.doi.org/10.1145/2858036.2858498> (cit. on p. 66).
- [9] V. Cesário. “Guidelines for Combining Storytelling and Gamification: Which Features Would Teenagers Desire to Have a More Enjoyable Museum Experience?” In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (2019). DOI: [10.1145/3290607.3308462](https://doi.org/10.1145/3290607.3308462). URL: <https://doi.org/10.1145/3290607.3308462> (cit. on p. 17).

- [10] J. Cheng, K. Chen, and W. Chen. "Comparison of marker-based AR and markerless AR: A case study on indoor decoration system". In: *Lean and Computing in Construction Congress (LC3): Proceedings of the Joint Conference on Computing in Construction (JC3)* (2017), pp. 483–490. DOI: [10.24928/JC3-2017/0231](https://doi.org/10.24928/JC3-2017/0231) (cit. on p. 8).
- [11] J. Cortright. "How gentrification benefits long-time residents of low income neighborhoods". In: *City Commentary* (2019-07). URL: <https://cityobservatory.org/how-gentrification-benefits-long-time-residents-of-low-income-neighborhoods/> (visited on 2023-02-10) (cit. on p. 2).
- [12] D. Denomme. *Gentrification Pros and Cons*. 2022-04. URL: <https://study.com/learn/lesson/gentrification-pros-cons.html> (visited on 2023-02-10) (cit. on p. 2).
- [13] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. "From Game Design Elements to Gamefulness: Defining "Gamification"". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (2011). DOI: [10.1145/2181037.2181040](https://doi.org/10.1145/2181037.2181040) (cit. on p. 15).
- [14] C. Diaz, M. Walker, D. A. Szafir, and D. Szafir. "Designing for Depth Perceptions in Augmented Reality". In: *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2017, pp. 111–122. DOI: [10.1109/ISMAR.2017.28](https://doi.org/10.1109/ISMAR.2017.28) (cit. on p. 8).
- [15] J. Dormans. "Adventures in Level Design: Generating Missions and Spaces for Action Adventure Games". In: *Proceedings of the 2010 Workshop on Procedural Content Generation in Games* (2010). DOI: [10.1145/1814256.1814257](https://doi.org/10.1145/1814256.1814257) (cit. on p. 18).
- [16] J. Dormans and S. Bakkes. "Generating Missions and Spaces for Adaptable Play Experiences". In: *IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES* 3 (3 2011). DOI: [10.1109/TCIAIG.2011.2149523](https://doi.org/10.1109/TCIAIG.2011.2149523). URL: <http://ieeexplore.ieee.org>. (cit. on pp. 18, 19).
- [17] A. Dünser and M. Billinghurst. "Evaluating Augmented Reality Systems". In: *Handbook of Augmented Reality* (2011), pp. 289–307. DOI: [10.1007/978-1-4614-0064-6_13](https://doi.org/10.1007/978-1-4614-0064-6_13). URL: https://link.springer.com/chapter/10.1007/978-1-4614-0064-6_13 (cit. on p. 65).
- [18] M. M. Giakalaras. "Gamification and Storytelling". In: *Univ. Aegean* 8 (2016) (cit. on p. 17).
- [19] A. Hattab, I. Gonsheer, D. Moreno, and G. Taubin. "Differential 3D Scanning". In: *IEEE Computer Graphics and Applications* 38 (3 2017), pp. 43–51. ISSN: 02721716. DOI: [10.1109/MCG.2017.39](https://doi.org/10.1109/MCG.2017.39). URL: https://www.researchgate.net/publication/316501481_Differential_3D_Scanning (cit. on p. 14).
- [20] A. Holynski and J. Kopf. "Fast Depth Densification for Occlusion-aware Augmented Reality". In: *ACM Transactions on Graphics (ToG)* 37.6 (2018) (cit. on p. 8).

- [21] J. Hu, Y. Cheng, Z. Gan, J. Liu, J. Gao, and G. Neubig. "What makes a good story? designing composite rewards for visual storytelling". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 7969–7976 (cit. on p. 16).
- [22] C. C. Ibarra-Herrera, A. Carrizosa, J. A. Yunes-Rojas, and M. A. Mata-Gómez. "Design of an app based on gamification and storytelling as a tool for biology courses". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 13 (2008), pp. 1271–1282. DOI: [10.1007/s12008-019-00600-8](https://doi.org/10.1007/s12008-019-00600-8). URL: <https://doi.org/10.1007/s12008-019-00600-8> (cit. on p. 17).
- [23] N. Kashyap. *3C model of storytelling*. 2022-03. URL: <https://www.blackboardradio.com/post/3c-model-of-storytelling> (visited on 2023-02-10) (cit. on p. 16).
- [24] A. Katifori, M. Karvounis, V. Kourtis, S. Perry, M. Roussou, and Y. Ioanidis. "Applying interactive storytelling in cultural heritage: Opportunities, challenges and lessons learned". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11318 LNCS (2018), pp. 603–612. ISSN: 16113349. DOI: [10.1007/978-3-030-04028-4_70](https://doi.org/10.1007/978-3-030-04028-4_70) (cit. on p. 17).
- [25] J. Koivisto and J. Hamari. "The rise of motivational information systems: A review of gamification research". In: *International Journal of Information Management* 45 (2019-04), pp. 191–210. ISSN: 02684012. DOI: [10.1016/j.ijinfomgt.2018.10.013](https://doi.org/10.1016/j.ijinfomgt.2018.10.013) (cit. on p. 15).
- [26] D. van Krevelen and R. Poelman. "A Survey of Augmented Reality Technologies, Applications and Limitations". In: *International Journal of Virtual Reality* 9 (2 2010-01), pp. 1–20. ISSN: 2727-9979. DOI: [10.20870/IJVR.2010.9.2.2767](https://doi.org/10.20870/IJVR.2010.9.2.2767). URL: <https://ijvr.eu/article/view/2767> (cit. on pp. 6, 9).
- [27] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame. "Towards Collaborative Simultaneous Localization and Mapping: a Survey of the Current Research Landscape". In: *Field Robotics* (2021-08). DOI: [10.55417/fr.2022032](https://doi.org/10.55417/fr.2022032). URL: <http://arxiv.org/abs/2108.08325><http://dx.doi.org/10.55417/fr.2022032> (cit. on p. 7).
- [28] G. A. Lee, A. Dunser, S. Kim, and M. Billinghurst. "CityViewAR: A mobile outdoor AR application for city visualization". In: *11th IEEE International Symposium on Mixed and Augmented Reality 2012 - Arts, Media, and Humanities Papers, ISMAR-AMH 2012* (2012), pp. 57–64. DOI: [10.1109/ISMAR-AMH.2012.6483989](https://doi.org/10.1109/ISMAR-AMH.2012.6483989) (cit. on p. 11).
- [29] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. "The Digital Michelangelo Project: 3D Scanning of Large Statues". In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000) (cit. on p. 14).
- [30] V. Lombardo and R. Damiano. "Storytelling on mobile devices for cultural heritage". In: *New Review of Hypermedia and Multimedia* 18 (1-2 2012-03), pp. 11–35. ISSN: 13614568. DOI: [10.1080/13614568.2012.617846](https://doi.org/10.1080/13614568.2012.617846) (cit. on p. 17).

-
- [31] J. M. Lourenço. *The NOVAthesis L^AT_EX Template User's Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/main/template.pdf> (cit. on p. ii).
 - [32] J. Malpas. "Cultural Heritage in the Age of New Media". In: *New Heritage: New Media and Cultural Heritage* (2007-01), pp. 13–26. DOI: [10.4324/9780203937884-9/CULTURAL-HERITAGE-AGE-NEW-MEDIA-JEFF-MALPAS](https://doi.org/10.4324/9780203937884-9/CULTURAL-HERITAGE-AGE-NEW-MEDIA-JEFF-MALPAS). URL: <https://www.taylorfrancis.com/chapters/edit/10.4324/9780203937884-9/cultural-heritage-age-new-media-jeff-malpas> (cit. on p. 6).
 - [33] D. Neamtu, R. Comes, J.-J. Fang, D.-M. Popovici, T. A. Syed, M. S. Siddiqui, H. B. Abdullah, S. Jan, A. Namoun, A. Alzahrani, A. Nadeem, and A. B. Alkhodre. "In-Depth Review of Augmented Reality: Tracking Technologies, Development Tools, AR Displays, Collaborative AR, and Security Concerns". In: *Sensors* 23.1 (2022). ISSN: 1424-8220. DOI: [10.3390/s23010146](https://doi.org/10.3390/s23010146). URL: <https://doi.org/10.3390/s23010146> (cit. on p. 7).
 - [34] A. L. Negruşa, V. Toader, A. Sofică, M. F. Tutunea, R. V. Rus, A. J. Vasile, P. Catalin, and Z. Marian. "Exploring Gamification Techniques and Applications for Sustainable Tourism". In: 7 (2015), pp. 11160–11189. ISSN: 2071-1050. DOI: [10.3390/su70811160](https://doi.org/10.3390/su70811160). URL: www.mdpi.com/journal/sustainabilityArticle (cit. on p. 16).
 - [35] J. Nielsen. "A mathematical model of the finding of usability problems". In: *CHI '93: Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (1993). DOI: [10.1145/169059.169166](https://doi.org/10.1145/169059.169166). URL: <https://doi.org/10.1145/169059.169166> (cit. on p. 84).
 - [36] Z. Noh, M. S. Sunar, and Z. Pan. "A review on augmented reality for virtual heritage system". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5670 LNCS (2009), pp. 50–61. ISSN: 03029743. DOI: [10.1007/978-3-642-03364-3_7/COVER](https://doi.org/10.1007/978-3-642-03364-3_7/COVER). URL: https://link.springer.com/chapter/10.1007/978-3-642-03364-3_7 (cit. on p. 10).
 - [37] M. Ojer, H. Alvarez, I. Serrano, F. A. Saiz, I. Barandiaran, D. Aguinaga, L. Querejeta, and D. Alejandro. "Projection-based augmented reality assistance for manual electronic component assembly processes". In: *Applied Sciences (Switzerland)* 10 (3 2020-02). ISSN: 20763417. DOI: [10.3390/APP10030796](https://doi.org/10.3390/APP10030796) (cit. on p. 7).
 - [38] P. T. Palomino, A. M. Toda, W. Oliveira, L. Rodrigues, A. I. Cristea, and S. Isotani. "Exploring content game elements to support gamification design in educational systems: narrative and storytelling". In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)* 30 (1 2019-11), p. 773. ISSN: 2316-6533. DOI: [10.5753/cbie.sbie.2019.773](https://doi.org/10.5753/cbie.sbie.2019.773). URL: <http://ojs.sector3.com.br/index.php/sbie/article/view/8805> (cit. on p. 17).

- [39] C. Panou, L. Ragia, D. Dimelli, and K. Mania. "Geo-Information An Architecture for Mobile Outdoors Augmented Reality for Cultural Heritage". In: *ISPRS International Journal of Geo-Information* 7.12 (2018). DOI: [10.3390/ijgi7120463](https://doi.org/10.3390/ijgi7120463). URL: www.mdpi.com/journal/ijgi (cit. on p. 11).
- [40] C. Panou, L. Ragia, D. Dimelli, and K. Mania. "Outdoors Mobile Augmented Reality Application Visualizing 3D Reconstructed Historical Monuments". In: *GISTAM* (2018). DOI: [10.5220/0006701800590067](https://doi.org/10.5220/0006701800590067) (cit. on p. 11).
- [41] R. Pinto. "Web of Stories Web Authoring Platform For Interactive Narratives". Available at <http://hdl.handle.net/10362/159507>. Master Thesis. Nova School of Science and Technology, 2023-07 (cit. on p. 20).
- [42] M. Rahrig and A. Luib. "Sri Dalada Maligawa-3D-Scanning and Documentation of the Temple of the Sacred Tooth Relic at Kandy, Sri Lanka". In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2017). DOI: [10.5194/isprs-annals-IV-2-W2-229-2017](https://doi.org/10.5194/isprs-annals-IV-2-W2-229-2017) (cit. on p. 14).
- [43] G. Reitmayr and D. Schmalstieg. "Location based Applications for Mobile Augmented Reality". In: *Proceedings of the Fourth Australasian user interface conference on User interfaces 2003-Volume 18* (2003) (cit. on pp. 9, 10).
- [44] F. Remondino. "Remote Sensing Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning". In: *Remote Sens* 3 (2011), pp. 1104–1138. ISSN: 2072-4292. DOI: [10.3390/rs3061104](https://doi.org/10.3390/rs3061104). URL: www.mdpi.com/journal/remotesensingArticle (cit. on p. 15).
- [45] M. O. Riedl and R. M. Young. "From linear story generation to branching story graphs". In: *IEEE Computer Graphics and Applications* 26 (3 2006-05), pp. 23–31. ISSN: 02721716. DOI: [10.1109/MCG.2006.56](https://doi.org/10.1109/MCG.2006.56) (cit. on p. 17).
- [46] B. R. Robin and S. G. McNeil. "Digital storytelling". In: *The International Encyclopedia of Media Literacy* (2019), pp. 1–8 (cit. on p. 16).
- [47] J. Sauro and J. R. Lewis. "When designing usability questionnaires, does it hurt to be positive". In: *Conference on Human Factors in Computing Systems - Proceedings* (2011-05), pp. 2215–2224. DOI: [10.1145/1978942.1979266](https://doi.org/10.1145/1978942.1979266). URL: <https://typeset.io/papers/when-designing-usability-questionnaires-does-it-hurt-to-be-hocck38et2> (cit. on p. 67).
- [48] M. Schrepp, A. Hinderks, and J. Thomaschewski. *Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios*. Ed. by A. Marcus. Cham: Springer International Publishing, 2014, pp. 383–392. ISBN: 978-3-319-07668-3. DOI: [10.1007/978-3-319-07668-3_37](https://doi.org/10.1007/978-3-319-07668-3_37) (cit. on p. 66).
- [49] E. Selin. "10 Different types of 3D modeling techniques - Artisticrender.com". In: *Artistic Render* (). URL: <https://artisticrender.com/10-different-types-of-3d-modeling-techniques/> (visited on 2023-02-10) (cit. on p. 11).

- [50] C. G. Serna, R. Pillay, and A. Trémeau. “Data fusion of objects using techniques such as laser scanning, structured light and photogrammetry for cultural heritage applications”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9016 (2015), pp. 208–224. ISSN: 16113349. DOI: [10.1007/978-3-319-15979-9_20](https://doi.org/10.1007/978-3-319-15979-9_20). URL: https://link.springer.com/chapter/10.1007/978-3-319-15979-9_20 (cit. on p. 15).
- [51] J. A. L. D. Silva, F. Cristiano, S. Oliveira, D. Juliana, and S. Martins. “Gamificação e Storytelling como Estratégia Motivacional no Ensino de Programação”. In: *XVII SBGames. Foz do Iguaçu 17* (2016) (cit. on p. 17).
- [52] S. P. Singh, K. Jain, and V. R. Mandla. “A new approach towards image based virtual 3D city modeling by using close range photogrammetry”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (5 2014-05), pp. 329–337. ISSN: 21949050. DOI: [10.5194/ISPRSANNALS-II-5-329-2014](https://doi.org/10.5194/ISPRSANNALS-II-5-329-2014) (cit. on p. 14).
- [53] S. Sylaiou, K. Mania, F. Liarokapis, M. White, K. Walczak, R. Wojciechowski, W. Wiza, and P. Patias. “Evaluation of a cultural heritage augmented reality game”. In: *Cartographies of Mind, Soul and Knowledge* (2015), pp. 153–174 (cit. on p. 65).
- [54] T-Factor. *Trafaria Pilot*. 2022-09. URL: <https://trafaria.t-factor.eu/> (visited on 2023-02-10) (cit. on p. 2).
- [55] T. Tóth and J. Živčák. “A comparison of the outputs of 3D scanners”. In: *Procedia Engineering* 69 (2014), pp. 393–401. ISSN: 18777058. DOI: [10.1016/j.proeng.2014.03.004](https://doi.org/10.1016/j.proeng.2014.03.004) (cit. on p. 12).
- [56] Unesco. *Concept of Digital Heritage*. URL: <https://en.unesco.org/themes/information-preservation/digital-heritage/concept-digital-heritage> (visited on 2023-02-10) (cit. on p. 6).
- [57] V. Vlahakis, N. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, L. Almeida, D. Stricker, T. Gleue, I. T. Christou, and R. Carlucci. “Archeoguide”. In: *Virtual Reality, Archeology, and Cultural Heritage* 9.10.1145 (2001-11), p. 131. DOI: [10.1145/584993.585015](https://doi.org/10.1145/584993.585015). URL: https://www.researchgate.net/publication/220955275_ARCHEOGUIDE_first_results_of_an_augmented_reality_mobile_computing_system_in_cultural_heritage_sites (cit. on pp. 9, 10).
- [58] D. J. Wong. “60% of entire gaming market is now dominated by mobile gaming, study finds - Tech”. In: *Mashable SE Asia* (2022-05). URL: <https://sea.mashable.com/tech-1/20432/60-of-entire-gaming-market-is-now-dominated-by-mobile-gaming-study-finds> (cit. on p. 9).

- [59] B. Wyman, S. Smith, D. Meyers, and M. Godfrey. "Digital Storytelling in Museums: Observations and Best Practices". In: *Curator: The Museum Journal* 54 (4 2011-10), pp. 461–468. ISSN: 2151-6952. DOI: [10.1111/j.2151-6952.2011.00110.x](https://doi.org/10.1111/j.2151-6952.2011.00110.x). URL: <https://onlinelibrary.wiley.com/doi/full/10.1111/j.2151-6952.2011.00110.x> (cit. on p. 17).

USABILITY TESTS CONSENT FORM

Consentimento

Por favor, leia atentamente as informações abaixo antes de prosseguir:

Este formulário faz parte do estudo de testes de utilizador necessário para o desenvolvimento de uma tese de mestrado, realizada na FCT NOVA, na área de engenharia informática. A sua participação é voluntária e confidencial. Ao prosseguir com este formulário, está a concordar em participar neste estudo e dá permissão ao investigador principal e seus associados de apresentarem os resultados deste trabalho por escrito ou de forma oral em aulas ou apresentações académicas, sem necessidade de nenhuma autorização adicional sua, desde que a sua imagem e identidade não sejam reveladas nem divulgadas.

Participação voluntária:

A sua participação nos questionários é completamente voluntária. Se preferir não responder a uma questão ou se decidir interromper algum questionário a qualquer momento, pode fazê-lo sem qualquer penalização.

Recolha de informações:

Todas as informações recolhidas nestes questionários serão utilizadas apenas para os propósitos do estudo mencionado acima e serão tratadas de forma confidencial e anónima.

Confirmo que li e compreendi as informações acima e consinto participar e dar seguimento a este formulário.

☐ Consinto participar voluntariamente

USER EXPERIENCE QUESTIONNAIRE (UEQ)

RESULTS

<i>Left</i>	<i>Right</i>	<i>Mean</i>	<i>Var.</i>	<i>Std.</i>
annoying	enjoyable	2,5	0,4	0,6
not understandable	understandable	2,1	0,7	0,8
creative	dull	1,9	0,9	1,0
easy to learn	difficult to learn	2,0	0,4	0,6
valuable	inferior	2,4	0,5	0,7
boring	exciting	1,7	1,0	1,0
not interesting	interesting	2,1	0,6	0,8
unpredictable	predictable	1,1	0,8	0,9
fast	slow	2,4	0,3	0,5
inventive	conventional	1,9	1,4	1,2
obstructive	supportive	1,4	1,1	1,1
good	bad	2,6	0,4	0,6
complicated	easy	2,0	1,3	1,1
unlikable	pleasing	2,6	0,4	0,6
usual	leading edge	1,6	0,5	0,7
unpleasant	pleasant	1,8	0,4	0,7
secure	not secure	2,7	0,3	0,6
motivating	demotivating	1,6	1,3	1,1
meets expectations	does not meet expectations	2,1	0,6	0,7
inefficient	efficient	1,9	0,9	1,0
clear	confusing	1,8	0,7	0,8
impractical	practical	2,2	0,5	0,7
organized	cluttered	2,7	0,2	0,5
attractive	unattractive	2,2	0,6	0,8
friendly	unfriendly	2,6	0,5	0,7
conservative	innovative	1,5	0,6	0,8

C

CREATED FLYERS

Antigo Presídio



Question

Q_text

Para que foi o antigo presídio originalmente construído (1565) ?

Answer 1

A1_text

Quarentena de viajantes

A1_is Right Answer



Answer 2

A2_text

Forte defensivo

A2_is Right Answer



Answer 3

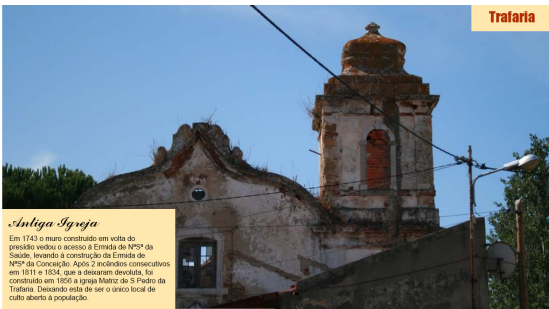
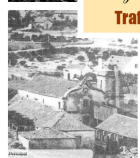
A3_text

Residência nobre

A3_is Right Answer



Antiga Igreja



Question

Q_text

Após expropriação dos terrenos do seu produtor em 1965/75, o interior da igreja passou a funcionar como...

Answer 1

A1_text

Centro recreativo e cultural

A1_js Right Answer

☐

Answer 2

A2_text

Fábrica de conserva de peixe

A2_js Right Answer

☐

Answer 3

A3_text

Vacaria para venda de leite

A3_js Right Answer

☒

Antigo Cinema



Question

Q_text

Em que ano foi criada a Sociedade do Salão Cinematográfico Trafeirense ?

Answer 1

A1_text

1914

A1_is Right Answer

☐

Answer 2

A2_text

1921

A2_is Right Answer

☒

Answer 3

A3_text

1927

A3_is Right Answer

☐

Cais de Embarque



Question

Q_text

Em que ano foram concluídas as obras de reabilitação do cais e demolido o pontão original (lado poente) ?

Answer 1

A1_text 2004

A1_is Right Answer ☐

Answer 2

A2_text 2007

A2_is Right Answer ☒

Answer 3

A3_text 2010

A3_is Right Answer ☐

Camionetas Piedense



SYSTEM USABILITY SCALE (SUS)

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.

II

USER EXPERIENCE QUESTIONNAIRE (UEQ)

	1	2	3	4	5	6	7		
annoying	☐	☐	☐	☐	☐	☐	☐	enjoyable	1
not understandable	☐	☐	☐	☐	☐	☐	☐	understandable	2
creative	☐	☐	☐	☐	☐	☐	☐	dull	3
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn	4
valuable	☐	☐	☐	☐	☐	☐	☐	inferior	5
boring	☐	☐	☐	☐	☐	☐	☐	exciting	6
not interesting	☐	☐	☐	☐	☐	☐	☐	interesting	7
unpredictable	☐	☐	☐	☐	☐	☐	☐	predictable	8
fast	☐	☐	☐	☐	☐	☐	☐	slow	9
inventive	☐	☐	☐	☐	☐	☐	☐	conventional	10
obstructive	☐	☐	☐	☐	☐	☐	☐	supportive	11
good	☐	☐	☐	☐	☐	☐	☐	bad	12
complicated	☐	☐	☐	☐	☐	☐	☐	easy	13
unlikable	☐	☐	☐	☐	☐	☐	☐	pleasing	14
usual	☐	☐	☐	☐	☐	☐	☐	leading edge	15
unpleasant	☐	☐	☐	☐	☐	☐	☐	pleasant	16
secure	☐	☐	☐	☐	☐	☐	☐	not secure	17
motivating	☐	☐	☐	☐	☐	☐	☐	demotivating	18
meets expectations	☐	☐	☐	☐	☐	☐	☐	does not meet expectations	19
inefficient	☐	☐	☐	☐	☐	☐	☐	efficient	20
clear	☐	☐	☐	☐	☐	☐	☐	confusing	21
impractical	☐	☐	☐	☐	☐	☐	☐	practical	22
organized	☐	☐	☐	☐	☐	☐	☐	cluttered	23
attractive	☐	☐	☐	☐	☐	☐	☐	unattractive	24
friendly	☐	☐	☐	☐	☐	☐	☐	unfriendly	25
conservative	☐	☐	☐	☐	☐	☐	☐	innovative	26



2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!

2024 Webinars: Start your journey with our expert advice today!