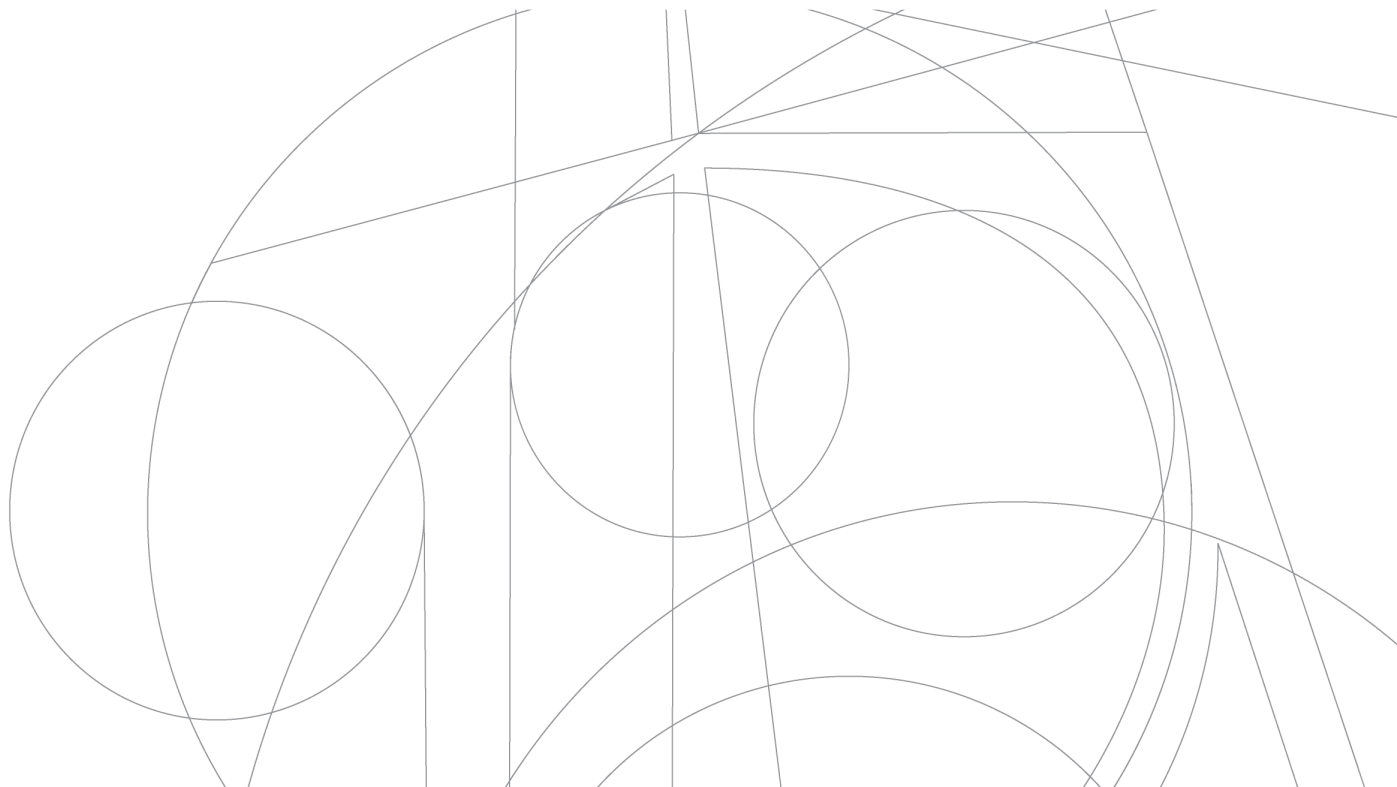
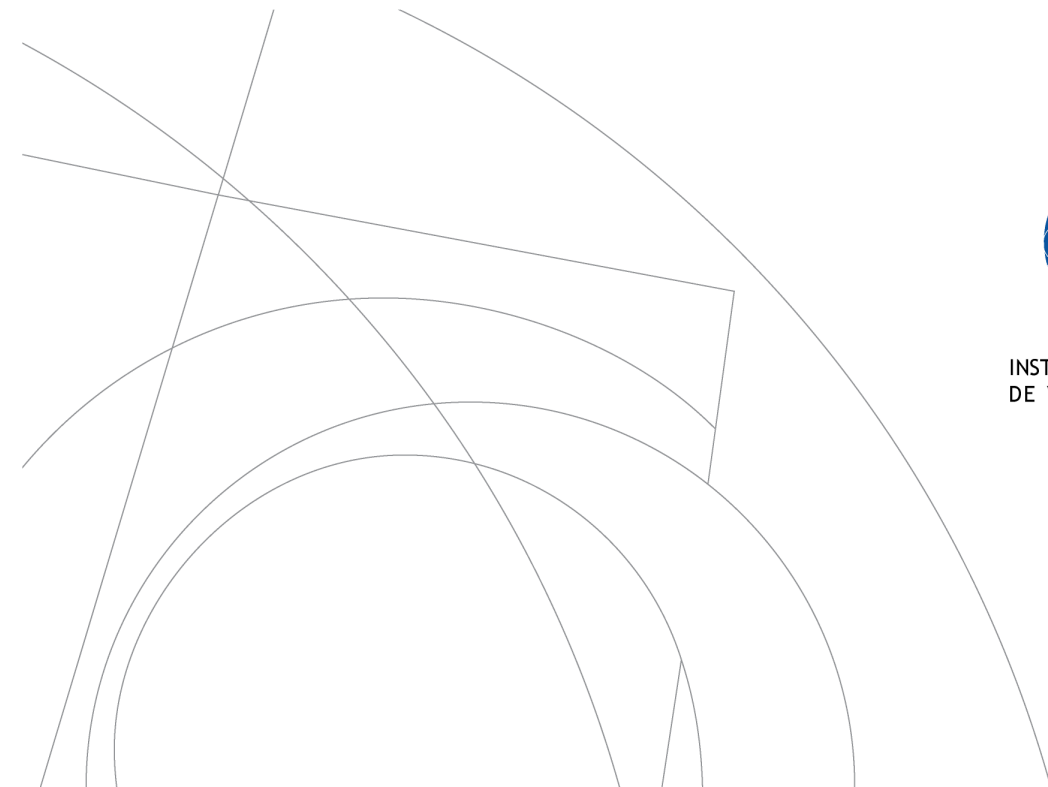




FRAMEWORK FOR CREATING AR EXPERIENCES

João Nuno Paiva Azevedo

2022



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

FRAMEWORK FOR CREATING AUGMENTED REALITY (AR) EXPERIENCES

João Nuno Paiva Azevedo



Escola Superior de Tecnológica e Gestão



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

João Nuno Paiva Azevedo

FRAMEWORK FOR CREATING AUGMENTED REALITY (AR) EXPERIENCES

Nome do Curso de Mestrado
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação do
Professor Doutor Luís Miguel Cabrita Romero
Professor Doutor Pedro Miguel Teixeira Faria

November 2022

ABSTRACT

This work proposes the architecture of a system whose main goal focuses on implementing a complete framework for creating augmented reality (AR) experiences, allowing creators to digitize reality to create their storylines. Moreover, it is an internal process with the objective of merging/grouping multimedia content, enabling clear and intuitive navigation within infinite augmented realities (based on the captured real world). This way, the user can create points of interest within their parallel realities, allowing them to navigate and traverse their new augmented worlds through an AR experience.

INDEX

ABSTRACT	2
LIST OF CONTENTS	5
Glossary	6
Chapter 1 - Introduction	8
1.1 - Historical Background & Motivation	8
1.2 - Methodology & Objectives	9
1.3 - Dissertation Structure	12
Chapter 2 - Related Work	13
2.1 - Real-World Acquisition and Digitization	13
2.2 - Augmented/Virtual Reality Application	14
2.3 - Multimedia and Hypermedia	15
2.4 - Storytelling with Augmented Reality	16
Chapter 3 - Specifications	18
3.1 - Software & Hardware	18
3.1.1 - Software	18
3.1.2 - Hardware	20
3.2 - System Architecture	21
3.2.1 - Components Installation	21
3.2.2 - Components Integration	22
Chapter 4 - Implementation	26
4.1 - Digitalization of Reality	27
4.2 - Multimedia Content Creation	36
4.3 - Multimedia in Relation to the Digital World	39
4.4 - User Navigation within Digital Worlds	44
4.5 - Installation and Use of the Framework	49
4.5.1 - Creator App Screens	49
4.5.2 - End User App Screens	52

4.6 - Use Cases	53
Chapter 5 - User Experiences & Tests	56
5.1 - SUS Survey	58
5.2 - Results Analysis	59
Chapter 6 - Conclusions	61
Chapter 7 – Bibliography	63

LIST OF CONTENTS

Figure 1 - Creator's APP features	10
Figure 2 - End User's APP features	11
Figure 3 - Conceptual architecture	18
Figure 4 - Components architecture	24
Figure 5 - System Data Structure	25
Figure 6 - Framework architecture	26
Figure 7 - Framework mockups	28
Figure 8 - Interaction between creator and framework for object mapping	29
Figure 9 - Connection between captured photos	31
Figure 10 - Incorrect approach to mapping	31
Figure 11 - Optimal approach to mapping	32
Figure 12 - Mapping an outdoor point of interest	32
Figure 13 - Mapping outdoor entrance, wall, etc...	33
Figure 14 - Combining "Outside-in" method with map alignment	34
Figure 15 - Framework menu for Anchor Image addition	35
Figure 16 - Framework menu for Map Alignment generation	36
Figure 17 - Immersal Developer Portal	36
Figure 18 - Information and different types of generated map files	37
Figure 19 - Map construction from point-cloud mapping	38
Figure 20 - Framework content creation menus	40
Figure 21 - Framework "Map Download" feature	42
Figure 22 - Multimedia selection screen	43
Figure 23 - Interaction with the multimedia present on the augmented world	45
Figure 24 - New augmented world, maps and respective multimedia relation	46
Figure 25 - Framework's new storyline creation feature	47
Figure 26 - JSON structure for new augmented world	48
Figure 27 - Waypoints and targets connections	49
Figure 28 - Storyline visualization screen	50
Figure 29 - Scanned Maps Creation and Alignment	52
Figure 30 - Multimedia Creation	53
Figure 31 - Adding Content to Augmented World	53
Figure 32 - Create Augmented World and Storyline	54
Figure 33 - Augmented World Navigation and Experience	55
Figure 34 - Indoor Use Case	56
Figure 35 - Outdoor Use Case	56
Figure 36 - Mapped points of interest from Forte Santiago da Barra	59
Figure 37 - Escola Superior de Tecnologia e Gestão (Entrance)	60
Table 1 - Technologies and algorithms used in AR applications	8
Table 2 - Comparison between reality capture software	19
Table 3 - Comparison between AR development tools	20

Glossary

Term	Definition
Augmented Reality (AR)	Enhanced version of the real physical world achieved through the use of digital elements
Global Positioning System (GPS)	A system that can show the exact position of a person or thing by using signals from satellites
Hypermedia	A combination of videos, images, etc. that are connected together and allow interaction between them and the user
Simultaneous Localization And Mapping (SLAM)	Method used for autonomous vehicles that lets you build a map and localize your vehicle at the same time.
Light Detection And Ranging (LiDAR)	A popular remote sensing method used for measuring the exact distance of an object on the earth's surface.
Software Development Kit (SDK)	Set of software tools and programs used by developers to create applications for specific platforms.
System Usability Scale (SUS)	A questionnaire that offers a quick and effective way to evaluate the suitability of a product or design
Virtual Reality (VR)	Computer-generated environment with scenes and objects that appear to be real creating a sense of immersion for the user.
Scanned Map (SM)	Point-cloud map automatically constructed by a real-world scanning process

Augmented World (AW)

The real world with an extra layer of reality (AR) seen and interactible through the user's device. Consists of several scanned maps put together, creating a new world.

AR Experiences Creator (Creator)

User able to digitize the real-world, create new multimedia content, prepare scanned maps and create new augmented worlds to be navigated and experienced by an end user.

Chapter 1 - Introduction

In this chapter, the historical background and motivation are defined alongside the objectives and methodology for the execution of the presented work.

1.1 - Historical Background & Motivation

The acquisition and registration of real/augmented content is currently a topic under much investigation in the context of Augmented and Virtual Reality applications. The quality and performance of these applications are highly dependent on the accuracy of that initial registration.

It is relevant to mention that, according to the study carried out in (W. Liu, 2019), although there is already a considerable amount of Augmented Reality projects (both under development and already developed) these are, in a vast majority, designed and applied to indoor environments. Some inconsistent outdoor variables such as the size or scale of objects, the instability of the natural lighting, and the high number of objects can easily distort and affect the usage of such software in that domain.

Back in the 90s, Virtual Reality technology started to become used more frequently due to the increase in its popularity. At this time, Augmented Reality, which emerged a bit as an offshoot of Virtual Reality, began to capture the attention of more researchers who quickly saw enormous potential in this new technology, as it could be applied in the most diverse sectors as a support tool.

<i>Environment</i>	<i>Most Important Technologies</i>
Indoor	Marks & Simultaneous Localization And Mapping (SLAM)
Outdoor	GPS & Simultaneous Localization And Mapping (SLAM)

Table 1 - Technologies and algorithms used in AR applications

Nowadays, therefore, it is common to find these applications associated with mobile devices, both handheld and headset. We can categorize these into two different research categories (TABLE I): indoor and outdoor; both with their advantages, disadvantages and specific conditions of use.

Naturally, navigation carried out inside a building makes the use of GPS (Global Positioning System) difficult due to its inability to obtain accurate results at a low cost. A simple and effective way to address this problem is by using marker-based systems. This involves placing marks in strategic places inside buildings so the user can navigate through the system. SLAM (Simultaneous Localization and Mapping) and GPS are commonly used in the outdoor environment, as shown in (M. W. M. G. Dissanayake, 2001). Although these technologies and algorithms are widely used in robotics, they cannot guarantee the user perfect performances or real-time data. In addition to robotics, the SLAM algorithm has also served as a basis and a key element to develop autonomous vehicles (land, air, and water) as observed in (P. Wu, 2018) (F. Demim, 2018).

Covering large areas using marker-based systems can be challenging, especially outdoors. This can be considered a limitation since all marks must be manually prepared and placed in the locations to be explored due to the absence of an automated/intelligent alternative to this process.

In addition to navigation, the relationship between the individual multimedia contents is also an important aspect of an Augmented Reality application, since these are two problems that are commonly encountered. In simple and intuitive ways, these applications should allow users to always know where they are and where their destination is. It is especially important in systems where the user can choose manually so that the navigation is as straightforward as possible. It is known as situated hypermedia when 3D content presented in real-world locations is observable and modifiable in these applications.

1.2 - Methodology & Objectives

The main objective is to develop an entire framework that allows the user to fully control this mixture between the real world and the augmented world, from the moment of capturing reality to the creation of content and navigation between infinite worlds. Thus, the user will be able to easily create and control their augmented reality application.

The implementation of the proposed system can be divided into 3 parts:

1. Preparation of Augmented Reality: this first phase will involve studies and tests on how to obtain and digitize physical parts of reality to build

scanned maps (SM) of any given point of interest. These will be used as a bridge between the real world and the augmented worlds.

2. Fusion/merger of models: after determining a way to achieve the previous step, all the models need to be integrated into each augmented world (AW) correctly to provide an accurate and credible AR experience. This step involves correctly categorizing the various types of media/models and relating them to the scanned maps (SM) obtained during the first step.
3. User navigation/AR experience: the previous steps lead to the final objective of the project - to allow an end user to experience a new augmented world prepared by a creator. Once the relationships between all the contents of the system are set, the creator is able to create and edit multiple versions/perspectives of reality. In other words, creating multiple parallel augmented realities based on different points of interest.

Following the identification of the central goals of the project, it is necessary to point out that this framework will consist of two systems that share numerous functionalities but have different users and utilities. The framework will then be divided into two parts: one dedicated to the creator and another to an end user.

The first part developed will be the system for the creator and it includes a large portion of the second part. In Figure 1, there is an illustration of all the creator app's features:

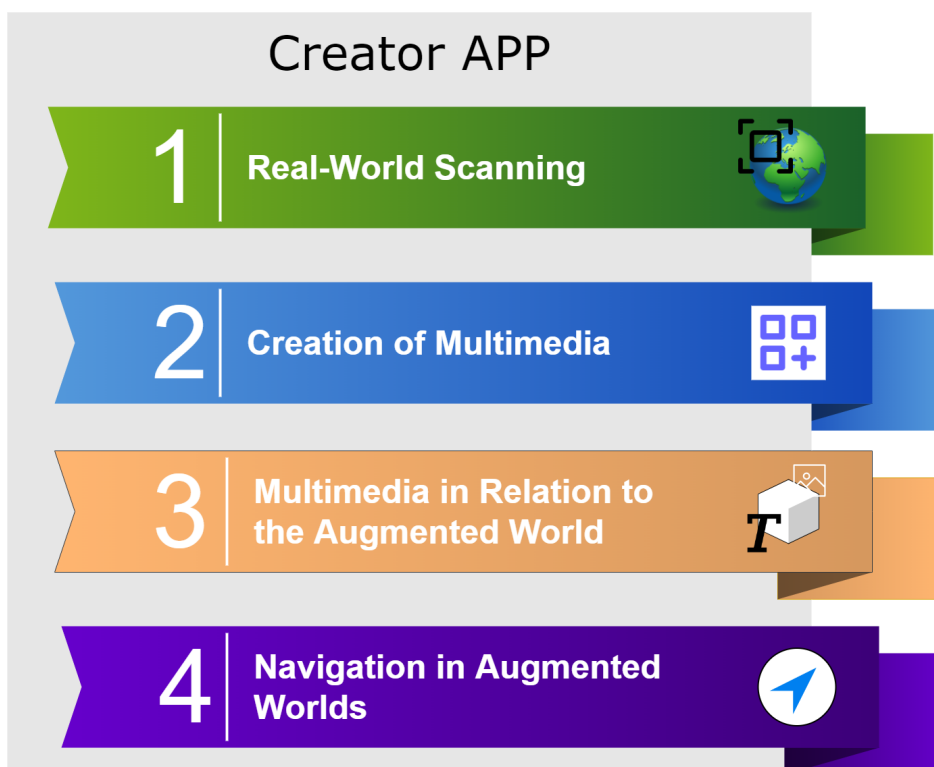


Figure 1 - Creator's APP features

- Real-world scanning: the placement of augmented elements on a scanned map requires not only the use of GPS coordinates but also the digitization of parts of reality. Both GPS coordinates and scanned maps, combined, will be recognized as points of interest later, by the user's camera when pointed in the direction of the real-world object
- Creation of multimedia: creating augmented content is a step that allows users to create new digital elements (e.g., texts, images, videos). Later, these can be added into the augmented world while editing any given scanned map
- Multimedia in relation to the augmented world: in a scanned map, each point of interest may contain a variety of augmented objects added and positioned by the creator. Augmented elements must be associated with a position relative to their parent (point of interest) so that the entire multimedia content remains in the exact position defined by the creator
- Navigation in augmented worlds: new augmented worlds are formed by the confluence of several scanned maps (points of interest) that, in turn, have several types of multimedia content. A set of points of interest can be chosen by the user to create a map that can be used to build a route to reach a certain destination, which will be the last point of interest in that particular augmented world's experience

The second part includes a visualization of all the contents previously prepared by creators, along with the option to navigate through different routes, following stories and points of interest. Figure 2 shows the features of the User APP:

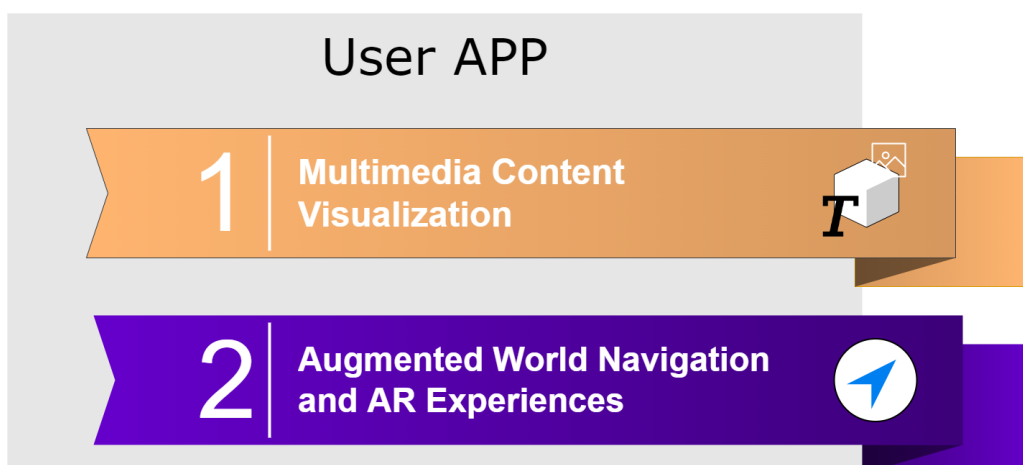


Figure 2 - End User's APP features

- Multimedia content visualization: all augmented content placed under each point of interest will be visible to users, along with the scanned maps built by the creators
- Augmented world navigation and AR experience: the user's main objective is to traverse an augmented world previously prepared by a creator, visiting each point of interest, following a route, and arriving at a final destination within the chosen AW

1.3 - Dissertation Structure

To answer essential questions related to the limitations and usage of technologies such as real-world scanning, multimedia/hypermedia in AR applications and AR experiences with storytelling, this work will proceed to review the related work critically in Chapter 1. Chapter 2 will then be divided into four main topics of study: real-life acquisition/digitization, AR/VR application, multimedia and hypermedia with AR, and storytelling with AR. All the necessary software and hardware along with all the designed system architectures for this Framework will be discussed in Chapter 3. All the required installation and integration steps will also be demonstrated. Chapter 4 shows the implementation of the system is explained and demonstrated. This chapter is divided into six subsections: digitization of reality, multimedia content creation, multimedia concerning the digital world, storytelling and route navigation, installation and use of the framework and, at last, use cases. Moreover, all the critical user experiences and tests carried out during its implementation and at the final stage of the project will be shown and discussed in Chapter 5. Finally, Chapter 6 will include all the conclusions that summarize the implementation and usage of this Framework. Furthermore, all current limitations and future work will be reported in this final chapter.

Chapter 2 - Related Work

The analysis in this section includes four main topics: real-world acquisition and digitization, augmented/virtual reality applications, hypermedia, and points of interest/real-world perspectives. These themes were focused since they are directly related to the next section (Chapter 3).

2.1 - Real-World Acquisition and Digitization

In projects like (W. Liu, 2019) (S. Skuratovskyi, 2019) the authors propose methods to acquire and digitize parts of outdoor reality to subsequently obtain virtual 3D/2D models in a more automated manner. Using images captured from the air/terrestrial and related information, such as position and orientation, enabled to map a large outdoor area reasonably accurately. It is thus verified that a way to obtain more automated and accurate digitization of one or several objects within an area is through capturing an initial 2D model aerial view, constituting a kind of plan of the building(s) and its surroundings.

A study carried out by (Pileun Kima, 2018) proposes a framework capable of building point-cloud maps in real time. Based on this article, it is important to highlight some of these point-cloud mapping limitations:

- lack of key features in the mapping sites make it difficult to build an accurate point-cloud registration
- collecting data manually to provide key features and overlapping the same areas from different perspectives requires some time and effort

Mapping with photogrammetric technology has been an evolving field in recent years and, in 2015, (Hyojoon Bae, 2015) proposed a pixel-accurate method for 3D modeling and point-cloud mapping using mobile devices' cameras. They pointed out a strong advantage of this approach which is the availability of material and texture data for each model. This allows for improved recognition when pointing a mobile camera into a real-world object/area, as it detects both key features and the materials, as seen in (Andrey Dimitrov, 2014). Even though these are great advantages, this method still has some limitations. For example, when objects are moved in the real world after the point-cloud map has been generated, it may fail to recognize when compared to the original mapped area.

Other technologies and methods have been explored and investigated as they grow in popularity intending to reduce, or even solve, the limitations of

photogrammetry mapping. Two of these methods are the SLAM and the LiDAR-based approaches. According to (Pileun Kima, 2018) SLAM is often used in robotics as it enables robots to estimate their position and orientation while mapping their surroundings. This was used in (D. Roca, 2013) to build a 3D model of a real-world building with a Kinect sensor. The results revealed that a few discontinuity points were detected making it less accurate when compared to another 3D model of the same building obtained from a laser scanner.

Compared to other approaches, LiDAR technology is unable to detect colors and textures. Nonetheless, following the research of (Mani Golparvar-Fard) and (E.J. Jaselskis), it enables high-resolution, large-scale world mapping and isn't limited by external conditions such as weather, lighting, etc... during its operation. Even though a lot of great progress has been made to automate point-cloud data acquisition, (Pileun Kima, 2018) highlights that all of the above still relies on manual adjustments, made by engineers, which are of course subject to human error.

While analyzing and investigating this topic, we were able to conclude that the process of acquiring real-world content, converting it into a 2D or 3D, and then importing it to our system would be highly time-consuming, as well as resource-consuming for a step that should be done in real-time by a creator. This led our initial investigation to shift towards real-world digitization using a point-cloud mapping technology.

2.2 - Augmented/Virtual Reality Application

In a more general context regarding Augmented Reality applications, it is also important to highlight (C. C. Ho, 2019) (Yu, 2018) as they explore and demonstrate slightly more advanced techniques for the implementation of AR applications, in indoor and outdoor spaces, respectively.

In the study and project presented in (C. C. Ho, 2019), the implementation of two more efficient methods than the SLAM algorithm stands out. These are based on the study of the user's positioning through an ORB-visual-odometry estimation and image registration based on the boundary between the floor and the walls. According to the article, these achieve more reliable results in real-time.

On the other hand, in an outdoor environment, (Yu, 2018) proposes a navigation system that mostly uses a map (Baidu Map), similar to Bing Map, which, with the assistance of the Unity3D software, places a small avatar to indicate the correct path to the user. However, some problems were detected in its execution. Among them is the fact that the avatar does not always rotate in the

correct direction, thus confusing the user. Another problem is the instability of the hardware because as the device undergoes a constant change of position, this causes some problems with the avatar, which mirrors these changes in its position, making this not an ideal experience.

From the resulting analysis of the articles and projects discussed above, it appears that some tools and techniques for digitizing reality have already been explored, which facilitate, in a certain way, the acquisition of real models. However, the merging of these obtained models is still relatively unexplored.

2.3 - Multimedia and Hypermedia

Multimedia has always been highly present in AR applications as it allows the creator of these systems to directly provide feedback to a user by creating some guidelines, providing information, and even allowing interaction between the real world and the augmented world. In the study of (Chia-Yen Chen, 2014), an AR system was implemented to guide museum visitors through the museum. With this system, visitors can view more information about each point of interest within the museum, therefore having an enhanced/augmented experience provided by a multiple sets of multimedia (3D Models, Texts, Images).

In projects like (R. Nóbrega, 2017) we can visualize how interesting and engaging the usage of multimedia can be on an AR system. This mobile location-based AR application for urban tourist storytelling combines a regular mobile game with the power of AR to lead tourists to certain locations. Users are given a map within the app where they can view all the locations where there is a collectible augmented barrel model. Once the user reaches their destination, their score is increased. Various minigames such as virtual games and quizzes can also be played, which illustrates the endless possibilities when multimedia and AR are combined.

Many researchers have investigated and tried to define the best ways of visualizing hypermedia content and structures for a long time. According to (Feiner, 2006)], several techniques for grouping nodes were developed and implemented, using concepts such as hierarchy, abstraction, filtering, and organization, which makes navigation through the system easier and more interactive.

However, in examples like (P. Sinclair, 2002) (T. Starner, 1997) it appears that this study and research work was mostly done to deal with hypermedia structures in an indoor or small-scale context.

As noted above, (Feiner, 2006) explored techniques to allow users to browse and view. With this, whoever uses the developed system can lift and move

different objects, including entire buildings. To make this interaction as interesting as possible, the authors have placed hidden treasures that are only discovered after the user interacts with the hypermedia content for some time, thus making browsing more interesting.

In the context of AR applications, hypermedia content is used to provide users a way to navigate through different augmented dimensions, almost like a portal to a different world. This gives users different options and directions to experience their AR system. A clear implementation of the concept defined above makes it possible to create a multitude of realities. This allows the user to choose or lose himself in a set of infinite augmented worlds.

The work presented in (I. Herbst, 2008) discusses various types of presence from the perspective of the user, such as physical, spatial, temporal, and social. This project consists of the development of a small game based on Augmented Reality that allows the user to experience a journey across a city, through time and space, thus presenting realities of different temporal epochs. In the designed system, however, some problems arose in terms of the stable presentation of all the hypermedia as well as in obtaining and registering it.

2.4 - Storytelling with Augmented Reality

As our society evolves with digital technology, new forms of storytelling provide more context and are more engaging. According to (Ryokai, 2001) storytelling is a form of communication where people share understandings and experiences. As for the usage of AR in storytelling, recently, (Walczak, 2013) introduced a mobile AR authoring tool called MARAT and (Terrell Glenn, 2020) proposed a system called StoryMakAR, which merges electro-mechanical devices with virtual characters to create stories.

The authors of the article (John V. Pavlik, 2013) explored how AR storytelling can be applied to journalism and how it can become a reality. Newspapers could be a great starting point for this innovation, according to their study. Incorporating codes and marks next to the news could provide readers with access to videos or other types of multimedia related to it. Following that logic, the possibility to enhance our perception of reality and receive more information about certain news while standing next to where the event took place is also a strong option. Providing better-contextualized news content and engaging citizens in democratic discourse is a commendable goal.

Researchers have been working on automating the process of storytelling on a AR environment and, on projects such as (Changyang Li, 2022), it is noticeable how far this technology has evolved. They have created an automatic script generator with animated 3D models that position themselves correctly in a room

according to the position of the user. Even though the 3D models' behavior is fully automatic, it still requires some manual pre-preparations on the scene so that the spatial point-cloud map and the scene semantics are correctly placed.

Most of these prior arts are mainly focused on telling or showing pre-defined stories and include at least one manual preprocessing step in their system architecture. However, in recent years, some studies and projects such as (J. Bacca, 2019) and (J. Zubizarreta, 2019), aimed to achieve a complete framework to simplify the process of creating AR applications. Since both creating and maintaining AR applications require a lot of time and effort, the idea of automating these by allowing almost any person with any level of technological knowledge to create an AR system is really exciting. Both these works were applied on specific areas, like industry and education, and were not thought out for general use. Their frameworks were able to automatically detect some machinery parts and allowed users to share AR content which, in the educational field has the advantage of increasing student motivation.

The presented work focuses on providing creators with a fully autonomous AR storytelling framework in which they can prepare infinite storylines for users to navigate through and experience.

Chapter 3 - Specifications

As seen in Figure 3, a conceptual perspective, creators should be able to create augmented content (3D Text, Images and Videos) and AR experiences using world scanning and mapping algorithms from a middleware. These scanned maps and experiences would then be stored and aligned on a backend service.

Following this preparation, end users should then be able to visualize content and experience the worlds and stories previously built and put together by creators. All this data, processed and prepared in real-time by the middleware, is then shown for all end users to experience it through their mobile devices.

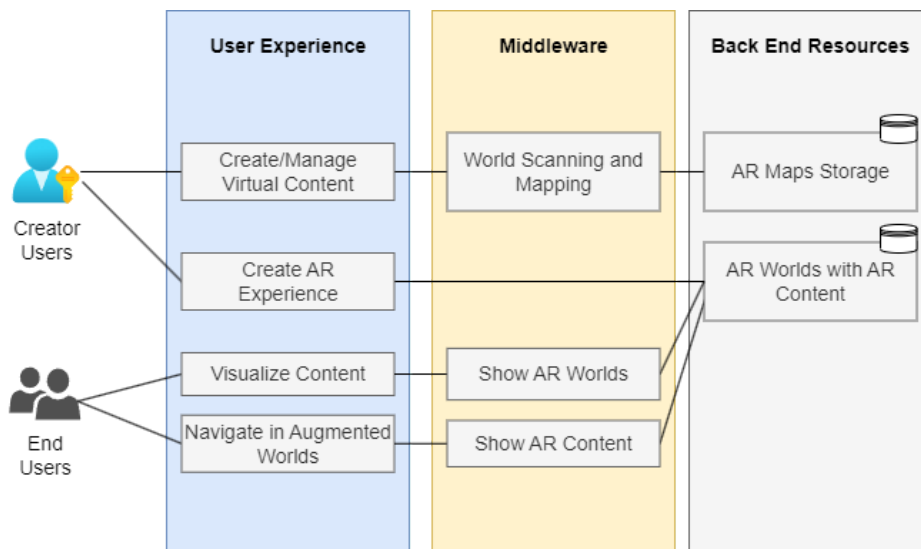


Figure 3 - Conceptual Architecture

3.1 - Software & Hardware

To implement this framework, a variety of technologies and software will be needed, as each of the steps described in the sections above requires different approaches. The system, however, is not solely dependent on technology. Due to its role as a framework for augmented reality, it is important to mention and reflect on the hardware necessary for its use. For a successful final product, the framework was continuously tested using a variety of devices in parallel with the development process.

3.1.1 - Software

A central engine, Unity 3D, was chosen as the platform to link all the components of the project, as it was an open-source game engine, recognized for its ability to support several augmented and virtual reality plugins and add-ons, allowing its full integration for both Android and iOS devices, as well as "headset" devices such as Microsoft HoloLens.

A few tests were carried out using a different engine (Unreal Engine), however, previous experiences creating AR applications and systems using Unity led us to choose Unity over Unreal for this work.

Having Unity 3D as the foundation of the framework, a second tool is needed to map outdoor and indoor areas to digitize reality, which is the first step of the system.

To choose only the most suitable software for the proposed project, two comparison tables are presented below, comparing all of the programs that were analyzed during the initial selection process.

Software	Standalone Software	Unity Integration	Real-world Mapping	Open Source
Meshroom	✓	✗	✓	✓
Insight3d	✓	✗	✓	✓
RealityCapture	✓	✗	✓	✗
Autodesk Recap	✓	✗	✓	✓

Table 2 - Comparison between reality capture software

The first four software to be tested and analyzed were Meshroom, Insight3D, RealityCapture, and Autodesk Recap. It was found that all these are great options when it came to mapping the real environment, and some even supported other 3D modeling software such as Autodesk Maya or Blender. While all of these tools are open source, except for RealityCapture, the biggest issue is that they cannot be used in real time by a creator, preventing visualizing the content right after digitization.

Furthermore, none of these technologies can be directly integrated into a Unity application, that is, photographs or videos must be captured by the creator in a first step, being then necessary to upload them to one of this software for digitization and creation of the spatial model of that piece of reality. Only after all

that, a .obj or .fbx file is generated that can be imported into Unity, this process also being carried out manually. The framework that is described and presented in this report requires technology capable of automating this initial step.

Importing 3D models to Unity in run-time is a very complex process that requires a lot of time and resources so we decided to look for other options. Instead of scanning and importing the whole 3D model of a point of interest, we started investigating a different way to digitize reality.

By then, the point-cloud map construction began to be explored and investigated. All that was required was to digitize the point of interest through photogrammetry mapping and then build a point-cloud of the scanned map and store its information on .ply, .glb, and .JSON structures.

Software	Standalone Software	Unity Integration	Real-world Mapping	Open Source
Vuforia SDK	×	✓	×	✓
Unity AR+GPS	×	✓	✓	✓
Immersal SDK	×	✓	✓	✓

Table 3 - Comparison between AR development tools

We then proceeded to analyze AR development tools: Vuforia SDK (Software Development Kit), Unity AR+GPS and Immersal SDK.

All of these programs integrate directly with the Unity Engine, making it possible to develop augmented reality applications of high quality. Moreover, they may also be used on Android and iOS devices, as well as on headsets.

The Vuforia Engine is a great SDK, especially when used indoors. This tool can be used to create multiple types of targets (image, model, and area targets), which the user's camera will recognize when pointing at them. Despite this, its free version allows the loading of only 20 area and model targets and the content must be imported through Vuforia's web developer portal, ensuring that the content complies with its predetermined rules of image/model quality. Therefore, this is too manual a step for what is intended of this paper.

The Unity AR+GPS plugin is a superb way of expanding the technology offered by Vuforia Engine since it works by mixing GPS data and AR data. This plugin uses a web map to place any type of augmented content on certain GPS coordinates. Due to GPS precision, we quickly discovered limitations when using this plugin. In good conditions, precision can range from 2 to 5 meters, and in bad conditions from 10 to 20 meters. Additionally, it does not work well indoors.

Lastly, the Immersal SDK software was analyzed. According to its website, it allows mapping up to 500-1,500m² of indoor space or even up to 5,000-10,000m² of open outdoor space. Using the free license, it is possible to map up to 1000 spatial maps per month. Through integration with Unity Engine, it also allows its visualization, allowing all of this to be done and prepared in just a few minutes.

Based on the comparative analysis carried out between the aforementioned software, the Unity Engine and the Immersal SDK were chosen as the main tools in the development of this framework.

Finally, it is important to note that it is also necessary to store some critical data from points of interest mapped from the real world to augmented content created by the user.

When it comes to storing digitized maps and points of interest, the Immersal SDK provides a web developer portal where all the spatial mappings performed are kept. The relationship between these maps and the augmented contents, however, must be stored and loaded from a second database.

For this step, we compared and analyzed a few different options for data storage: MongoDB, MySQL, PostgreSQL and Cassandra.

MySQL is a relational database system that stores data in tables, making it well-suited for applications that require complex data structures and relationships between entities. MongoDB, on the other hand, is a document-oriented database that stores data in JSON-like documents, making it well-suited for applications that deal with semi-structured or unstructured data. MongoDB offers better scalability options than MySQL, and is more efficient at handling large volumes of data.

PostgreSQL is a relational database system that is known for its robustness, extensibility, and support for advanced SQL features. Like MySQL, it stores data in tables, making it well-suited for complex data structures and relationships. However, it can be more difficult to set up and maintain than MongoDB. MongoDB offers better performance and scalability options, especially for applications that require high write throughput.

Cassandra is a distributed NoSQL database system that is designed for high availability and resilience. Like MongoDB, it stores data in JSON-like documents, but is better suited for write-heavy workloads and distributed environments. However, MongoDB offers better support for complex queries and data models, making it a better fit for applications that require more flexible data structures.

Overall, MongoDB is a versatile database management system that offers scalability, flexibility, and ease of use. It is a great choice for applications that require semi-structured or unstructured data management, and offers better scalability and performance options than MySQL and PostgreSQL. While Cassandra may be a better fit for write-heavy workloads and distributed environments, MongoDB's support for complex queries and data models makes it a top contender among other good database options available in the market.

Therefore, MongoDB was chosen as it is a lightweight option, easy to use and to integrate on Unity, perfect for quickly saving and loading .JSON data.

3.1.2 - Hardware

In terms of augmented reality, the following hardware will be used:

- 6.67" Smartphone: this device will allow us to carry out tests with the most common hardware in our daily lives as it can perfectly be used for an augmented reality application, both in an outdoor and in an indoor environment.
- 10" Tablet: using smartphones and tablets to conduct tests will provide a better picture of the differences that will occur between different resolutions. This will facilitate a standardization process to discover the best settings and properties, optimizing the framework.

3.2 - System Architecture

3.2.1 - Components Installation

In the development of this framework for creating augmented and virtual reality environments, three components must then be installed and integrated: Unity 3D, Immersal SDK, and MongoDB.

Considering that Unity 3D is the framework's engine and centerpiece, the first step is to download and install it, selecting the version we want to use, in this case, on a Windows x64 machine. The version of Unity installed to implement the proposed features is Unity 2020.3.18f1 as it is supported by the second component to be installed, the Immersal SDK, according to their documentation.

This initial installation takes a few minutes to finish, but once it is complete we can begin creating the application.

Due to Immersal's nature as a Software Development Kit, it must be integrated into Unity via the custom packages installation tool. There is a .unitypackage on the Immersal website that has been specifically designed to be imported into a Unity project. This includes Immersal's scripts and core, which can be loaded into an active project. In addition, it is also essential to import a number of additional packages to the project. These are dedicated to augmented reality application development and include the following: AR Foundation, ARCore XR Plugin, ARKit XR Plugin, and TextMesh Pro. These four additional packages are used by the main Immersal package so that it is possible to use all its features avoiding warnings and errors in the console.

On the Immersal developer portal, in addition to this Unity Package, it is provided a set of examples that contains a series of premade scenes to showcase some of the main features of this SDK. This can, therefore, be used as a basis for adding new layers of features relevant to the development of this framework. The main example focuses on the mapping functionality of the SDK, allowing the user to take photographs of an element of reality and then build a scanned map of it. There is also a scene for augmented content visualization under a certain point of interest and another scene for adding waypoints and targets under a set of previously loaded maps, aligned with each other. These examples were then kept and used as a starting point for the development of the framework.

Finally, MongoDB was the last component to be installed and added to the project architecture. Using MongoDB Atlas as a service, this initial procedure involves creating a cluster and preparing the necessary databases for system development.

3.2.2 - Components Integration

In Figure 4, there is a diagram that describes how these three components interconnect. It also describes how they relate to reach the objective initially defined for the development of a framework for creating AR experiences both indoor and outdoor, separated into two applications: creator version and user version.

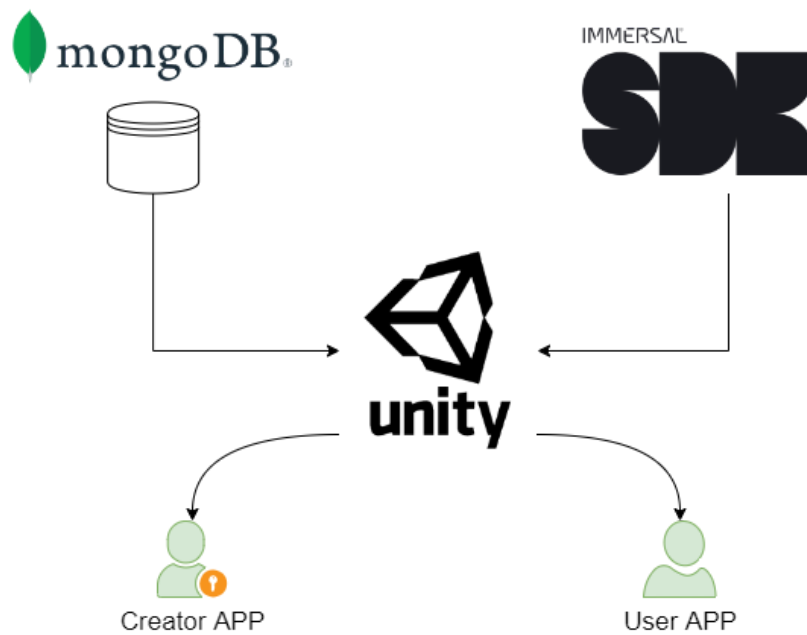


Figure 4 - Components architecture

As discussed in the previous subsection, the central game engine for the development of this framework is Unity 3D. It is, therefore, under this that all the layers necessary for the development of the framework will be installed and added, such as, the Immersal SDK and MongoDB, the first to handle the augmented reality component and the other to store the necessary data online.

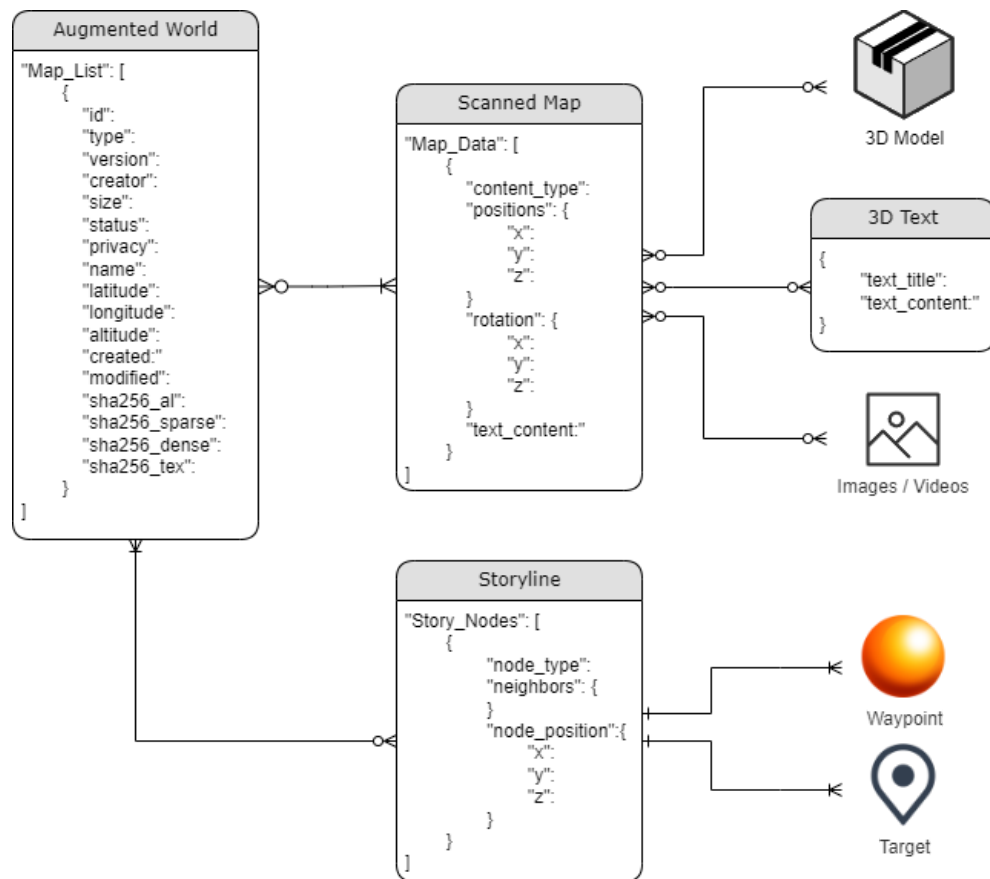


Figure 5 - System Data Structure

Figure 5 illustrates the system data structure. All the main data structures will be handled using .JSON which is a lightweight data-interchange format. Each augmented world is composed of zero to multiple scanned maps and each scanned map may contain multiple multimedia content. There is, then, the possibility of creating multiple storylines within each augmented world, each storyline formed by a set of different waypoints and targets.

In Unity, all the necessary functionalities were developed and implemented so that, both in the creator version and in the user version, all the proposed features and objectives are achievable.

In Figure 6 we have, in detail, the whole Framework architecture, illustrating how all the components relate to each other:

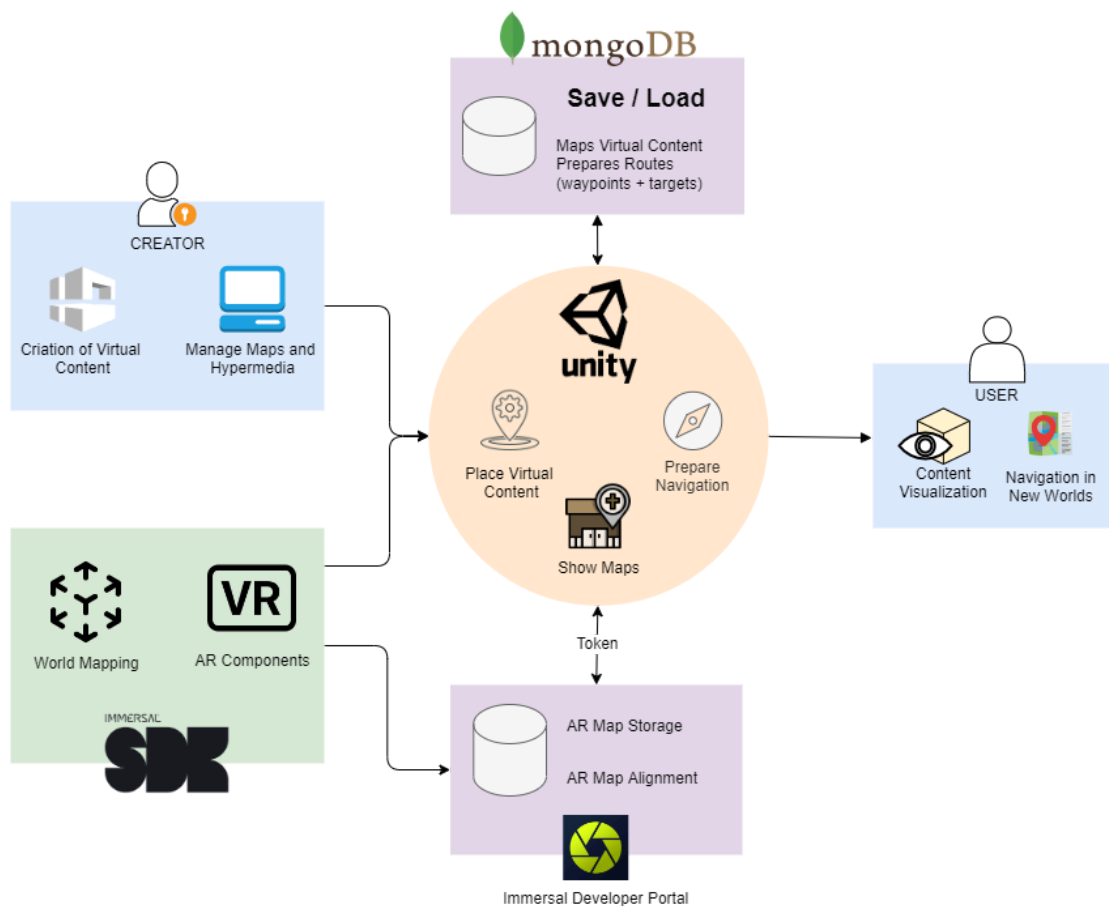


Figure 6 - Framework architecture

The integration of all components is then designed to allow the creator to prepare an infinity of augmented worlds so that they can later be viewed and explored by an end user.

On the left side, we see the first phase of using the framework. On the one hand, we have the creator that can create its augmented content: text and images, as well as manage all other available multimedia and augmented content like maps and 3D models. At the same time, the Immersal SDK allows the mapping of any points of interest that are necessary, whether in an outdoor or indoor environment.

All this augmented content of augmented reality components is sent and handled by the framework's central engine, Unity 3D. It is from here that the relationship between all the creator inputs and the Immersal components starts to be made. At this stage of the application, we have the first view of all multimedia content in the real world. Using the mobile device camera, the creator can add content to their new augmented world for the first time. By interacting with this content and clicking on it directly through the device, it is possible to manipulate its position and orientation ensuring it is exactly in the ideal position. This position is then stored in a MongoDB database and can be updated or edited when necessary while using the framework. In addition to

saving the type and position of various multimedia contents, the creator can also create navigation routes to be followed by end users. This is done by creating a new world and joining a set of different points of interest, ending in a target. Hence, the possibility of creating infinite worlds that can be explored by any user that has access to the token provided by the creator comes from this feature.

Each creator needs to create an account on the Immersal SDK developer portal, where they can manage all their scanned maps. It is also from this developer portal that the application will generate a unique token for the augmented worlds of each creator.

Once all this augmented experience is prepared, an end user will have access to a second application that will be responsible for the experience prepared by a creator. Through a valid creator token, this user can view and navigate through all the augmented worlds created by the creator, traversing all the waypoints to the different targets/destinations.

Finally, it is important to note that, to be able to make full use of the Framework, both the creator and the end user need to be connected to the internet. A network connection is required to allow the system to communicate with both the Immersal Developer Portal and the MongoDB database, to download, upload and create new scanned maps, so that all the multimedia can be successfully loaded and stored on each respective scanned map. Without this connection, the creator is still able to capture photos, however, they will not be able to create any scanned maps or add any multimedia to them.

Chapter 4 - Implementation

In the previous chapter, the architecture of this project was defined. Here we will cover in detail all the steps and phases of development and implementation of the Framework.

As a first step towards achieving a design that was both practical and easy to use, some mockups were designed on paper for the different menus and layouts of the Framework. There is an emphasis on the importance of this initial process, as it is essential to test a variety of visual scenarios to visualize how all the components and elements of the system are grouped and related.

The following figure (Figure 7) shows the result of some of these initial mockups that were later used as design models to follow in the Framework implementation.

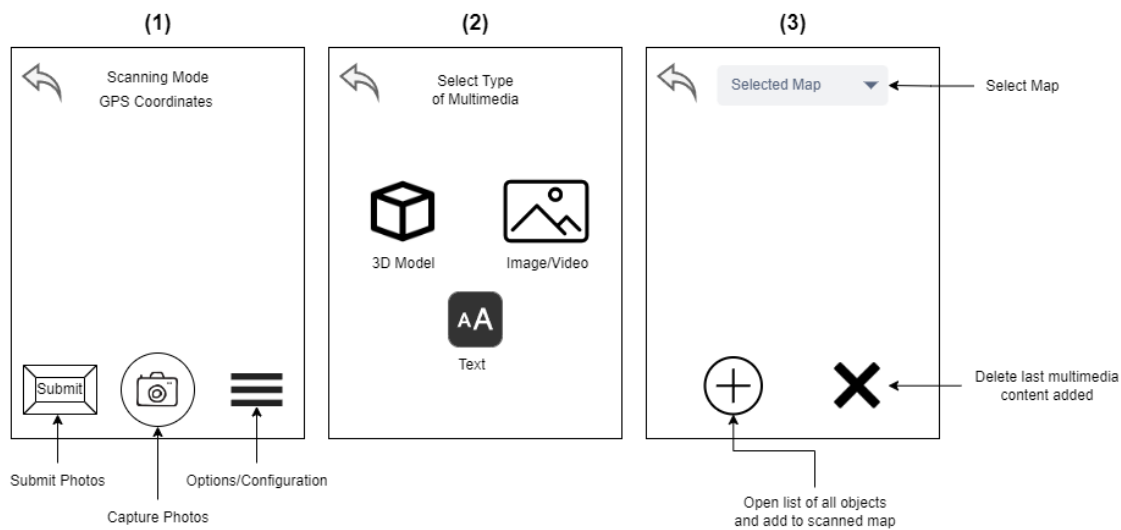


Figure 7 - Framework mockups

Above is a sketch of the design and layout of four of the various menus and boards present in the final product. There is a main menu, on the top-left corner of the figure, where the creator can initiate one of the different functionalities of the application, and a display of three of the tools offered by the framework: reality scanning (1), multimedia creation (2) and multimedia addition to the scanned map (3).

In addition to helping to visualize the system design, these mockups also allowed for a new analysis of the possible need to add new sub-functionalities

or adjust the previously proposed features. An example of this was the addition of an options menu in reality scanning mode, which makes it possible to adjust the size of the points (from the point-cloud demo of mapped maps), to list only maps near the current location and to change visibility of these points.

The following topics include a detailed analysis of how the implementation of each of the main objectives was conceived and achieved, outlining the main challenges encountered, their respective solutions, some images of their development and final results.

4.1 - Digitization of Reality

Digitizing a piece of the real world requires transforming and converting that piece of reality into an augmented element. This new augmented element must be as similar to reality as possible so that, when using the Framework, the camera of the user's mobile device can recognize that a certain area or object in the world we know corresponds to something previously scanned.

Visual scanning is associated with the GPS coordinates of the real-world points so that a mobile device is not only able to perceive the location of these points through the use of the camera, but also discern how close it is to them, and even observe if there are (and what are) other points scanned/mapped nearby.

This step is therefore extremely important as it serves as the basis for the entire Framework. Once an area of interest has been scanned, it can be found by the mobile device camera and allows the creator to associate any other type of multimedia with it.

Figure 8 shows the initial interaction of a creator with the framework to scan a point of interest, in this case, a painting hanging on a wall.



Figure 8 - Interaction between creator and framework for object mapping

One of the main objectives of the Framework is to allow the scanning of various points of interest, as it is crucial to let the creators define any area or a real object to, in a final phase, create a storyline by defining a route between them to be navigated and experienced by an end user.

Considering that different types of creators may use this Framework to varying degrees of experience, in this initial process the practicality of using it was also taken into account.

Summarily, we use Immersal's powerful mapping capabilities to implement this since this functionality is available within Immersal.

When initializing this component, the Framework automatically turns the camera of the creator's mobile device on allowing him to perform a series of actions from there:

- Visualize the environment exactly as is;
- Capture photos of the areas and/or objects you want to map;

The construction of these scanned maps works by finding key points in different images, that is, its algorithm actively looks for similarities between the various captured images, through various angles and perspectives, thus then making an estimate for the creation of a 3D structure, by triangulating these different key points.

According to the documentation of the Immersal SDK, it is important to cover as much of the environment that surrounds our point of interest and preferably at as many different distances as possible. It is imperative to have about 50% of overlap between the last captured photo and the next one for the system to be able to recognize these key points. As a way of getting feedback, the mapping tool generates a pop-up notification that indicates whether enough key points were found between the last two photographs or not.

In Figure 9, an illustration can be found of four (4) photographs exemplifying how they are connected to each other:

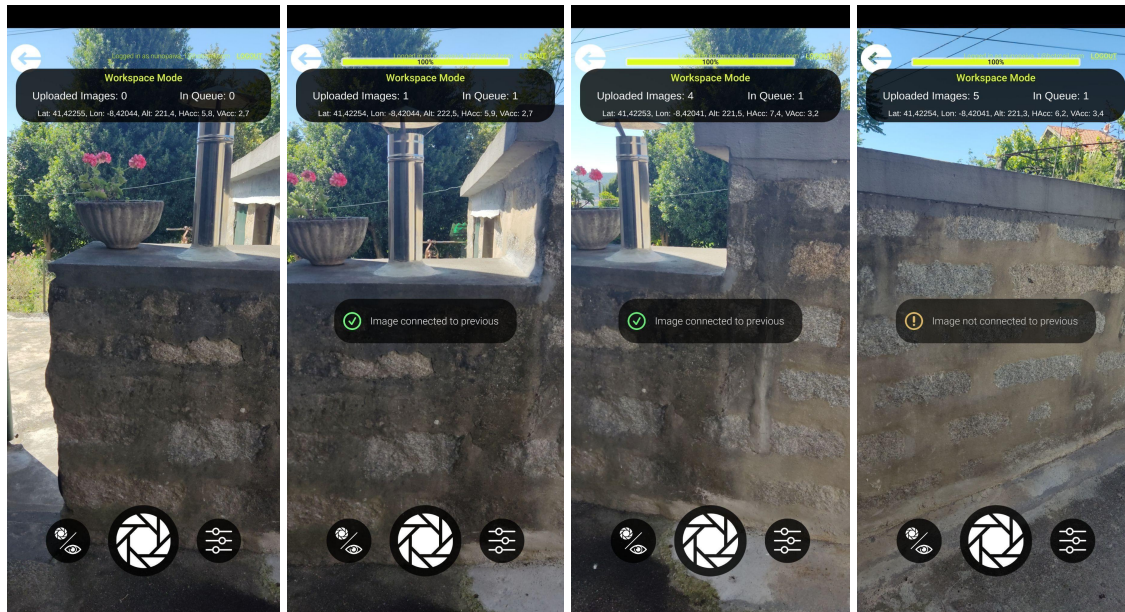


Figure 9 - Connection between captured photos

To sum up, to build a map of the highest quality possible, the captured images must cover the same areas across various angles and distances.

When locating a map, it must contain enough information corresponding to the perspective in which we want to find it. In other words, the more images that are used to build the map from the perspective, angle, and distance we are from the map when we want to locate it, the better its accuracy and responsiveness will be.

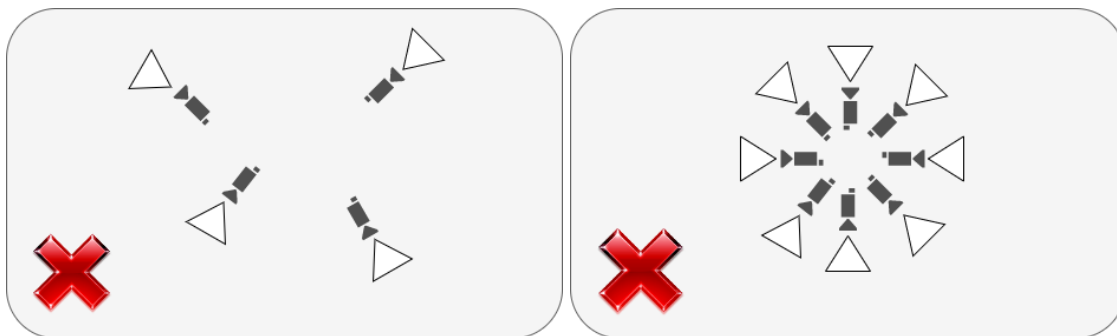


Figure 10 - Incorrect approach to mapping

In the figure above (Figure 10), on the left side, we see that there is no overlap between the images obtained since they were captured from different angles and perspectives. On the right, there is also no overlap as all images were captured from a single point of view, with no diversity of angles. Key points will still be found between the different images, however, the mapping may result in

an error or it may even be generated successfully, however leaving us with an inaccurate and irregular construction.

To fully optimize these results, it is therefore advisable to try to obtain the maximum perspectives and angles of the object or area to be mapped, in a panorama style as seen in the figure below (Figure 11).

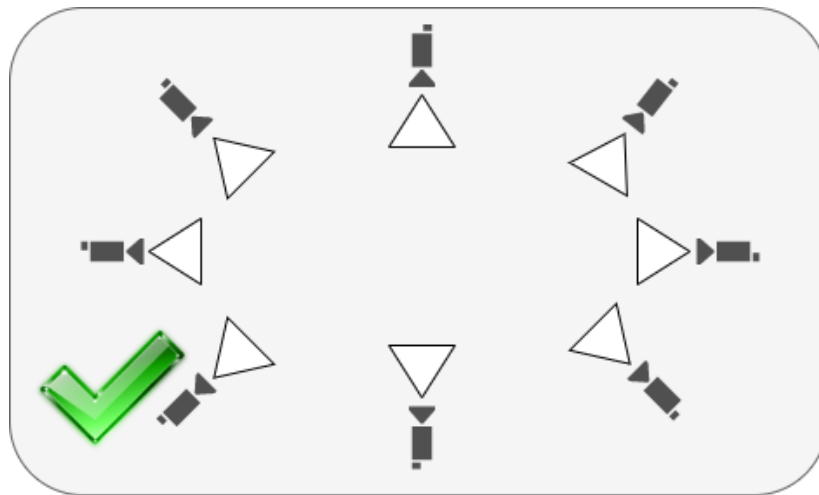


Figure 11 - Optimal approach to mapping

Outdoor Mapping

When mapping an outdoor environment, it is not necessary to digitize all that surrounds the area or object that we intend to map, only the point of interest. A great example of a point of interest is a statue. In this case, there is usually space for us to walk around it, allowing us to take several photos from different perspectives, always with the point of interest as the central focus, as seen in Figure 12.

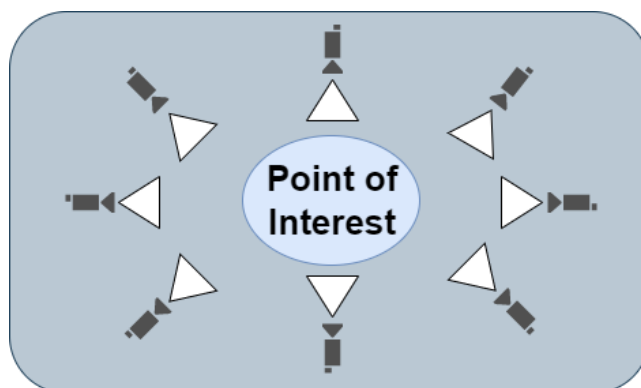


Figure 12 - Mapping an outdoor point of interest

However, if the objective is to map a point of interest that is the entrance of a house/commercial space, a mural on a wall, or any other point that does not allow the creator to walk around the object or capture photographs from all angles possible, they should try to cover all perspectives, creating as many overlaps as possible in this particular sequence of images so that they obtain the most detailed mapping possible. It is even advisable to get some close-ups of more detailed areas so that the map construction contains them in high detail.

In short, photographs taken in close range of the point of interest will contribute to the quality of the detail of the map constructed and photographs with higher distances will help the Immersal mapping algorithm to locate all these details allowing for a greater precision in the construction of the map. This ultimately results in a final generated map that appears to be realistically connected. In the figure that follows (Figure 13), we see the most recommended mapping technique of a point of interest that does not allow the capture of photographs around.

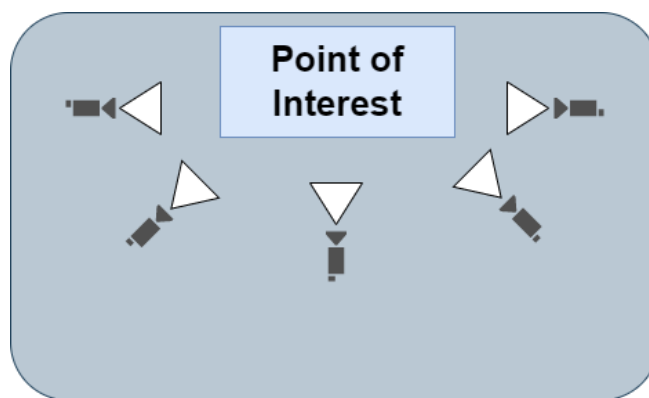


Figure 13 - Mapping outdoor entrance, wall, etc...

Indoor Mapping

The first thing to be taken into account when mapping an interior space is the size of the space we intend to digitize. Mapping a smaller space is possible by simply taking photographs in landscape mode, since a larger horizontal area is able to be captured. This allows for a quick construction of the augmented environment that can be complemented with various close-ups of important details in order to increase the quality of the detail in the generated map, as was mentioned above.

In general, especially for larger spaces, the use of the “outside-in” method is recommended, either in portrait or landscape mode. This method consists of capturing images whilst moving around the perimeter of the space, always looking towards the center of the room, as demonstrated in Figure 13 above. Although possible to map any interior space in other ways, it is through this

method that we will obtain the most accurate results, as the algorithm is able to recognize the space more easily and quickly associate all the objects that comprise it.

Combining this "outside-in" method with the alignment of several maps allows us to build the digital mapping of an entire floor. We can see how it is possible to map the floor of a three-room building in the figure below (Figure 14).

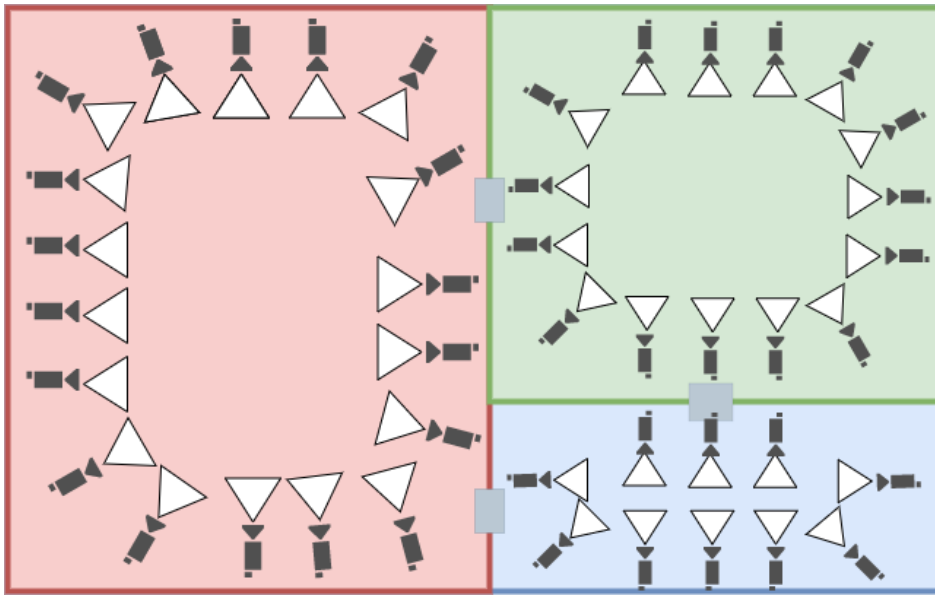


Figure 14 - Combining "Outside-in" method with map alignment

Once the creator has gathered all the necessary photos to map the three rooms shown above, only two more steps need to be completed:

- Capture an anchor photo
- Align the three maps

Ideally, the anchor photography should be the final image to be captured immediately before submitting all images for map construction.

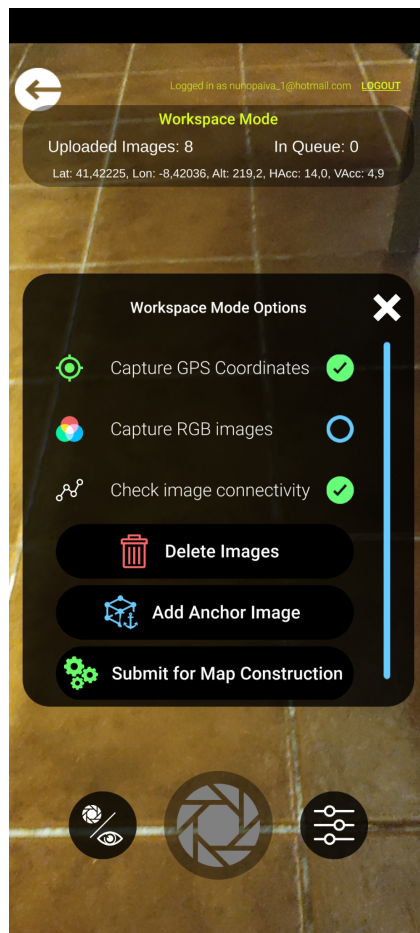


Figure 15 - Framework menu for Anchor Image addition

According to the Immersal SDK algorithm and guidelines, the Y-axis of the map might be correctly pointing up, but rotation around the Y-axis is not constrained and can change between different constructions if you remap the same space. To assist in this process, the mobile device's GPS component and compass are also used, but they alone do not guarantee excellent results. Because of this, in the absence of the anchor image, it would be extremely difficult to add and position multimedia content within the generated maps.

Aligning the scanned maps within the same building or along the same path is the last step for this new augmented world creation. Immersal SDK's map alignment functionality is used here, comparing, associating, and connecting the different selected maps using their GPS coordinates, in order to create a complex mapping of an entire floor that is both accurate and efficient.

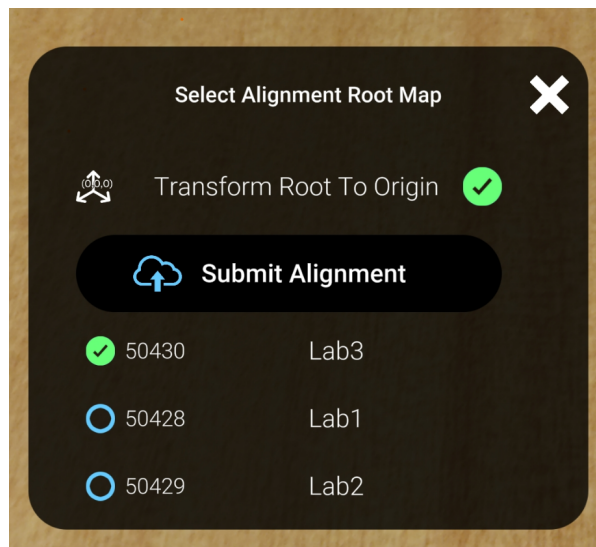


Figure 16 - Framework menu for Map Alignment generation

Once the entire process of photographing the point of interest is completed, the images are then sent for construction of a 3D map. During this build process, it is possible to track its status directly from within the Framework until it is completed and ready to use or alternatively the creator is able to check it on their personal Immersal Developer Portal page.

It is in the Web Developer Portal that all maps and alignments are generated and saved to be loaded later from the Framework. In the Developer Portal, the creator has access to a Dashboard that allows to easily manage all the scanned maps, allowing the creator to quickly see information about them, delete, and even search or share public content with other creators.

The screenshot shows the Immersal Developer Portal dashboard. The top navigation bar is yellow and contains the text "IMMERSAL Developer Portal" and links for "Reset password", "Reset token", and "Logout". The main content area is divided into two sections. The left section, titled "SDK Download & Documentation", contains links to "Download Immersal SDK Core (unitypackage)", "Samples on GitHub", "Demo App", and "Jobs". The right section, titled "Private developer token", displays a token and a "Copy token" button. Below these sections is a "Jobs" section with a "View my maps" button. The main content area features a table of maps with the following columns: Map ID, Name, Status, Images, Sharing, Latitude, Longitude, SDK, Creator ID, Created, and Downloadable files. The table lists three maps:

Map ID	Name	Status	Images	Sharing	Latitude	Longitude	SDK	Creator ID	Created	Downloadable files
52301	ExtSegundo	done	29	private	41.42227	-8.42037	1.16.1	4363	2022-05-26 10:34:03	ExtSegundo.bytes ExtSegundo-scenes.aly ExtSegundo-scenes.aly ExtSegundo-text.aly ExtSegundo-metadata.json
52300	ExtPorta	done	21	private	41.42236	-8.42036	1.16.1	4363	2022-05-26 10:32:55	ExtPorta.bytes ExtPorta-scenes.aly ExtPorta-scenes.aly ExtPorta-text.aly ExtPorta-metadata.json
50430	Lab3	done	43	private	41.69476	-8.84696	1.16.1	4363	2022-04-22 15:38:48	Lab3.bytes

Figure 17 - Immersal Developer Portal

As we can see in the figure above (Figure 17), this dashboard presents us with a table that contains all the information about our maps. As we can see in the next figure (Figure 18), there is a distinct difference between the different “Downloadable Files” that it presents to us.

☐	50430	Lab3	done	43	private	41.69476	-8.84696	1.16.1	4363	2022-04-22 15:38:48	Lab3.bytes Lab3-sparse.ply Lab3-dense.ply Lab3-tex.glb Lab3-metadata.json
☐	50429	Lab2	done	52	private	41.69473	-8.84700	1.16.1	4363	2022-04-22 15:36:53	Lab2.bytes Lab2-sparse.ply Lab2-dense.ply Lab2-tex.glb Lab2-metadata.json
☐	50428	Lab1	done	31	private	41.69479	-8.84689	1.16.1	4363	2022-04-22 15:33:13	Lab1.bytes Lab1-sparse.ply Lab1-dense.ply Lab1-tex.glb Lab1-metadata.json

Figure 18 - Information and different types of generated map files

Thus, there are five different types of files generated, each of which plays a critical role in the development of this Framework:

- .bytes and metadata.json: these contain all the individual information and geographic coordinates of the constructed map, making it possible to show it in the context of the space that it represents and to align it with other maps.
- .ply: this is a file type known as Polygon File Format and generates the point-cloud mapping of this construction. A superimposed version of it is then used to give feedback to the creator about the quality of the mapping, allowing to see whether it is well-oriented and superimposed on reality. With these points and polygons, a second .ply map is created that is the first recreation of reality.
- .glb: a 3D format file type that is often used in Augmented Reality, Virtual Reality, games and even web applications as it supports motion and animation, takes up very little space (small file) and has a very fast loading time. In this file, we have the best view of our mapping, having already inserted textures over the previously created polygons.

Below is a Figure that compares one of the .ply files side by side with the .glb file:



Figure 19 - Map construction from point-cloud mapping

As we can see in Figure 19, on the right side, the constructed map is very similar to the reality that is reconstructed from the information and point-cloud mapping in the image on the left.

By integrating all the components mentioned above, the Framework allows the user to perform all these actions in real-time using only their mobile device. After capturing all the necessary images, the user takes one last photograph, the anchor photo, and sends the map for construction. The Framework also verifies, in real-time, the status of this process until it is completed. Once finished, you can load the scanned map and place it under the real world to see how accurate and similar it looks, and finally align it with other maps or eliminate any scanned maps that were not built to the required quality.

Finally, sharing maps between creators is possible through the Developer Portal. That is, scanned maps can be defined as public or private so that anyone can access them or only those who have access to the creator's token can view them.

4.2 - Multimedia Content Creation

There are several types of multimedia elements available from our mobile devices: text, images, videos, 3D objects, and even 3D animations. However, in the context of this project, we are working with all these media files in an

augmented reality environment. That is, in addition to being able to visualize the multimedia objects under the layer of reality, we are also able to interact with them.

In order to create infinite augmented worlds, there are two ways to get the perfect multimedia content to be added to each augmented world:

- Pre-set multimedia content
- Multimedia content created and added by the creator

Regarding the pre-set content of the Framework, these are mostly 3D models and 3D animations. These are provided together with the Framework since the process of creating and adding them in real-time would be very complex and time-consuming. Therefore, the creator has at his disposal a series of useful and universal objects, such as stones, trees, tables, etc... that allow them to customize worlds and points of interest in detail, making them more aesthetically pleasing to the eyes of an end user. The 3D content in question was modeled when creating and developing the application in the Autodesk Maya program.

However, it is also necessary to leave some space for the addition and insertion of fully customizable content such as text, images, and videos. Using this Framework component, it is possible to create texts, with one or more lines, to add pieces of information at a place of choosing near the respective points of interest, as well as images and/or videos the user deems pertinent.

In the end, you are left with a multimedia set of various types of mixed multimedia that give a completely new and unique look to a new augmented world.

While using the Framework, the creator has access to this functionality through the “Create” option in the main menu. They will then find a sub-menu in which it is possible to choose between creating a new text or adding a photo/video. In Figure 20, we may verify how this user can create and add new content to the framework:

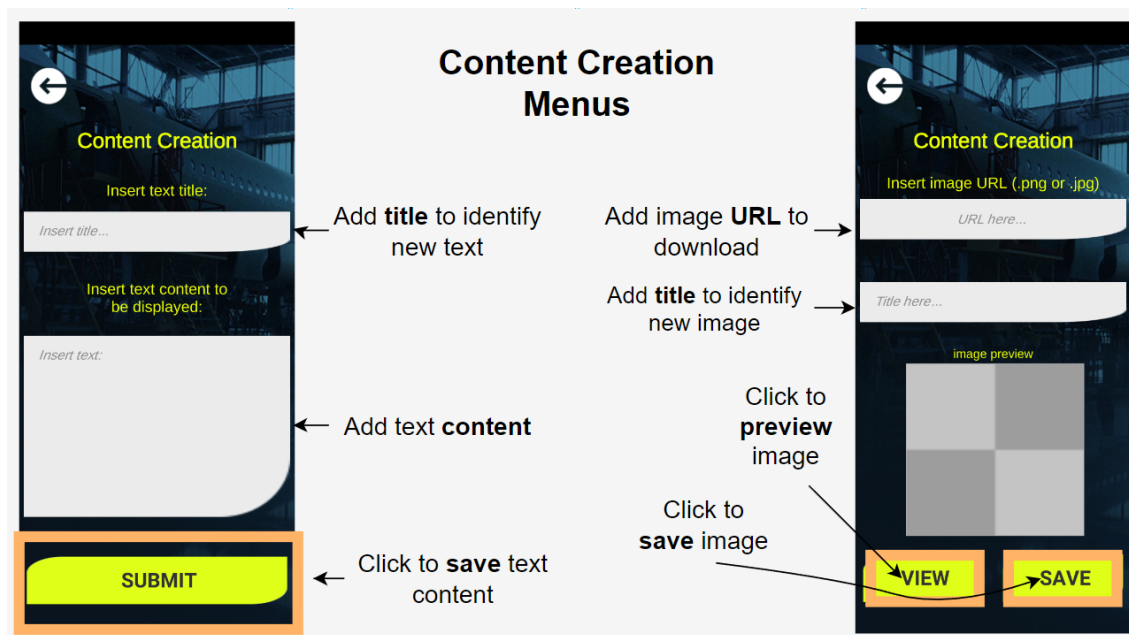


Figure 20 - Framework content creation menus

In order for the creator may create a new text, he must add a title, which will be used as a unique identifier for this new element and then be saved and loaded when necessary, as well as the content/description of that element, and ultimately be the visible text for end users.

Adding a new image to the Framework requires the insertion of a URL so that the system can find and download that image online. The URL should be followed by an identifying title, similarly to adding a new text. There is also a “View” button available to allow the visualization of that image. This makes it easier to verify its integrity before saving it, thus creating new multimedia content that can be added to any previously prepared map or point of interest.

For this content to be generated in real-time, prefabs such as 3D Text (for texts) and 3D Plane (for images and videos) were prepared and integrated into Unity, which serves as placeholders for the actual content to be loaded with the correct information and originally defined position.

Thus, the next step is to save the information obtained through the creator input in the content creation menus presented above, using the respective placeholder objects. All this information is stored in .json files for texts but not for images as these types of multimedia require different treatments. For texts, all that is needed is a simple structure capable of storing the title and content of each element. However, since images are retrieved from a URL entered by the creator, they must first be converted to bytes[].

Since it is not possible to save an image or video obtained from an online URL in text format or a .json structure, an image conversion algorithm was implemented. The algorithm receives the online multimedia content in Unity's WWW type and saves it in a variable of type bytes[] so that it is possible to store it in a light and fast-reading file. This way it is also possible to quickly associate the content of this file, of type .bytes, to a Unity Texture, allowing us to add that texture onto a 3D object, for example, a Plane.

In regards to the 3D models and animations that exist by default in the Framework, those are attached to the system through Unity's "Resources" folder, which allows to easily instantiate any of them in the desired AR space. Currently, it is not possible to customize or create/import new 3D models into the system in run-time as this process would be highly complex both to implement as a feature and to be used by a creator. When importing a new model to Unity, with or without animations, some manual adjustments and configurations are always required to set everything up correctly.

All .json structures responsible for storing information related to multimedia content are directly stored in an online database, using the MongoDB Atlas service. This allows a quick interaction between Framework and the content available online, allowing its use and management by any creator or end user.

4.3 - Multimedia in Relation to the Digital World

The third step in the implementation process of this Framework is the relationship between multimedia elements and the new augmented worlds. In other words, it involves allowing a creator to add and associate augmented content, created by themselves or made available by the system, to all their worlds. We can divide the development into four parts:

1. Importing multimedia content into the augmented world
2. Association of one or multiple multimedia contents to an augmented world
3. Interaction between creator/user and multimedia content in runtime
4. Storage and loading of each type of content (multimedia and scanned maps) within the respective augmented world

To associate augmented content with a scanned map, it is first necessary to define which scanned map we are working on. This is because each scanned map will have a particular and unique set of contents defined by a creator. For this reason, the system was developed to allow the creator to choose which map they want to edit, right at the beginning of the editing process, as we can see in Figure 21.

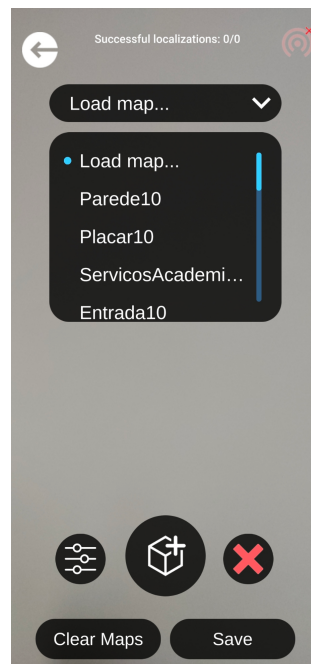


Figure 21 - Framework "Map Download" feature

After clicking on the scanned map you want to work with, the system loads it directly from the Immersal Developer Portal and starts a search process, activating the map localizer algorithm using the camera. This will compare the map obtained by the construction previously made by capturing images for digitization of reality with all the camera input from that moment on.

At the top of the screen, there is an informative text that aims to show the creator how many attempts were made to locate the desired map and how many of these had a positive result. On the right side of this text, there is also an icon to indicate the quality of that detection/localization. This icon has three distinct states: high quality, medium quality, and failure detection. In high-quality detection, we know that in that direction, angle, and perspective, the system can perfectly detect the desired map. If the detection is unstable, on the other hand, it is recognized as the medium quality, which serves as a warning to the creator. This allows the creator to know that at that particular angle, distance or positioning there may be some instability in the detection of the point of interest.

Then it comes to adding a multimedia element to the new augmented world, the first step involves choosing the type of multimedia that will be added to the environment. The creator has three options: a 3D model, text, or image/video. In Figure 22, we see the illustration of the choosing process:

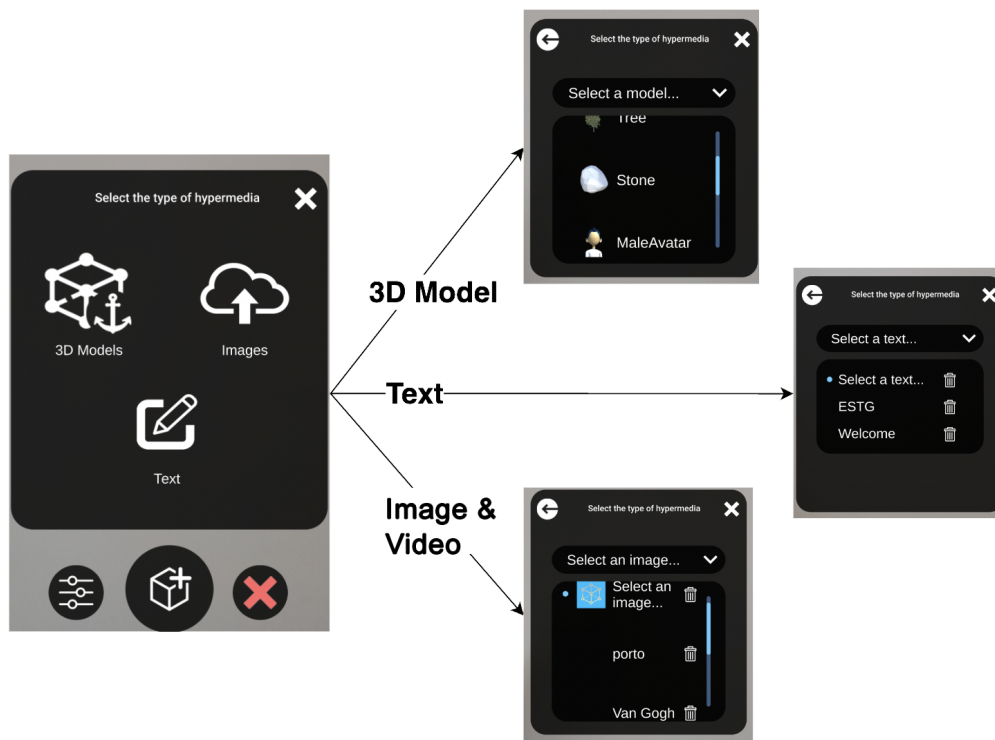


Figure 22 - Multimedia selection screen

Each of these types of multimedia has its list of options, from which the creator can choose which one to add to the world.

The 3D elements, as they already exist by default in the application, are instantiated directly from their prefab, about one meter away from the camera, with their predefined rotation and scale. These can be animated or not, and contain, like all other multimedia content, scripts to prepare direct interaction between creators and colliders so that the collision detection function is enabled.

3D text elements, however, are added in a slightly different way. In the process of adding 3D text to a scene and world, the system instantiates a 3D text object in the scene, also about one meter from the mobile device's camera. Using the information provided by the respective .json file, this will incorporate the description and content associated with the title of the text chosen when this element is added.

Finally, there is the addition of images and videos. After selecting the desired image or video, it will be incorporated into that respective active point of interest in a similar way to the other types of multimedia files discussed above. After selection, the system instantiates a prefab of a previously prepared 3D element (plane) immediately converting its texture to the texture corresponding to the image, which is converted into byte format and stored when added to the

Framework. It was therefore necessary to add a re-conversion algorithm to transform the byte content to Unity's Texture2D.

Regarding the association between the augmented world and the multimedia content, it is important to mention the importance of two game objects used by the Immersal SDK. These are used to transform all the augmented content so that they correspond to points, both in the real world and in the new augmented worlds, thus relating both. These game objects are AR Space and AR Map.

The AR Space game object contains a set of AR Maps, which are all the active maps in a scene, that is, all points of interest to be searched by the Framework at any given time. When a space is detected by a creator's camera with the help of a scanned map (via Immersal's ARLocalizer.cs script), the AR Space object is transformed so that all augmented content visually corresponds to the previously mapped real world. Therefore, this AR Space game object is a parent in a hierarchy of various objects of all maps and their active content in a given scene, therefore being able to manage and transform all the active augmented content. Each multimedia content that is added to the scene is associated with a map/point of interest, position, rotation, and scale relative to the associated map. This way, a high-precision association between the map and content is achieved.

All types of multimedia allow direct interaction between them and the creator after being added to the augmented world. As discussed above, each of these contains a script that detects the click under the 3D collider of each respective object, allowing the creator to move these contents within their new augmented world. The figure below (Figure 23) shows a quick exchange of positions between two multimedia elements through direct interaction between the creator and augmented content.

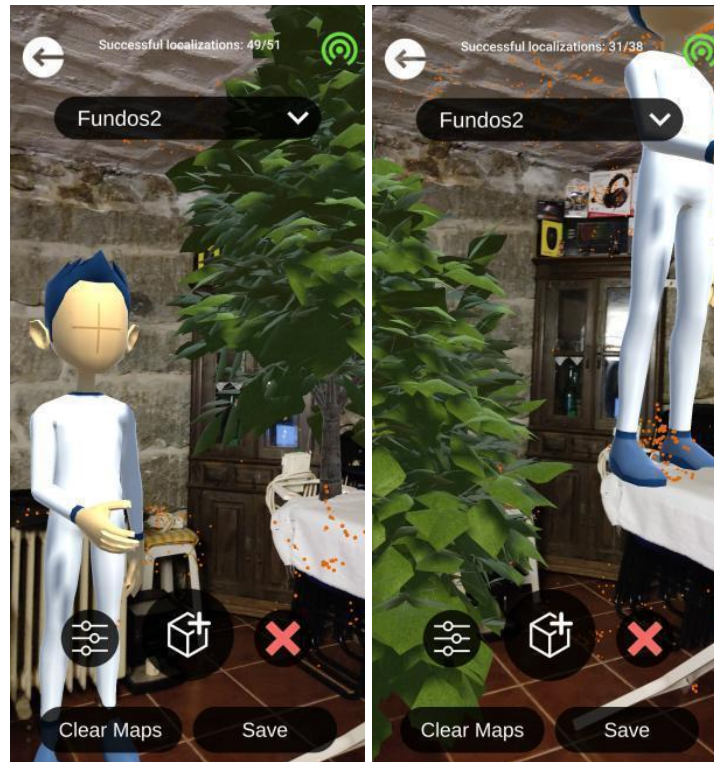


Figure 23 - Interaction with the multimedia present on the augmented world

After composing and preparing all the multimedia content of a certain scanned map, it is necessary to allow the creation storage so that, later, it is possible to load and show all its content. Both are used for editing the scanned map from the creator perspective user and for viewing by an end user.

At this stage, after selecting the "Save" button visible in Figure 23, a new .json file is created (or edited, if it already exists) with the name of the map ID under which the creator is working. The structure of this file allows for saving any type of object since its first parameters are exactly the type of content and its name. It follows its position, rotation, and scale concerning the scanned map. All contents present in the active scene are added to an object list that stores all this information, ensuring that no two objects are exactly the same, when converted to .json format to be saved and loaded later.

This loading is done whenever a previously edited map is opened both in a creator's edit mode and in an end user's view mode. After selecting a scanned map of interest for visualization, a script called LoadContents.cs starts a search for a .json file named after the ID of the selected map, traversing all the existing information and previously saved contents. Each of the found contents is then instantiated in the scene along with the scanned map, respecting its original type, positioning, rotation, and scale.

4.4 - User Navigation within Digital Worlds

To integrate all the features mentioned above, the implementation of user navigation within an augmented world is a highly important step. This final stage of development will allow the application of the framework in a real context, for example, as assistance or support during a tourist tour, both indoor and outdoor. The main purpose of this last feature is, therefore, to allow a creator to create and share a story by navigating through various points of interest. In other words, in the case of a tourist guide, the goal is to be able to prepare a narrative and allow any end user to dive in and explore the history of any place or area of interest at their own pace while navigating both in an augmented and real world.

In this step, a new augmented world is initially created, consisting of a series of maps or points of interest previously mapped and constructed, as we can see in the figure below (Figure 24).

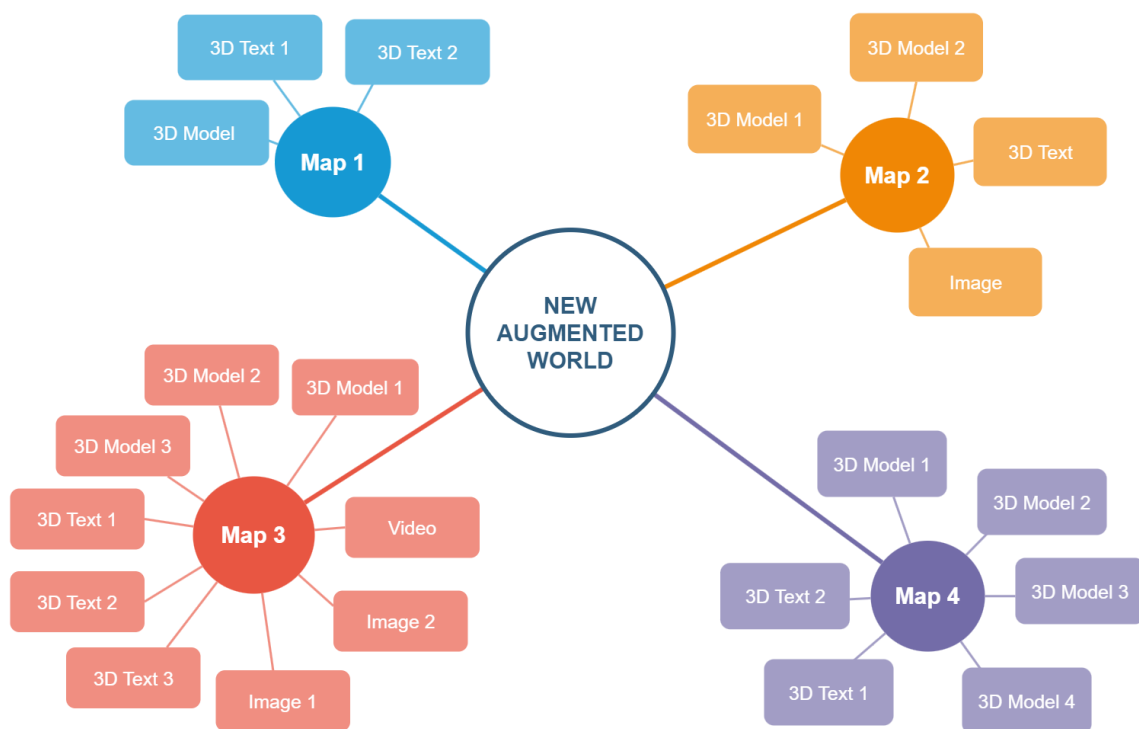


Figure 24 - New augmented world, maps and respective multimedia relation

Therefore, in this initial step, the creator decides which maps will constitute their new augmented world, delineating their story.

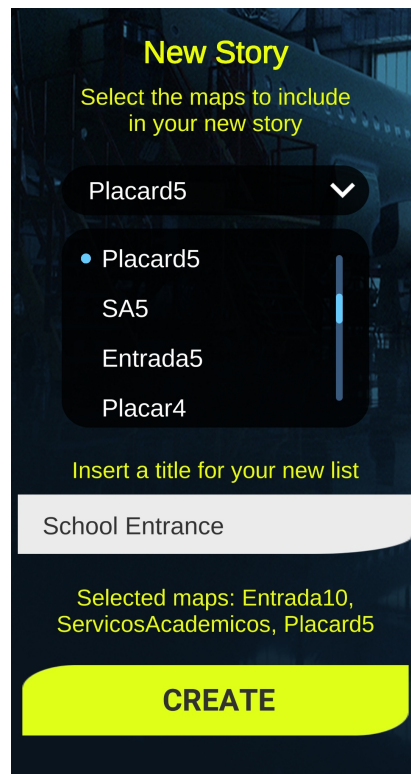


Figure 25 - Framework's new storyline creation feature

Figure 25 shows the scanned map selection screen for creating a new storyline. It allows for the choice of all scanned maps available in the Framework, showing, at the bottom of the layout, the list of all selected maps so that the creator can confirm the selection before creating the new augmented world.

The creation of this augmented world leads to the generation of a new .json file, with the name of that AW, which contains the information about all the maps that make up this story. The result is a .json structure that stores all the maps in a single file, as shown in Figure 26.

```

{
  "Map_List": [
    {
      "id": 50243,
      "type": 0,
      "version": "1.16.1",
      "creator": 4363,
      "size": 40,
      "status": "done",
      "privacy": 0,
      "name": "Fundos1",
      "latitude": 41.4223752295,
      "longitude": -8.420373646249998,
      "altitude": 209.60657196044924,
      "created": "2022-04-19 15:46:46",
      "modified": "2022-04-19 15:49:23",
      "sha256_al": "9812df00c2ec1fc93fe40a9deb67e5365485f45ef99ccd81ab455b24cebb5e6c",
      "sha256_sparse": "4aba1ea719f6f8529bf3f75da77607e1682a342c1753c898cc9f561d7bab4108",
      "sha256_dense": "570479d08ab3018b15986096dcd0ff084ef9063219854660ef1aa196828ee7ce",
      "sha256_tex": "57a0fee018b21eee17bf5e617c72d4954393c50a662a71bbf2166c2b572a13e4"
    },
    {
      "id": 50244,
      "type": 0,
      "version": "1.16.1",
      "creator": 4363,
      "size": 32,
      "status": "done",
      "privacy": 0,
      "name": "Fundos2",
      "latitude": 41.42226831562501,
      "longitude": -8.420367174999999,
      "altitude": 204.46742630004884,
      "created": "2022-04-19 15:47:51",
      "modified": "2022-04-19 15:49:23",
      "sha256_al": "5949914797c33bad7d602fbace03f9a2b465c4e78c87a7ea8b620e9772d9dde3",
      "sha256_sparse": "6bfb53a1c879aef69f83b9021ccd0ea65243acfc066aaf75f7cdc3d5d0147f1",
      "sha256_dense": "c4154f326cb5a12ae5a4fa4f8f640bc1895a1c985e77b23600acb920d3e95630",
      "sha256_tex": "943935e84e7134b3409b30b5bd3ff3e869ae967c625379a8ca73832eef4b3eb8"
    }
  ]
}

```

Figure 26 - JSON structure for new augmented world

For each scanned map, its ID, dimensions, original creator, GPS coordinates, and a reference to all the files resulting from its creation, among others, are stored. The ID serves as the map identifier, the creator defines who can view this content, and the size and other files ensure the correct import and location of maps in the real world.

Once a new augmented world is created, the next step is to allow the creator to be able to work on it, to prepare a path that shapes its history. Therefore, we move on to loading this augmented world for visualization and configuration. To load an augmented world it is necessary to repeat the map import process discussed in the previous sub-chapter (4.3), but this time to extend this loading from an individual map to multiple ones. Therefore, a cycle is carried out that goes through and loads all maps present in the .json file of an augmented world. One by one, all the scanned maps that are part of this augmented world are imported and the system starts trying to locate all of them immediately after. This time, the localizer algorithm works a little differently than before, when localizing individual maps. In this story view and editing mode, once a point of interest is found, all the other maps that were previously aligned will be pre-placed under their predefined positions even before the creator or end user get close to their location. This method allows the next step to be carried out more accurately and consistently, as the Framework will actively try to locate

and detect each of the maps in two different ways: through its GPS coordinates and the localizer algorithm.

The third step, after loading an augmented world, is to allow the creator to define the route he wants all end users to take. To do this, two main game objects were created and used: waypoints and targets.

A waypoint is a point where the route will pass, that is, it will help to build and define the general route. It includes a script to manage its behavior in an augmented reality context and a list of neighboring waypoints. The listing of neighboring points is essential for this functionality to work, as it will determine the direction and order that the route will take until it finally reaches the defined destination.

A destination (target) is similar to a waypoint, however, both its 3D model and its purpose are different. Its purpose is to define the end of a route and it is automatically linked to the last waypoint added to the scene.

In the editing mode of an augmented world, the creator can walk along his route in the real world and, at the same time, add waypoints for the points of interest where they want the end users to pass or visit in the particular story they are creating. In this way, it is possible to verify the integrity of all the augmented contents added while a new route is being outlined. Right after adding a new waypoint, it is essential to link the last one added and the new one so that the system recognizes the order and direction to be followed.

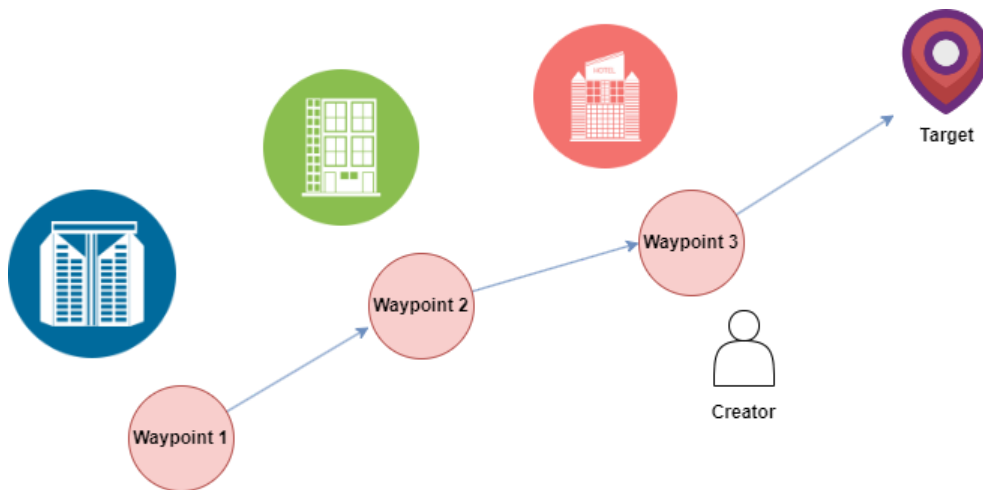


Figure 27 - Waypoints and targets connections

In Figure 27 we may observe a graphic representation of how the routes are defined, connecting, in order, all the waypoints and defining a target that will be automatically linked to the last inserted waypoint. These links between waypoints are made manually by the creator by clicking on the augmented

object of the waypoints and dragging a raycast line from one waypoint to the next, forming a link and thus generating an oriented graph.

In addition to the final visualization of the route, points of interest, and multimedia content, what remains is to save and load a route created by a creator so that he can later resume editing it or so that an end user can view and explore this world. This is a step that requires, once again, data structuring. In this case, all the data and information about the waypoints and targets have access to a previously defined history, when necessary.

A new .json structure was prepared to receive several nodes (waypoints or targets). Storing the node type, the name of neighboring nodes, in order, and their positions relative to their respective augmented world. Thus, when a route is prepared, it can be saved in its .json file which is later used to position and recreate the entire route after instantiating all the necessary game objects (waypoints and targets). Finally, it remains to be ensured that the end-user has at his disposal all the necessary tools for intuitive and efficient navigation, being able to visit each of the defined points without getting lost.

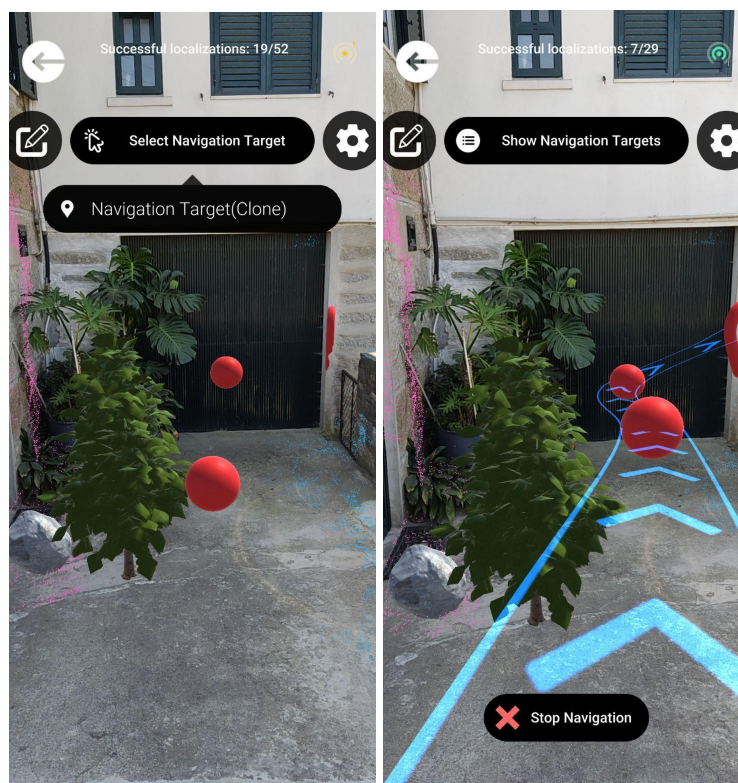


Figure 28 - Storyline visualization screen

As shown in Figure 28, an animated directional line was then added to clearly indicate the route to be taken, following each of the defined waypoints ending in the final target. It is also possible to view all the multimedia content made

available along the route. These will always stay in the same starting position defined by the creator, promoting the accuracy and consistency of this augmented experience.

4.5 - Installation and Use of the Framework

As discussed above, this Framework is divided into two different applications: the creator and the end-user version. To initialize any of these, the users need to install each respective .apk into their mobile device. Creators will then be prompted to enter their Developer Portal credentials to log in, so if it is their first time using the app, they will need to access the Portal and create a new account to obtain this authorization. The login process is necessary to ensure a creator is working under their Token which is then used by an end user to visualize any particular creator's augmented worlds and navigate through them.

4.5.1 - Creator App Screens

A storyboard is shown in the figures below, illustrating how all the creator app screens interrelate:

Map and Align Maps

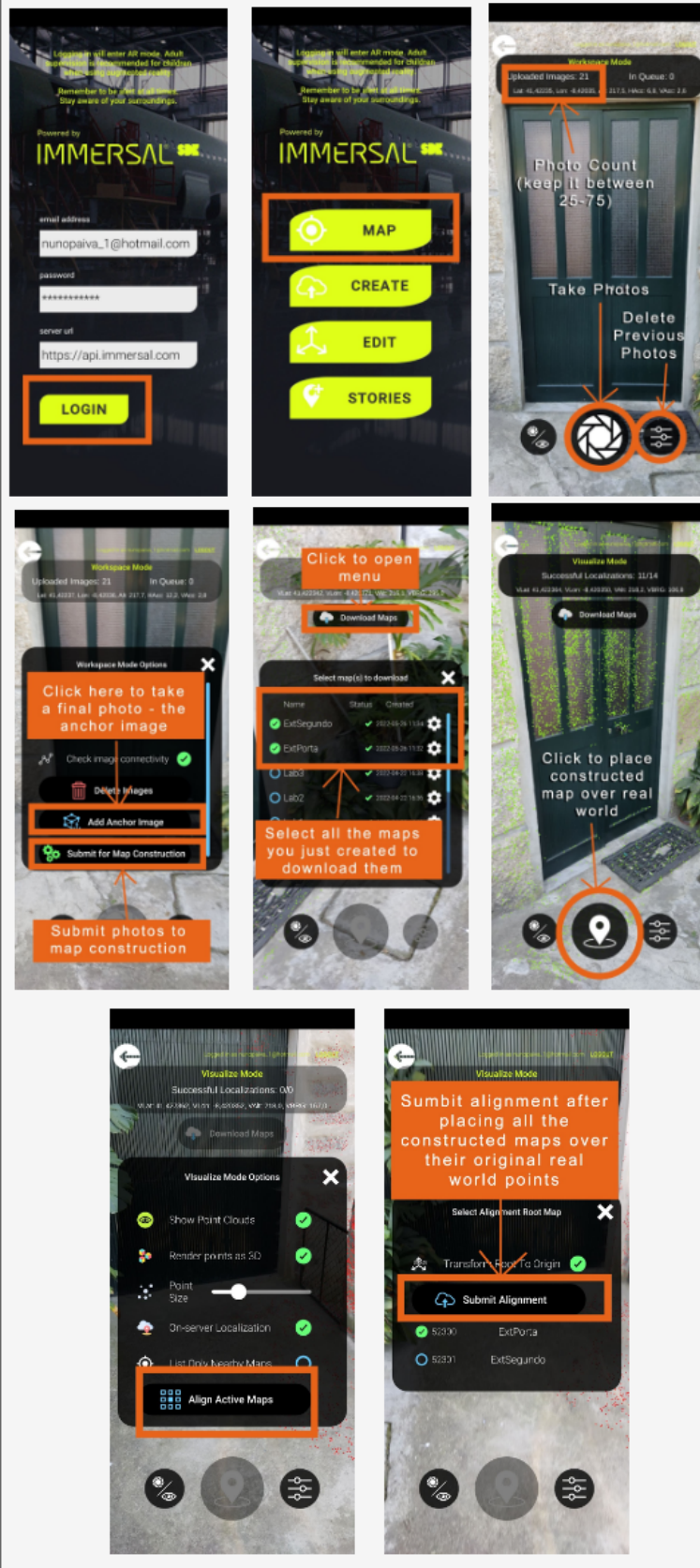


Figure 29 - Scanned Maps Creation and Alignment



Figure 30 - Multimedia Creation

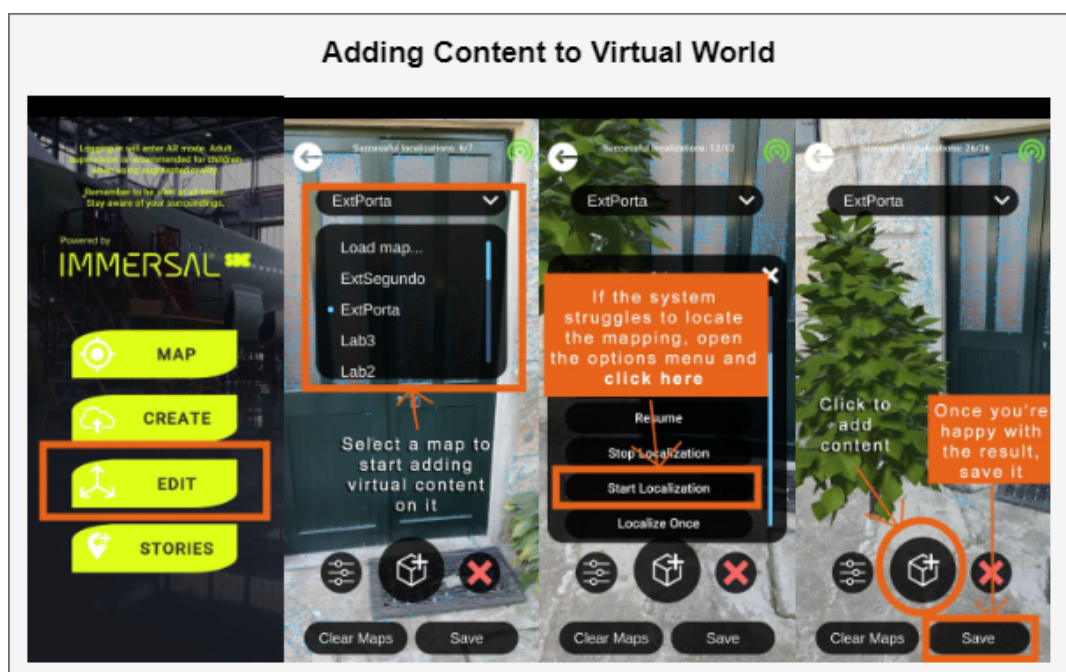


Figure 31 - Adding Content to Augmented World



Figure 32 - Create Augmented World and Storyline

4.5.2 - End User App Screens

Following are the storyboards showing all end user app screens and how they relate:

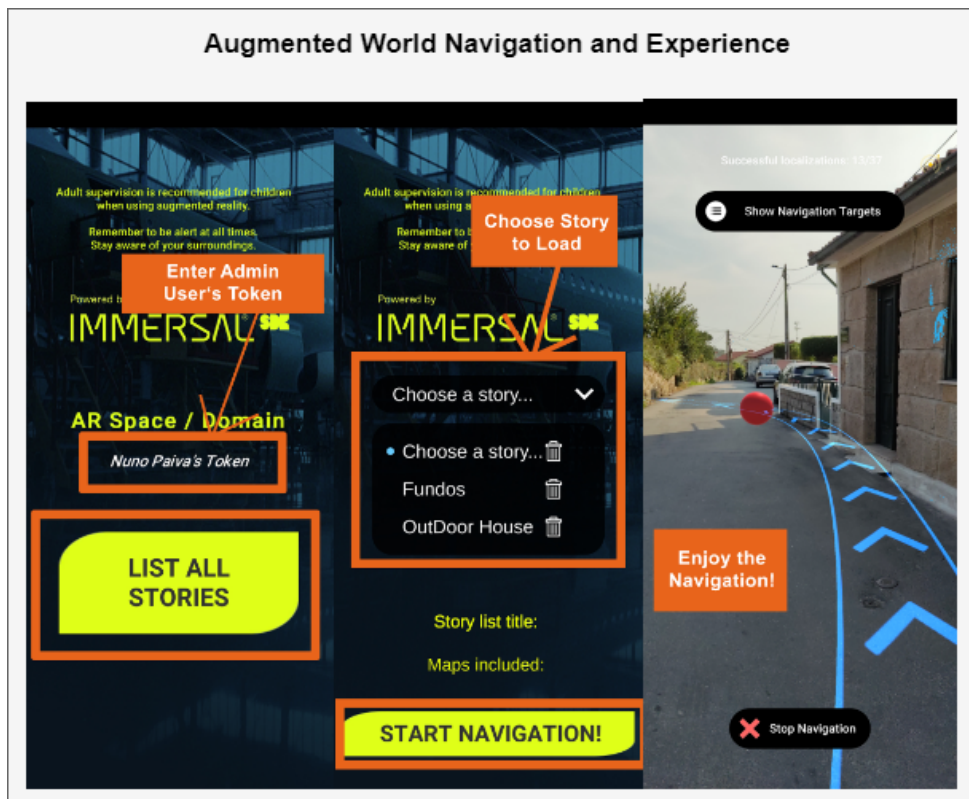


Figure 33 - Augmented World Navigation and Experience

4.6 - User Scenarios

In the real world, this Framework has a lot of different applications in various areas where both a creator and an end user would benefit and improve their daily experiences by using it.

In an indoor environment, this system could be used and integrated by a museum tour guide. Since the framework promotes totally autonomous work and management, the guide would be able to fully scan the interior of a museum, allowing them to place any multimedia content within this constructed indoor augmented world. Thus, preparing a storyline for any museum visitor to go through and experience at their own pace. Multiple routes can be defined within the same world so an infinite amount of experiences could be prepared inside the same museum space. On the other hand, a museum visitor could choose which route to explore and which storyline to experience. For example, a route going through all the art pieces of a particular artist that the visitor is a big fan of. By following this particular route, the visitor would be able to go through all these favorite artist art pieces, discovering extra information and even curiosities about each of them from the AR content previously placed by the museum guide (creator).

Nowadays, there are a lot of people who enjoy walking trails as an outdoor experience. This Framework can also be a great aiding tool for this activity. With the usage of the photo captures and the GPS coordinates of each map, it is possible to prepare a really large outdoor augmented world, setting up a route for any visitor to follow. This addition to the walking trail experience brings two benefits: the visitor can check the Framework on their mobile device from time to time to verify where to go in case the trail gets confusing and the trail guide can also leave some AR multimedia on strategic spots or landmarks within the walking trails to improve the experience.

Figure 34 and Figure 35 illustrate some possible uses of each application in the real world:

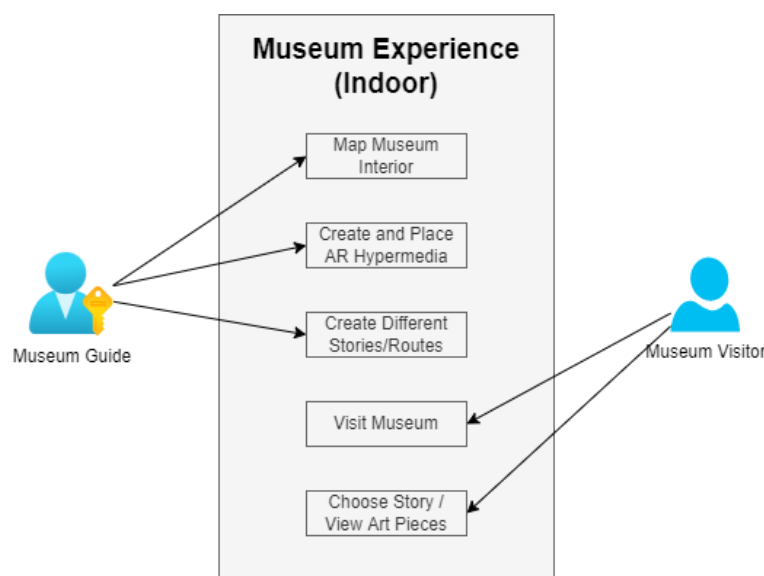


Figure 34 - Indoor Use Case

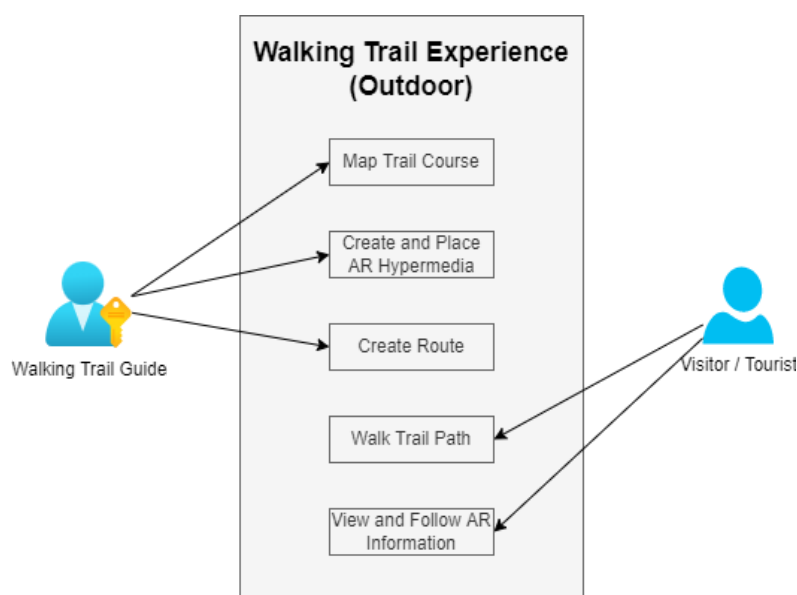


Figure 35 - Outdoor Use Case

There are many other situations where it could be useful to map either an indoor or an outdoor environment, allowing any manager or guide to present newcomers with a fresh and fully autonomous experience such as schools (as an introduction to new students or new professors), hospitals, supermarkets, etc... Even outdoor city events could make use of this system, by preparing a small presentation about the city's history so that residents (and even tourists) can explore using their mobile phones, walking through streets and city landmarks.

Chapter 5 - User Experiences & Tests

To validate the developed Framework, a series of tests were carried out with creators and end users to gather quantitative usability metrics. Specifically, two questionnaires were applied: the System Usability Scale (SUS) questionnaire (Brooke J. , 1996) (Brooke J. , 2013) translated to European Portuguese according to (A. I. Martins, 2015), and a second, more informal one, to gather more specific feedback about the different features present on the system, allowing the test subjects to provide clear observations and ideas for improvements or elements to be added to the system UI and UX.

The main objective of this series of tests was to understand how simple and effective both applications (creator version and end user version) would be in a real-use context. To achieve this, two real stories were developed so that all test users could understand the applicability of this system, thus being able to fully experience this Framework prototype both in an outdoor and indoor environment.

In the outdoor context experience, they followed a story based on the interaction between a tour guide (creator) and a visiting tourist (final user). In this example, the tourist guide aimed to map a tourist spot in the city of *Viana do Castelo*, specifically the *Forte de Santiago da Barra* (Figure 36), thus creating a augmented history about the place so that any tourist can navigate through that world, learning more about this area of interest.



Figure 36 - Mapped points of interest from Forte Santiago da Barra

In the indoor experience, the story was between a finalist student (creator) and a new student (user). In this one, the finalist student's objective was to prepare a story to welcome and accompany any new student at the *Escola Superior de Tecnologia e Gestão (IPVC)* (Figure 37) showing them everything they need to know about the first floor of the school.

Both indoor and outdoor tests were conducted with quick demonstrations of how the applications operate. In this way, both creators and users were able to become familiar with all the features available on the system, allowing them to take full advantage of them.



Figure 37 - Escola Superior de Tecnologia e Gestão (Entrance)

5.1 - SUS Survey

This evaluation was performed with 25 subjects, with 17♂ and 8♀. The average age of users was 24 years old with their ages ranging from 16 to 48. This is a questionnaire with 10 items each one rated with a five-level Likert Scale and options from “Strongly Agree” to “Strongly Disagree”, with a numeric match that ranges from 1 to 5 values.

Item	Average
I think that I would like to use this system frequently	4.16
I found the system unnecessarily complex	2.12
I thought the system was easy to use	4.28
I think that I would need the support of a technical person to be able to use this system	2.16
I found the various functions in this system were well integrated	4.40
I thought there was too much inconsistency in this system	1.64
I would imagine that most people would learn to use this system very quickly	4.24
I found the system very cumbersome to use	1.60
I felt very confident using the system	4.20
I needed to learn a lot of things before I could get going with this system	1.44

Table 4 - SUS Survey results

SUS Score = 80.6

The table above shows all the questions that make up this questionnaire, as well as the average obtained for each one of them. The score obtained in these tests was 80.6, which according to (A. Bangor, 2009) corresponds to a qualitative evaluation of “Good”.

5.2 - Results Analysis

Analyzing the results obtained in the SUS questionnaire, items 2 and 4 stand out, with an average score of 2.12 and 2.16, respectively, which leads to the conclusion that the Framework needs some adjustments in terms of ease of

use. This was, in general, the clearest feedback topic when carrying out the tests, since the system contains a series of different features, so in some of them, namely in mapping and creating stories, users felt some difficulties in using the full potential offered by these components efficiently.

In order to improve this aspect, and so that future users do not feel that they need a technician to be able to use the system in its entirety, some videos, and 3D animations will be developed that serves as a usage tutorial for any new user. When a feature is started for the first time, a demo video will be played to display all the necessary information to the user, which can be repeated later, using a “Help” button.

The remaining criticisms were mainly on the following aspects:

- Improvement on the multiple map selection menu (when creating a new augmented world)
- Addition of more types of interaction with the augmented content (rotation and scaling of multimedia elements)
- Addition of a 2D view map of the created world, showing the creator a top-view map of the entire augmented world, making it easier to verify the integrity and positioning of all the present elements.

In a general sense, after some time of use, users felt more and more comfortable and confident with all the features and menus available, being able to map, create content, add content to maps and even create the desired stories in just a few minutes.

Chapter 6 - Conclusions

The aim of this dissertation was to fill a gap in the available technology regarding Augmented Reality and intended to provide more autonomy to users in the current paradigm.

The objective of this work was to build a framework capable of allowing the user, with any level of experience, to build and interact with a unique and personal AR application. While using this framework, they would be able to easily and autonomously: digitize the real world, create and interact with augmented content and, finally, create their own stories within their infinite augmented worlds.

In order to achieve that, the Immersal SDK was modified and integrated with Unity 3D and MongoDB. This SDK already provides a lot of solutions and well-designed algorithms when it comes to an AR application. However, new structures of data, scripts, and scenes were created and implemented into this system, enabling users to fully manage their AR application from a singular mobile device.

This work devised the creation of two applications: one able to map areas or objects, create new augmented content, add and relate multimedia with augmented worlds and create stories by preparing a route for any other user to navigate (creator version); and the other able to view all the prepared routes/stories, select them and navigate through all the available augmented worlds (end user version).

According to the final results obtained from user experiences and internal tests, the application was successful in its objective. From the user tests, it was possible to conclude that even though the application is user-friendly, it requires some level of previous technical understanding. It also became obvious that after some time getting acquainted with the app, the user was better equipped to get the most out of all the tools provided. All users were capable of fulfilling all objectives introduced in Chapter 1 which leads us to the conclusion that all objectives were successfully attained.

Both during the implementation of the system and all the tests, it was clear the application would benefit from improvements. However, these are general limitations of the current AR technology. While testing the app in an indoor environment, it was noted that the lack of GPS signal indoors would sometimes interfere with the maps' positions, causing problems during the alignment process. On the user end, it was established that a correct mapping was paramount given that an incorrect mapping would lead to imperfect map

detection. Inconstant distances in map detection are also undesirable for the same reason. The technology is also not equipped to deal with sudden alterations to the environment mapped so it was possible and common to lose track of a map by making changes to the object layout in the mentioned map or even by having parts of it slightly obscured like, for example if a person happened to walk across. As for the Framework itself, the fact that a stable internet connection is required is also seen as a limitation, as without this it is impossible to load or save any content.

Concerning future work that could be added to the current thesis, two big features could immensely help the creator: adding both help/tutorial videos for users interacting for the first time with the app to help them get to know and understand all the features provided and an optional 2D view of each augmented world for easier route/story creation. Another feature to be implemented is the addition of multimedia content to allow creators to prepare different options within their augmented worlds. This could greatly improve the final user experience, giving them more control and possibilities to choose from when navigating through a route.

Despite some limitations, the application showed itself useful for real-world activities both indoors and outdoors. It can be used as a tool by any tourist guide or commercial space owner/manager to set up a training route, walking trail, or showcase points of interest to any end user that wants to visit the place and experience a new level of reality, while navigating through an infinite amount of augmented worlds previously prepared by a creator.

Chapter 7 – Bibliography

- A. Bangor, P. K. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. pp. 114-123.
- Andrey Dimitrov, M. G.-F. (2014). Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. pp. 37-49.
- A. I. Martins, A. F. (2015). European portuguese validation of the system usability scale (sus). pp. 293-300.
- Brooke, J. (1996). Usability evaluation in industry.
- Brooke, J. (2013). Sus: A retrospective. pp. 29-40.
- Chia-Yen Chen, B. R.-S. (2014). Multimedia augmented reality information system for museum guidance. pp. 315-322.
- C. C. Ho, M. H. (2019). Markerless indoor/outdoor augmented reality navigation device based on orb-visual-odometry positioning estimation and wall-floor-boundary image registration. pp. 199-204.
- Changyang Li, W. L.-F. (2022). Interactive augmented reality storytelling guided by scene semantics. pp. 1–15.
- D. Roca, S. L.-V. (2013). Low-cost aerial unit for outdoor inspection of building façades. pp. 128-135.
- F. Demim, A. N. (2018). Robust svsf-slam algorithm for unmanned vehicle in dynamic environment. pp. 1-5.
- Feiner, S. G. (2006). Visualizing and navigating complex situated hypermedia in augmented and virtual reality. pp. 155-158.
- Hyojoon Bae, J. W.-F. (2015). Fast and scalable 3D cyber-physical modeling for high-precision mobile augmented reality systems. pp. 1275–1294.
- I. Herbst, A. B. (2008). Multi-dimensional interactive city exploration through mixed reality. pp. 259-260.
- John V. Pavlik, F. B. (2013). The Emergence of Augmented Reality (AR) as a Storytelling Medium in Journalism. pp. 4-59.
- M. W. M. G. Dissanayake, P. N. (2001). A solution to the simultaneous localization and map building (slam) problem. pp. 229–241.

- P. Sinclair, K. M. (2002). Links in the palm of your hand: Tangible hypermedia using augmented reality. pp. 127-136.
- Pileun Kima, J. C. (2018). SLAM-driven robotic mapping and registration of 3D point clouds. pp. 38-48.
- P. Wu, Z. S. (2018). Improved ekf-slam algorithm of unmanned helicopter autonomous landing on ship. pp. 5287–5292.
- Ryokai, J. C. (2001). Making space for voice: Technologies to support children's fantasy and storytelling. *Personal and ubiquitous computing* 5, pp. 169–190.
- R. Nóbrega, J. J. (2017). Mobile location-based augmented reality applications for urban tourism storytelling. pp. 1-8.
- S. Skuratovskyi, I. G. (2019). Outdoor mapping framework: from images to 3d moded. pp. 296-399.
- Terrell Glenn, A. I. (2020). StoryMakAR: Bringing stories to life with an augmented reality & physical prototyping toolkit for youth. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14.
- T. Starner, S. M. (1997). Augmented reality through wearable computing. pp. 386-398.
- W. Liu, C. W. (2019). Ground camera images and UAV 3D model registration for outdoor augmented reality.
- Walczak, D. R. (2013). Creation of interactive AR content on mobile devices. *International Conference on Business Information Systems*, pp. 258–269.
- Yu, L. W. (2018). Outdoor navigation with handheld augmented reality. pp. 237-241.
- J. Bacca, S. Baldiris, R. Fabregat, Kinshuk (2019). "Framework for designing motivational augmented reality applications in vocational education and training", *Australasian Journal of Educational Technology*, pp. 102–117
- J. Zubizarreta, I. Aguinaga, A. Amundarain (2019). "A framework for augmented reality guidance in industry", *International Journal of Advanced Manufacturing Technology*, pp. 4095–4108